Computational Methods in Applied Sciences

João Manuel R. S. Tavares
R. M. Natal Jorge (Eds.)

Advances in Computational Vision and Medical Image Processing

Methods and Applications

Springer

Advances in Computational Vision and Medical Image Processing

Computational Methods in Applied Sciences

Volume 13

Series Editor

E. Oñate
International Center for Numerical Methods in Engineering (CIMNE)
Technical University of Catalunya (UPC)
Edificio C-1, Campus Norte UPC
Gran Capitán, s/n
08034 Barcelona, Spain
onate@cimne.upc.edu
www.cimne.com

For other titles published in this series, go to
www.springer.com/series/6899

João Manuel R.S. Tavares · R.M. Natal Jorge
Editors

Advances in Computational Vision and Medical Image Processing

Methods and Applications

Springer

Editors

João Manuel R.S. Tavares

University of Porto (UP)
Faculty of Engineering (FEUP)
Institute of Mechanical Engineering
and Industrial Management (INEGI)
Rua Dr. Roberto Frias, s/n
4200-465 Porto
Portugal
tavares@fe.up.pt

R.M. Natal Jorge

University of Porto (UP)
Faculty of Engineering (FEUP)
Institute of Mechanical
Engineering (IDMEC)
Rua Dr. Roberto Frias, s/n
4200-465 Porto
Portugal
rnatal@fe.up.pt

ISBN: 978-1-4020-9085-1 e-ISBN: 978-1-4020-9086-8

Library of Congress Control Number: 2008934777

Printed on acid-free paper

springer.com

Preface

Computational methodologies of signal processing and imaging analysis, namely considering 2D and 3D images, are commonly used in different applications of the human society. For example, Computational Vision systems are progressively used for surveillance tasks, traffic analysis, recognition process, inspection purposes, human-machine interfaces, 3D vision and deformation analysis.

One of the main characteristics of the Computational Vision domain is its inter-multidisciplinary. In fact, in this domain, methodologies of several more fundamental sciences, such as Informatics, Mathematics, Statistics, Psychology, Mechanics and Physics are usually used. Besides this inter-multidisciplinary characteristic, one of the main reasons that contributes for the continually effort done in this domain of the human knowledge is the number of applications in the medical area. For instance, it is possible to consider the use of statistical or physical procedures on medical images in order to model the represented structures. This modeling can have different goals, for example: shape reconstruction, segmentation, registration, behavior interpretation and simulation, motion and deformation analysis, virtual reality, computer-assisted therapy or tissue characterization.

The main objective of the ECCOMAS Thematic Conferences on Computational Vision and Medical Image Processing (VIPimage) is to promote a comprehensive forum for discussion on the recent advances in the related fields trying to identify widespread areas of potential collaboration between researchers of different sciences.

The present book contains the extended versions of sixteen papers selected from the works presented in the first ECCOMAS thematic conference on Computational Vision and Medical Image processing (VIPimage 2007), held in Faculty of Engineering of University of Porto, Portugal. It collects the state-of-the-art on the subject of Computational Vision and Medical Image processing contributing for the development of these areas of knowledge.

The Editors would like to take this opportunity to thank to the European Community on Computational Methods in Applied Sciences, the Portuguese Association

of Theoretical, Applied and Computational Mechanics, the University of Porto, all sponsors, all members of the International Scientific Committee, and to all Invited Lecturers and Authors.

Faculty of Engineering,
University of Porto, Portugal

João Manuel R.S. Tavares
R. M. Natal Jorge

Contents

Chapter 3
Robust Shape Estimation with Deformable Models

Jorge S. Marques, Jacinto C. Nascimento, Arnaldo J. Abrantes, and Margarida Silveira

Abstract This paper addresses the estimation of 2D object boundary from noisy data, using deformable contours. First, it discusses the relationship between deformable contours and other Pattern Recognition algorithms (e.g., Kohonen maps, mean shift, fuzzy c-means) and derives a unified framework which allows a joint formulation for a wide set of methods. Afterwords, the paper addresses the estimation of deformable curves in cluttered images, assuming that there is a large number of outlier features detected in the image. The paper presents two robust algorithms: the adaptive snake for static objects and a robust tracker (S-PDAF) for moving objects in video sequences. The advantages of both algorithms with respect to classic methods are illustrated by examples.

3.1 Introduction

Object segmentation has been thoroughly investigated in the last two decades. Many algorithms have been proposed based on different assumptions (e.g., homogeneity criteria, edge linking, topographic maps, etc.). Deformable models are amongst the most popular techniques [1–11]. They approximate the object boundary using a deformable curve and estimate the curve configuration by the minimization of an energy function.

The energy function may include different types of information about the problem e.g., image cues (edge points, color and texture features), prior knowledge about the object shape and motion models. This information is either provided by the user or learned from the data. It depends on the application.

J.S. Marques, J.C. Nascimento, and M. Silveira
ISR, Instituto Superior Técnico, Lisboa

A.J. Abrantes
Instituto Superior de Engenharia de Lisboa

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

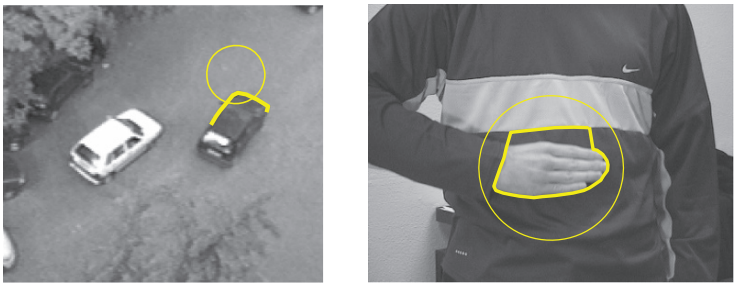


Fig. 3.1 Difficulties in elastic contour estimation: initialization (left) and outlier features (right). Each figure shows the initial contour (circle) and the final contour after adaptation

Despite their ability to fuse different sources of information in a sound framework, deformable models have a couple of drawbacks:

- *Initialization:* the final estimate of the contour is strongly dependent on the contour initialization.
- *Robustness:* the elastic contour is attracted by outlier features detected in the image (e.g., inner edges or edges associated to other objects in the scene).

These drawbacks have prevented a widespread use of deformable contours since the model has to be manually initialized close to the object boundary and no outlier features should be present in the vicinity of the object boundary. Figure 3.1 illustrates both difficulties.

We will address the robustness issue in this paper as well as a more fundamental question concerning the relationship between deformable models (Snakes) and other well known algorithms (e.g., clustering methods, mean shift, neural networks). All these methods approximate data points by a set of prototypes (model points) and it is useful to understand what are the similarities and differences among them. We will propose a unified framework which allows a joint formulation of several Pattern Recognition algorithms together with Snakes (see Section 3.3). We will show that they all share a common structure and can be studied together. Then we address the robustness issue and describe two new algorithms for robust shape estimation with deformable models. Both of them assign probabilistic weights to the detected data. The first algorithm uses an adaptive potential function to estimate static objects (Section 3.4) while the second uses a robust tracker based on data association filtering (Section 3.5). The work reviewed in this paper was published before in [12–22].

3.2 Snake Algorithm

This section describes object boundary extraction using the classic Snake algorithm [1]. We will assume that the deformable curve (Snake) is defined by a sequence of 2D points and the image features are edge points.

Let I be an image and let $y = (y_1, ..., y_m)$ be a set of edge points detected in the image. We want to approximate a subset of y by a curve $x = (x_1, \ldots, x_n)$ where x_i are 2D points. This problem has two main difficulties: (1) which subset of y should be approximated? and (2) how do we associate data points to model points?

The snake algorithm addresses both difficulties by minimizing an energy function E_T with two terms: an image dependent term and a regularization term

$$E_T(x) = E(x) + E_r(x) \tag{3.1}$$

The first term is given by a sum of samples of a potential function P along the curve

$$E = \sum_i P(x_i) \tag{3.2}$$

The curve will tend to follow the valleys of the potential function in order to obtain small energy values. The second term E_r is a regularization energy which assigns a high cost to long contours and contours with unusual shapes. The minimization of (3.1) tries to achieve both goals. This operation is often done by using a gradient descent algorithm starting from an initial curve configuration, provided by the user.

The potential function depends on the image and can be defined in several ways. The main idea is to guarantee that the valleys of the potential function follow the object boundary. The potential function can be obtained by filtering the gradient magnitude or by anisotropic diffusion [7].

The potential function can also be obtained by filtering the edge points detected in the image with a lowpass filter (Cohen potential [2])

$$P(x) = -\sum_j G(x - y_j) \tag{3.3}$$

where G is a Gaussian kernel

$$G(x) = Ce^{-\frac{1}{2\sigma^2}\|x\|^2} \tag{3.4}$$

3.3 Unified Framework

Snakes and several other Pattern Recognition methods try to represent a set of observations $y = (y_1, ..., y_m)$ by a smaller set of prototypes $x = (x_1, \ldots, x_n)$. The key question is: *what is the relationship between all these methods?* We will show that many of these methods share a common structure. They minimize the same type of energy and can be implemented using the same recursion.

To approximate the data y by a set of prototypes x we need to associate the data points y_j to model units x_i. The difficult question is: which model unit x_i corresponds to each data point y_j? We will answer this question by assuming that all model units are associated to all data points ... but with different weights (see Fig. 3.2).

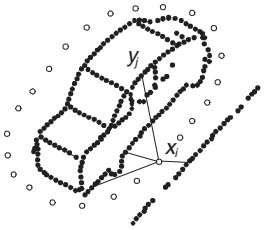


Fig. 3.2 Association of data points to model points

Table 3.1 Unified framework: weights and influence functions

Method	w_{ji}	v_{ji}
Snakes/Mean Shift	$-\frac{G(y_j-x_i)}{d_{ji}}$	$G(y_j-x_i)$
c-means	$\delta(i-i^*)$	$\delta(i-i^*)$
Kohonen maps	$\sum_{k=i^*-p}^{i^*+p}\delta(i-k)$	$\sum_{k=i^*-p}^{i^*+p}\delta(i-k)$
Fuzzy c-means	$\left[\sum_k\left(\frac{d_{ji}}{d_{ki}}\right)^{1/(q-1)}\right]^{-q}$	$\left[\sum_k\left(\frac{d_{ji}}{d_{ki}}\right)^{1/(q-1)}\right]^{-q}$
Elastic nets	$-\frac{\log\sum_k G(y_j-x_k)}{d_{ji}}$	$\frac{G(y_j-x_i)}{\sum_k G(y_j-x_k)}$

We will assume that the model x minimizes a weighted squared error criterion

$$E=\sum_{i,j} w_{ji}(d)\,\|y_j-x_i\|^2 \tag{3.5}$$

where $w_{ji}(d)$ depends on the set of distances from the data points to the model prototypes $d=\{d_{ji}=\|y_j-x_i\|^2\}$. The weight w_{ji} associated to a tentative match (y_j,x_i) depends on the distance between these two points but it may also depend on the distances from other data points to x_i. This creates competition among the model points. For example we can implement a winner-takes-all strategy by defining $w_{ji}=1$ if x_i is the closest prototype to y_j and $w_{ij}=0$ otherwise. Only the closest prototype is attracted by the data point.

The Snake algorithm with the Cohen potential can be written in this way. The Cohen energy

$$E=-\sum_{ij} G(y_j-x_i) \tag{3.6}$$

is equal to the squared error criterion (3.5) if the weights are defined by

$$w_{ji}(d)=-\frac{G(y_j-x_i)}{\|y_j-x_i\|^2} \tag{3.7}$$

In this case, the weight w_{ji} depends only on the distance $\|y_j-x_i\|$.

Many other algorithms try to minimize the energy (3.5). That is the case of the following algorithms: Kohonen nets, elastic nets, c-means, fuzzy c-means and mean

shift. The weights associated to each of these algorithms are shown in Table 3.1 (see [12, 15] for a proof) where i^* is the index of the model unit closest to y_j, $\delta(i)$ is the Kronecker symbol, p is the neighborhood radius of the Kohonen maps and q is the fuzziness parameter in fuzzy c-means. It is interesting to notice that most of these algorithms are based on competitive learning since each weight w_{ji} depends on all the distances $\|y_p - x_i\|, \forall p$. The only exceptions are Snakes and the mean shift algorithm.

To estimate the model $x = (x_1, \ldots, x_n)$ we have to minimize the energy function. This can be done using a gradient descent algorithm. Again the minimization of all these cases can be done in the same way. It can be shown that the gradient algorithm for the energy (3.5) is given by

$$x_i^{t+1} = x_i^t + m_i(\xi_i^t - x_i^t) \tag{3.8}$$

$$\xi_i^t = \frac{\sum_j v_{ji} y_j}{\sum_j v_{ji}} \tag{3.9}$$

where the ξ_i^t is a centroid of the data and v_{ji} measures the influence of data point y_j on the model point x_i. The coefficients v_{ji} can be derived from the weights w_{ji} by [12]

$$v_{ji} = w_{ji} + \sum_k d_{jk} \frac{\partial w_{jk}}{\partial d_{ji}} \tag{3.10}$$

and they are shown in Table 3.1.

Equation (3.8) has a simple interpretation. The model configuration x is recursively updated. In each iteration, each model point x_i is attracted toward a centroid of the data ξ_i in the vicinity of the model point. The algorithm stops when the model points converge to the centroids. When we consider the regularization energy in (3.1), Eq. (3.8) also includes a regularization force.

All the above algorithms can be implemented in this way. The only difference between them is related with the influence functions v_{ji} which measure the contribution of each data point y_j to the centroid ξ_i. It is enlightening to see the attraction regions associated to each model point x_i i.e., the regions such that $v_{ji} > T$ when we vary the feature position, y_j. This is shown in Fig. 3.3. The Snake algorithm has omni directional attraction regions while the attraction regions of competitive learning methods are shape dependent and they are often unbounded.

All these algorithms can be used for shape estimation but they behave in different ways. Figure 3.4 (left) shows the performance of Snakes when it is initialized with a circle outside the object. The elastic curve converges to the object boundary but it fails to represent the concavities. The reason is simple. Since the attraction regions of Snakes are omni directional the contour is never attracted by points inside the concavity. The fuzzy c-means algorithm solves this problem and is able to represent the concavities (Fig. 3.4 (right)). This improvement is due to competitive learning.

Competitive learning brings several advantages. First, the model tries to represent all the data. This is important if the contour initialization is poor and all the data points detected in the image are relevant. However, it is a major difficulty if we

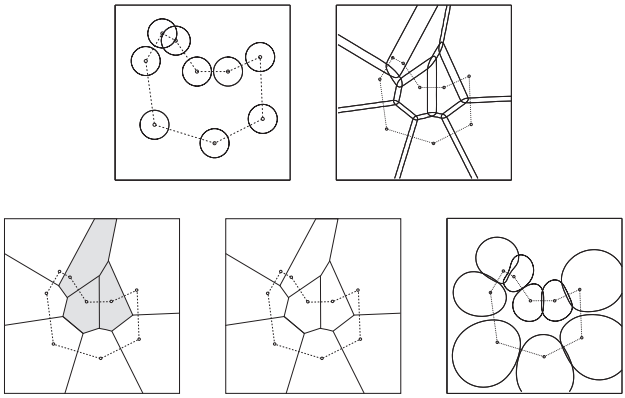


Fig. 3.3 Attraction regions of snakes/mean shift, elastic nets, Kohonen maps, c-means and fuzzy c-means (starting on the upper left image) in the feature space

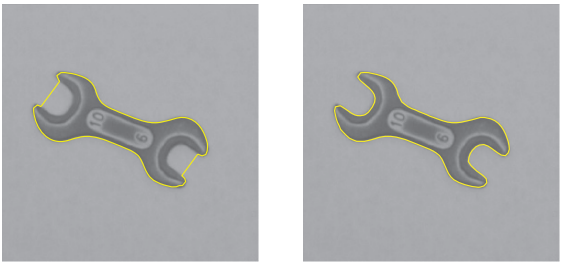


Fig. 3.4 Results obtained with snakes (left) and fuzzy c-means (right)

have edge points outside the boundary. To avoid this drawback we should bound the attraction regions. This can easily be done by creating an additional unit (noise model) with special properties, keeping the same type of energy. In addition, the snake points tend to collapse if they are in a region of high concentration of data. This effect is not observed in competitive learning methods since different units are attracted by different data points.

The unified framework described in this section can be extended in several ways. For example, it can be used with extended features containing color, texture of gradient information. Figure 3.5 shows an image of a ceramic material obtained with an electronic microscope. This image cannot be segmented using edge points only. The contour model is attracted by several grains. This problem can be overcome by using extended features y_j containing the edge coordinates and the gradient direction as well. The energy and the model update equations remain the same. The new model behaves as a directional active contour which is attracted by data points which are near and have similar gradient directions [13].

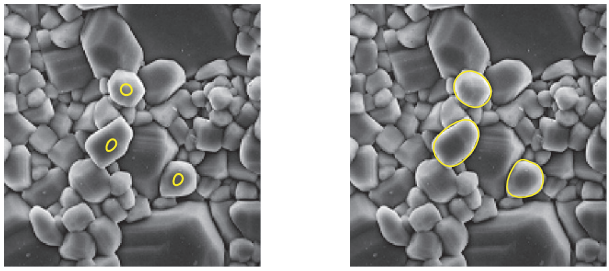


Fig. 3.5 Electronic microscopy: contour estimation using edge points and gradient direction: initial contours (left) and final contours (right)

3.4 Robust Shape Estimation

Most images have clutter (uninteresting objects and features). Only a subset of the features detected in the image belong to the object boundary and should attract the elastic contour. The other features should be considered as outliers and ignored. This segmentation problem is not addressed by the previous algorithms and that is the reason why they are not robust in the presence of outliers (see Fig. 3.1 and 3.6).

This problem is addressed in this section for static objects and in the next section for the case of object tracking in video sequences. We explore two ideas. First, we use middle level features (curve segments) instead of low level ones (edge points). Curve segments, also called strokes, are more robust and in smaller number. Second, we explicitly assume that each stroke is either valid or invalid (outlier). We therefore assign confidence degrees to the strokes and the estimation algorithms take into account the confidence degrees.

First, we apply an edge detection algorithm to the image and then link neighboring edges in order to obtain the curve segments. Let

$$y^k = (y_1^k, \ldots, y_{m^k}^k) \tag{3.11}$$

be the k-th stroke (where m^k is the number of edge points of the k-th segment) and let y be the set of all the strokes detected in the image. We will assign a binary label $v^k \in \{0,1\}$ to each stroke. We assume that $v^k = 1$ if the stroke belongs to the object boundary and $v^k = 0$ otherwise. The stroke labels $v = (v^1, ..., v^m)$ are not observed.

3.4.1 Known Labels

Let us first assume that the labels are known. This is not true of course. In this case, we can define an image potential

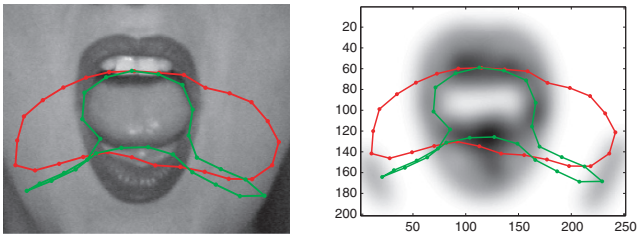


Fig. 3.6 Lip estimation with classic snakes: initial (red) and final (green) contours. Observed image (left) and potential function (right)

$$P(x;y,v) = \sum_k v^k P^k(x;y^k) + (1-v^k)K \tag{3.12}$$

where $P^k(x;y^k)$ is the potential created by the k-th stroke assuming that it is a valid one and K is a constant potential created by an invalid stroke. Adopting the Cohen potential we obtain

$$P^k(x;y^k) = -\sum_{j=1}^{m^k} G(y_j^k - x) \tag{3.13}$$

The data energy is given by

$$E(x;y,v) = \sum_i P(x_i;y,v) \tag{3.14}$$

Shape estimation is achieved by minimizing the energy

$$E_T(x;y,v) = E(x;y,v) + E_r(x) \tag{3.15}$$

This problem can also be addressed in a probabilistic framework by defining a Gibbs distribution

$$p(y,v,x) = \frac{1}{Z} e^{-E(x;v)-E_r(x)} \tag{3.16}$$

and estimate x as the most probable shape i.e. optimizing $p(y,v,x)$ with respect to x assuming that the other variables are known. This approach is equivalent to the minimization of the energy $E_T = E + E_r$.

3.4.2 Unknown Labels

In practice, the stroke labels v are not known. We therefore have to estimate x in the presence of incomplete observations. This could be done by using the joint probability $p(y,x)$ and computing the most probable configuration of x. We do not have an analytic expression for $p(y,x)$ since the marginalization of $p(y,v,x)$ defined in (3.16)

is not feasible. However, we can circumvent this difficulty by using the Expectation-Maximization (EM) method which allows the estimation of unknown variables in the presence of missing data [23, 24].

The EM method is an iterative procedure with two steps. The first step (E-step) computes an auxiliary function

$$U(x,\hat{x}) = E_v\{\log p(y,v,x)|y,\hat{x}\} \tag{3.17}$$

where $\hat{x}$ is the most recent estimate of the contour. Function $U(x,\hat{x})$ has an intuitive meaning. Since we do not know $\log p(y,v,x)$ (v is not observed) we replace it by its expected value with respect to v. The second step (M-step) maximizes $U(x,\hat{x})$ with respect to x and replaces $\hat{x}$ by this estimate. The sequence of estimates produced by the EM method converges to a local maximum of the function $p(x,y)$.

The auxiliary function for our problem can be easily computed (see Appendix) and it is a sum of two terms: a data dependent energy as before and a regularization energy

$$U(x,\hat{x}) = E(x,\hat{x}) + E_r(x) \tag{3.18}$$

The data dependent term is obtained by sampling a potential function along the curve as before

$$E(x,\hat{x}) = \sum_i P_a(x_i;y) \tag{3.19}$$

but the potential function has changed. The new potential function is a weighted sum of the potentials of all the strokes.

$$P_a(x) = \sum_k w^k P^k(x;y^k) \tag{3.20}$$

the weights being the label probabilities

$$w^k = P(v^k = 1|y^k,\hat{x}) \tag{3.21}$$

This is a very interesting result since all the strokes detected in the image contribute to the image potential but with different weights. The weights can also be interpreted as confidence degrees associated to the strokes. They change at each new iteration since they depend on the best shape estimate available. That is why $P_a(x)$ will be denoted as an **adaptive potential**.

Figure 3.7 shows an example of lip contour estimation starting with a very poor initialization. The Snake algorithm provides meaningless results (see Fig. 3.6) while the use of robust estimation method solves the problem well. The adaptive potential changes during the estimation process and at the end the outlier strokes (teeth, nose, cheek) receive negligeable weights while the lip stroke has a high weight. The same can be observed in Fig. 3.8 for another challenging example. These examples show the ability of the adaptive Snake to discard the influence of outliers.

Additional information about this algorithm can be found in [17].

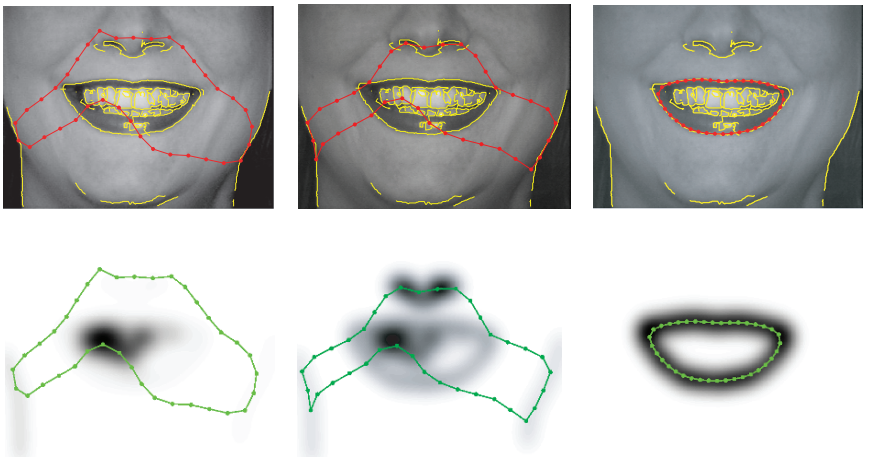


Fig. 3.7 Lip segmentation with adaptive potential: initial contour (left), intermediate contour (middle) and final contour (right). The top row shows the image and the bottom row shows the adaptive potential (From [17], 2005 IEEE)

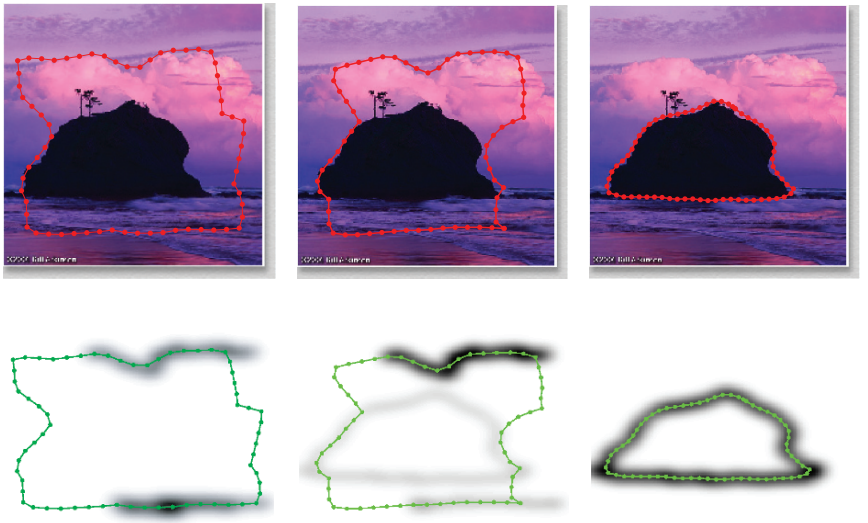


Fig. 3.8 Contour estimation using adaptive potential (From [17], 2005 IEEE)

3.5 Robust Shape Tracking

Elastic models are often used to track moving objects in video sequences and have been used in a wide variety of applications [8, 28–33]. Robustness is again a key issue since many features detected in the image are outliers and jeopardize the performance of classic trackers. Classic trackers are often based on Kalman filtering and they break after a few seconds since the curve becomes attracted by the outliers.

For the sake of simplicity, we will first assume that all the features are valid. The tracking algorithm derived in this way is not robust. We will then describe a robust algorithm to track moving objects in video sequences.

3.5.1 Tracking with Valid Features

We assume that the object boundary moves slowly during the tracking interval. In every new frame we should update the previous estimate using the new set of strokes detected in the image.

Suppose we detect a set of strokes in the vicinity of the object boundary and let us assume that all strokes are valid. Let $y_j^k(t)$ denote the j-th point of stroke k and let $y(t)$ be the set of all edge points detected at frame t.

The object shape changes during the tracking operation. We will assume that the boundary parameters $x(t)$ are described by a linear dynamical model[1]

$$\begin{aligned} x(t) &= Ax(t-1) + w(t) \\ y(t) &= Cx(t) + v(t) \end{aligned} \tag{3.22}$$

where A, C are known matrices $w(t) \sim N(0, Q)$ is white Gaussian noise which accounts for the random evolution of shape and $v(t) \sim N(0, R)$ is white measurement noise. We assume that shape evolution is described by a linear stochastic equation and the observation vector $y(t)$ containing the coordinates of the edge points is linearly related to the shape parameters. Vector $x(t)$ is denoted as state vector in this context.

We wish to estimate $x(t)$ given all the features detected in past and current frames $Y(t) = (y(1), \ldots, y(t))$. Under these hypothesis, the *a posteriori* distribution of the state vector $p(x(t)|Y(t))$ is a normal distribution $N(\hat{x}(t), P(t))$ with mean vector $\hat{x}(t)$ and covariance matrix $P(t)$. The mean and covariance are updated by the Kalman filter equations

$$\begin{aligned} \hat{x}(t) &= \hat{x}^-(t) + K(t)\,[y(t) - C\hat{x}^-(t)] \\ P(t) &= (I - K(t)C)P^-(t) \end{aligned} \tag{3.23}$$

where $\hat{x}^-(t) = A\hat{x}(t-1), P^-(t) = AP(t-1)A^T + Q$ and $K(t)$ is the Kalman gain [8]. The shape estimates computed by Kalman filtering are optimal provided that model (3.22) holds and all the image features are valid. This algorithm is known as Kalman Snakes [27].

[1] The contour model is often represented using a reference shape which changes according to a global motion model (e.g., rigid or affine transformation) plus an elastic deformation (see [8] for details).

3.5.2 Tracking with Invalid Data

The previous algorithm usually fails after a few seconds since the elastic contour is attracted towards invalid data. Figure 3.9 illustrates this problem for a synthetic example. Suppose we detect a set of strokes y^k in the vicinity of the object. How can we distinguish valid strokes from the invalid ones?

We do not know which strokes are valid. Therefore we will associate a binary label v^k to each stroke and we will consider all possible combinations of valid and invalid strokes $v = (v^1, \ldots, v^m)$. We denote this binary sequence by a data interpretation. For the sake of simplicity the binary sequence v will be represented by an integer number I in the range $\{0, ..., M-1\}$ where $M = 2^m$. Figure 3.10 shows eight interpretations for the synthetic problem of Fig. 3.9.

If we discard invalid data, we have a different set of valid observations for each interpretation I which we denote by $y^I(t)$. The state model is now given by

$$\begin{aligned} x(t) &= Ax(t-1) + w(t) \\ y^I(t) &= C^I x(t) + v(t) \qquad \text{(interpretation } I\text{)} \end{aligned} \tag{3.24}$$

where matrix C^I depends on the interpretation I and it is obtained from matrix C by removing the rows associated to invalid data.

The tracking problem is now modeled by a set of dynamical models which compete to represent the data. Only one model is active at each instant of time and we do not know which. The main question is: how do we extend Kalman filtering when we have multiple dynamic models and we do not know the sequence of active models?

This problem was studied in Control theory e.g., in the context of abrupt changes in dynamical systems and target tracking in radar systems [25, 26]. The *a posteriori* distribution $p(x(t)|Y(t))$ is known. It is a mixture of Gaussians but the number of

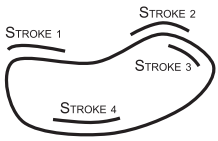


Fig. 3.9 Feature detection: valid and invalid data

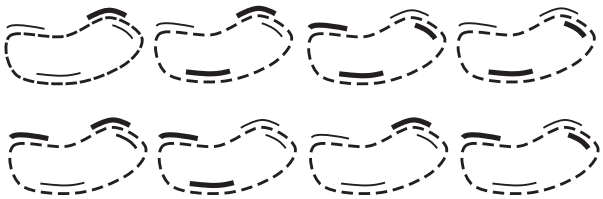


Fig. 3.10 Data interpretations. Valid strokes are represented in bold

modes grows exponentially with t. Therefore, the optimal solution is analytically defined but it is unfeasible: we cannot compute the parameters of all the modes.

Suboptimal methods have been proposed to overcome this difficulty. A robust algorithm which has been successfully used in practice is the probability data association filter (PDAF) proposed by Bar-Shalom [25]. This method assumes that the predictive distribution of the state vector is Gaussian

$$p(x(t)|Y(t-1)) = N(\hat{x}^-(t), P^-(t)) \tag{3.25}$$

where $x^-(t)$ is the expected value of the state at time t given the observations until time $t-1$ and $P^-(t)$ is the corresponding covariance matrix.

Under this hypothesis, the *a posteriori distribution* of the state $p(x(t)|Y(t))$ is also Gaussian $N(\hat{x}(t), P(t))$. Furthermore, the mean and the covariance matrix are obtained by combining the outputs of the Kalman filters associated to each of the models as follows [25]

$$\begin{aligned} \hat{x}(t) &= \hat{x}^-(t) + \sum_{I=0}^{M(t)-1} \alpha_I K_I(t)\,[y_I(t) - C_I\hat{x}^-(t)] \\ P(t) &= [I - \sum_{I=0}^{M(t)-1} \alpha_I(t) K_I(t) C_I] P^-(t) + \sum_{I=0}^{M(t)-1} \alpha_I(t)\hat{x}_I(t)\hat{x}_I(t)^T - \hat{x}(t)\hat{x}(t)^T \end{aligned} \tag{3.26}$$

where $\alpha_I = P(I|y(t), \hat{x}(t))$ is the probability of the I-th interpretation also called data association probability and $K_I(t)$ is the Kalman gain associated to model I. This algorithm has a simple meaning. All data interpretations are used to update the *a posteriori* distribution of the state variable (contour parameters), but with different weights.

The data association probabilities $\alpha_I(t)$ play an important role. Since they depend on the contour estimates, we repeat the update step several times and compute the association probabilities in each iteration. The process is stopped after a prespecified number of iterations.

The computation of the association probabilities is based on a simple model for the invalid data which is described in [16]. The robust shape tracker we have described is called Shape Probabilistic Data Association Filter (S-PDAF) tracker and the main steps are summarized in Table 3.2

A comparison between the Kalman Snakes and the S-PDAF tracker in lip reading is shown in Fig. 3.11. Kalman snakes are attracted by outliers and provide meaningless estimates after a few seconds while the S-PDAF tracker is able to track the lips in a robust way.

Figure 3.12 shows face tracking with three independent contours for the lips and eyebrows. This is useful for the recognition of facial expressions. The tracker manages to track face features well despite the large number of outliers detected in the face e.g., in the nostrils, in the eyes and inside the mouth. A correct evaluation of the data interpretations is automatically performed by the algorithm and the outlier strokes do not influence the shape estimates.

Table 3.2 S-PDAF tracker

The following steps are performed for every new frame.

1. **Contour Prediction:**
   Compute $p(x(t)|Y(t-1)) = N(\hat{x}^-(t), P^-(t))$

$$\hat{x}^-(t) = Ax(t-1) \qquad P^-(t) = AP(t-1)A^T + Q \tag{3.27}$$

2. **Feature extraction:**
   Detect edge points by directional search along lines orthogonal to the predicted contour $y^-(t) = C\hat{x}^-(t)$ and link them to obtain image strokes.

3. **Contour update:**
   Compute $p(x(t)|Y(t)) = N(\hat{x}(t), P(t))$ combining multiple Kalman filters.
   Repeat the following steps N times (typically $N = 3$)
   (a) compute data association probabilities [16]: $\alpha_I(t) = P(I|y(t), \hat{x}^-(t))$
   (b) PDAF filter

$$\begin{aligned} \hat{x}(t) &= \hat{x}^-(t) + \textstyle\sum_{I=0}^{M(t)-1} \alpha_I K_I(t)\,[y_I(t) - C_I\hat{x}^-(t)] \\ P(t) &= [I - \textstyle\sum_{I=0}^{M(t)-1} \alpha_I(t) K_I(t) C_I] P^-(t) + \\ &\quad + \textstyle\sum_{I=0}^{M(t)-1} \alpha_I(t)\hat{x}_I(t)\hat{x}_I(t)^T - \hat{x}(t)\hat{x}(t)^T \end{aligned} \tag{3.28}$$

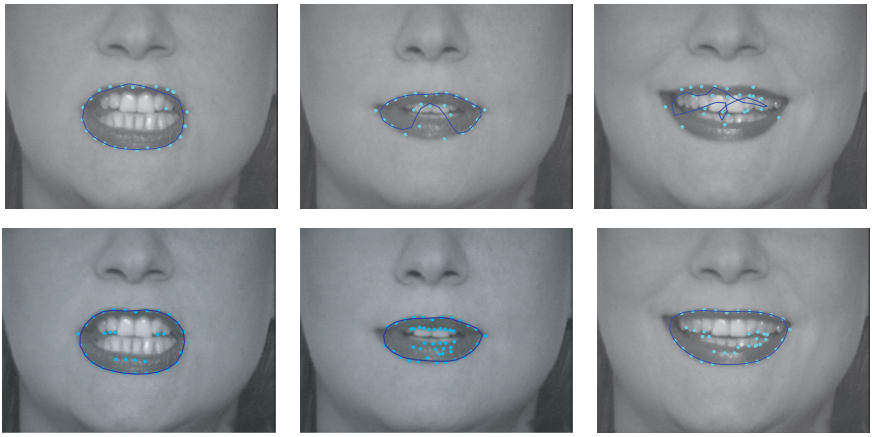


Fig. 3.11 Tracking results with Kalman snakes (top row) and S-PDAF (bottom row) (From [16], 2004 IEEE)

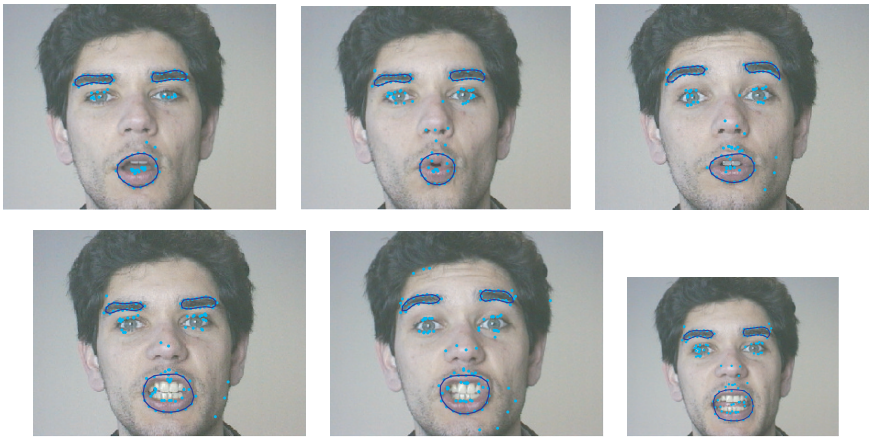


Fig. 3.12 Face tracking with S-PDAF tracker (From [16], 2004 IEEE)

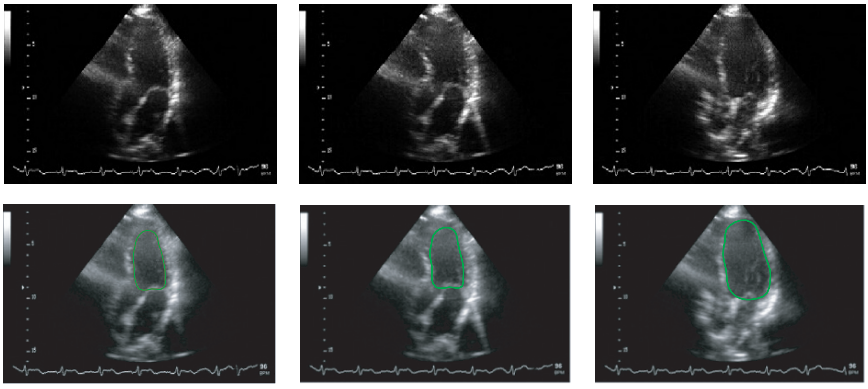


Fig. 3.13 Heart tracking with S-PDAF tracker: ultrasound sequence (top) and tracking results (bottom)

A third example is shown in Fig. 3.13 which displays images extracted from a long ultrasound sequence of the heart. The tracker tries to estimate the boundary of the left ventricle (endocardium) in several cardiac cycles. This is a difficult task due to the low signal to noise ratio and the presence of multiplicative noise. Furthermore, the ultrasound images of the heart often show an edge drop-out effect i.e., some regions of the ventricle boundary may not be seen. The S-PDAF tracker manages to solve this problem well in a robust way despite the presence of outliers.

A detailed description of the S-PDAF tracker can be found in [16].

3.6 Multiple Competing Contours

The previous algorithms are focused on the estimation of single objects in the presence of clutter (outlier features). They can also be used to estimate multiple objects with independent active contours. This approach performs well if the objects are well separated in the image domain (see Fig. 3.12). In more complex situations we need to perform a joint estimation of all the contours using competitive learning techniques [19]. This extension is discussed in this section.

We wish to jointly estimate N deformable contours

$$x^q = (x_1^q, \ldots, x_n^q) \qquad q = 1, \ldots, N \tag{3.29}$$

where x^q denotes the q-th elastic curve. We assume that the image features are curve segments (strokes) obtained by edge linking as before. Furthermore, we assume that each stroke belongs to the boundary of a single object or it is an invalid stroke. This means that shared strokes, belonging to multiple objects are not allowed. To account for these multiple hypotheses we associate to each stroke y^k, $N+1$ binary variables v_q^k such that $v_q^k = 1, (q > 0)$ iif stroke k is associated with model x^q and $v_0^k = 1$ iif stroke k is an outlier.

The contour energy E is now defined as a sum of the energies of the N contours

$$E(x) = \sum_{q=1}^{N} E^q(x^q, y, v^q) \tag{3.30}$$

where E^q has an expression similar to (3.14,3.12) (Section 3.4) with the flags v^k replaced by v_q^k.

Since the labels v are unknown, the EM method is again used to estimate the contour parameters x^q with incomplete observations, leading to a recursive estimation algorithm similar to the one described in Section 3.4. The auxiliary function associated to contour q is given by

$$U^q(x^q, \hat{x}) = \sum_i P_a^q(x_i^q; y) \tag{3.31}$$

where $P_a^q(x; y)$ is the adaptive potential associated to the q-th contour, given by

$$P_a^q(x) = \sum_k w_q^k P^k(x; y^k) \tag{3.32}$$

and $w_q^k = P(v_q^k = 1 | \hat{x}, y)$ is the probability of assigning stroke k to the deformable contour x^q.

Each contour model is estimated by minimizing the total energy

$$E_T(x^q) = U(x^q, \hat{x}) + E_r(x^q) \tag{3.33}$$

where $E_r(x^q)$ is a regularization energy.

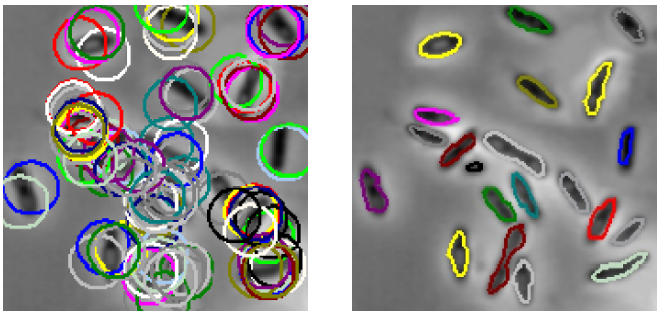


Fig. 3.14 Object estimation with multiple contours and model elimination: initialization (left) and final contour (right)

Comparing with the algorithm described in Section 3.4 the only difference concerns the estimation of the weights $w_q^k = P(v_q^k = 1|\hat{x}, y)$ which are now defined by

$$w_q^k \propto p(y^k|\hat{x}^q)\alpha_q \tag{3.34}$$

where $\alpha_q = P(v_q^k = 1|\hat{x})$ is the probability of the q-th model being active. The assignment of the k-th stroke to model q takes into account all the deformable models which compete to represent the data.

If we do not know the number of objects to be estimated or if we do not know where they are located (initialization problem) we can still use this algorithm. We initialize a large number of contours and eliminate the ones which have low probability α_q (see [19]). An example of the estimation of multiple objects at unknown locations is shown in Fig. 3.14.

3.7 Conclusions

Deformable contours are usually considered as non-robust techniques since the elastic curve must be initialized close to the object boundary and can be disturbed by outlier features outside the boundary. We have described two robust algorithms for shape estimation which allow a correct estimation of the object contour in the presence of outliers both in the case of static and moving objects. The proposed algorithms are able to distinguish valid strokes (curve segments) belonging to the object boundary from invalid ones. This is done by associating a confidence degree to each stroke, recursively updated using the EM method. A set of experiments is provided from different application areas ranging from surveillance tasks to heart tracking in ultrasound images. All these experiments show the ability of the proposed methods to discard the influence of outliers.

These ideas can be extended in several directions and we believe that interesting research opportunities can be formulated. We presented here one useful extension to

account for the estimation of multiple objects using multiple contours. The proposed algorithm is robust with respect to model initialization and it is able to estimate the number of models contained in the image. Another direction which has been followed concerns the tracking of moving objects with multiple dynamical models to account for abrupt motion and shape changes. This can be done in a robust way as well using an extension of the S-PDAF tracker. Additional information about multiple dynamical models can be found in [20–22].

Acknowledgements The work on multiple dynamical models was done in collaboration with Gilles Celeux (INRIA, Paris) and J. Miranda Lemos (IST, Lisbon). The work presented in this paper was partially supported by FCT (which includes FEDER funds).

Appendix

EM Algorithm for Shape Estimation

The EM method performs curve estimation using the auxiliary function

$$U(x,\hat{x}) = E_v\{\log p(y,v,x)|y,\hat{x}\} \tag{3.35}$$

Using the probabilistic model of the data we obtain (see Section 3.4)

$$\begin{aligned} U_v(x,\hat{x}) &= E\{-\log Z - E(x;y,v) - E_r(x)|y,\hat{x}\} \\ U_v(x,\hat{x}) &= C' - E\left\{\sum_i \sum_k v^k P^k(x_i;y^k) + (1-v^k)K \,\middle|\, y,\hat{x}\right\} - E_r(x) \\ U_v(x,\hat{x}) &= C' - \sum_i \sum_k w^k P^k(x_i;y^k) + (1-w^k)K - E_r(x) \end{aligned} \tag{3.36}$$

where

$$w^k = P(v^k = 1|y^k,\hat{x}) \tag{3.37}$$

Therefore,

$$U_v(x,\hat{x}) = C - \sum_i P_a(x_i;y) - E_r(x) \tag{3.38}$$

The degrees of confidence can be computed by the Bayes law

$$w^k = \alpha p(y^k|v^k = 1,\hat{x})P_1 \tag{3.39}$$

where P_1 is the probability of a valid label and

$$p(y^k|v^k = 1,x) = A\exp\left\{-\sum_i P^k(x_i)\right\} \tag{3.40}$$

References

1. M. Kass and A. Witkin and D. Terzopoulos, Snakes: Active Contour Models, International Journal of Computer Vision, Vol. 1, 321–331, 1988.
2. L. D. Cohen, I. Cohen, Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1131–1147, 1993.
3. T. Cootes, T. Hill, C. Taylor and J. Haslam, The Use of Active Shape Models for Locating Structures in Medical Images, Image Vision and Computing, Vol. 12, 355–366, 1994.
4. T. McInerney, D. Terzopoulos, Deformable Models in Medical Image Analysis: A Survey, Medical Image Analysis, Vol. 1, No. 2, 91–108. June 1996.
5. R. Malladi, J. A. Sethian, B. C. Vemuri, Shape Modeling with Front Propagation: A Level Set Approach, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 2, 158–175, February 1995.
6. V. Caselles, R. Kimmel, G. Sapiro, Geodesic Active Contours, International Journal of Computer Vision, Vol. 22, 61–79, 1997.
7. C. Xu, J. L. Prince, Snakes, Shapes, and Gradient Vector Flow, IEEE Transactions on Image Processing, Vol. 7, No. 3, 359–369, March 1998.
8. A. Blake, M. Isard, Active Contours, Springer, 1998.
9. A. Nealen, M. Müller, R. Keiser, E. Boxerman, M. Carlson, Physically Based Deformable Models in Computer Graphics, Computer Graphics Forum, Vol. 25, No. 4, 809–836, December 2006.
10. J. Cheng, S. Foo, Dynamic Directional Gradient Vector Flow for Snakes, IEEE Transactions on Image Processing, Vol. 15, No. 6, 1563–1571, June 2006.
11. J. Gambini, M. Mejail, J. Jacobo-Berlles, A. Frery, Feature Extraction in Speckled Imagery Using Dynamic B-Spline Deformable Contours Under the G0 Model, International Journal of Remote Sensing, Vol. 27, No. 22, 5037–5059, November 2006.
12. A. J. Abrantes, J. S. Marques, A Class of Constrained Clustering Algorithms for Object Boundary Extraction, IEEE Transactions on Image Processing, Vol. 5, No. 11, 1507–1521, 1996.
13. A. J. Abrantes, J. S. Marques, A Constrained Clustering Algorithm for Shape Analysis with Multiple Features, International Conference on Pattern Recognition, 916–919, 2000.
14. J. C. Nascimento, J. S. Marques, An Adaptive Potential for Robust Shape Estimation, Image Vision Computing, Vol. 21, No. 13–14, 1107–1116, 2003.
15. A. J. Abrantes, J. S. Marques, The Mean Shift Algorithm and the Unified Framework, International Conference on Pattern Recognition, 244–247, 2004.
16. J. C. Nascimento, J. S. Marques, Robust Shape Tracking in the Presence of Cluttered Background, IEEE Transactions on Multimedia, Vol. 6, No. 6, 852–861, 2004.
17. J. C. Nascimento, J. S. Marques, Adaptive Snakes Using the EM Algorithm, IEEE Transactions on Image Processing, Vol. 14, No. 11, 1678–1686, 2005.
18. J. C. Nascimento, J. S. Marques, Corrections to Adaptive Snakes Using the EM Algorithm, IEEE Transactions on Image Processing, Vol. 15, No. 3, 788–788, 2006.
19. M. Silveira, J. S. Marques, Estimation of Multiple Objects at Unknown Locations with Active Contours, Iberian Conference on Pattern Recognition and Image Analysis, LNCS 4477, Vol. 2, 372–379, June 2007.
20. J. S. Marques, J. M. Lemos, Optimal and Suboptimal Shape Tracking Based on Multiple Switched Dynamic Models, Image Vision Computing, Vol. 19, No. 8, 539–550, 2001.
21. J. Nascimento, J. S. Marques, Robust Multi-Model Filter for Shape Tracking in the Presence of Outliers, Pattern Recognition, Vol. 35, 2711–2718, December 2002.
22. G. Celeux, J. S. Marques, J. Nascimento, Learning Switching Dynamic Models for Objects Tracking, Pattern Recognition, Vol. 37, No. 9 , 1835–1840, September 2004.
23. A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum Likelihood from Incomplete Data Via the EM Algorithm, Journal of the Royal Statistical Society, B, Vol. 39, 1–38, 1977.

24. S. Kay, Fundamentals of Statistical Processing, Volume I: Estimation Theory, Volume I: Estimation Theory, Prentice Hall, Upper River Saddle, NJ, 1993.
25. Y. Bar-Shalom,T. Fortmann, Tracking and Data Association, Academic, New York, 1988.
26. K. Tugnait, Detection and Estimation for Abruptly Changing Systems, Automatica, Vol. 18, 607–615, 1982.
27. D. Terzopoulos, R. Szeliski, Tracking with Kalman Snakes, in Active Vision, A. Blake and A. Yuille (eds.), MIT Press, Cambridge, MA, Ch. 1, 3–20, 1992.
28. Y. Zhong, A. Jain, M.-P. Dubuisson-Jolly, Object Tracking Using Deformable Templates, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 5, 544–549, May 2000.
29. K. Toyama, A. Blake, Probabilistic Tracking with Exemplars in a Metric Space, International Journal of Computer Vision, Vol. 48, No. 1, 9–19, 2002.
30. D. Comaniciu, X.S. Zhou, S. Krishnan, Robust Real-Time Tracking of Myocardial Border: An Information Fusion Approach, IEEE Transactions Medical Imaging, Vol. 23, No. 7, 849–860, 2004.
31. S. Dambreville, Y. Rathi, A. Tannenbaum, Tracking Deformable Objects with Unscented Kalman Filtering and Geometric Active Contours, 14–16, June 2006.
32. Y. Rathi, N. Vaswani, Allen Tannenbaum, Anthony Yezzi, Tracking Deforming Objects Using Particle Filtering for Geometric Active Contours, to appear in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007.
33. F. Moreno-Noguer, A. Sanfeliu, D. Samaras, Integration of Deformable Contours and a Multiple Hypotheses Fisher Color Model for Robust Tracking, Image and Vision Computing, Vol. 25, 285–296, 2007.

Chapter 1
Modeling Cardiovascular Anatomy from Patient-Specific Imaging

Chandrajit Bajaj and Samrat Goswami

1.1 Introduction

The importance of modern imaging techniques for capturing detailed structural information of a biological system cannot be understated. Unfortunately images do not reveal the "full functional story" and a spatially realistic computer model is often necessary for a comprehensive understanding of the complicated structural and physiological properties of the biological system's entities under investigation [1]. Deeper insights into structure-to-function relationships of different entities is achieved via finite element simulations of the modeled biomedical process. A 3D (three dimensional) finite element meshed computer model of the biological system is therefore a first step to perform such simulations.

The behavioral attributes of a biological entity or the physiological interaction between different participating components of a biological system are often modeled mathematically via a coupled set of differential and integral equations, and quite often numerically evaluated using finite element (or boundary element) simulations. To further emphasize the premise of cardiac modeling from imaging data, we state a few computational biomedical modeling and simulation examples: 3D computational modeling of the human heart for a quantitative analysis of cyclical electrical conductance on the heart membrane [2–6]; the biomechanical properties (stress-strain, elasticity) of the heart ventricular walls [7–12]; 3D modeling and simulation of pulsatile blood flow through human arteries/veins for vascular by-pass surgery pre-planning on a patient specific basis [13–18]. A finite element decomposition of the geometric domain, capturing the detailed spatial features that can be gleaned from the imaging, is therefore the essential first step toward performing the necessary numerical simulations [19–22].

C. Bajaj and S. Goswami
Computational Visualization Center, Institute of Computational Engineering and Sciences, University of Texas, Austin, Texas 78712
e-mail: bajaj@cs.utexas.edu; samrat@ices.utexas.edu

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

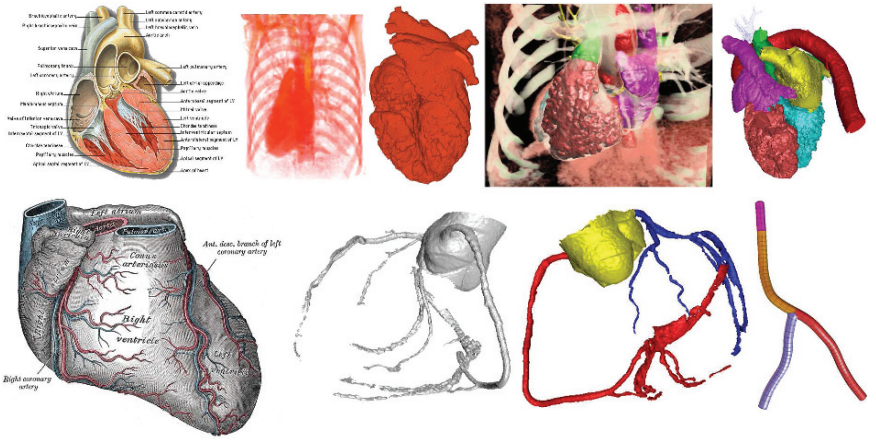


Fig. 1.1 Cardiovascular Modeling from Imaging Data. Top row illustrates the modeling of patient heart from imaging data. From left to right: an illustration of internal structure of human heart (Courtesy [23]), volume visualization of a typical Computed Tomography (CT) reconstructed imaging volume, the extracted geometric model of a patient heart from the rest of the thoracic bone and tissue and the annotated (multi-colored) segmentation of the same into biologically meaningful components, namely Aorta, Pulmonary Artery, Left and Right Ventricle and Atrium (colored differently). Bottom row illustrates the modeling of coronary artery. From left to right: an illustration of annotated coronary arterial tree (Courtesy http://en.wikipedia.org/wiki/Coronary_circulation), a geometric model extracted from CT volumetric imaging, segmentation into left and right subtree (colored red and blue), and the NURBS model of a small portion that was used for Isogeometric analysis of blood flow in [24]

Modeling of human vasculature from three-dimensional (3D) Computed Tomography (CT) images of the thorax is a critical step for computer-aided diagnosis (CAD) in disease domains such as lung nodules [25], coronary artery disease [26], and pulmonary embolism (PE) [27]. Even though there are many published approaches, the problem is still unsolved. Survey of various techniques on this topic can be found in [28,29] (Fig. 1.1). Thoracic CT angiography (CTA) imaging is often performed for patients suspected of having a PE that is defined as a thrombus (or a clot of blood) [30]. Therefore, in order to detect pulmonary embolism, a suitable model of vasculature distnguishing between arterial and venous blood vessel trees is a crucial step.

In subsequent subsections we highlight the computational pipeline, the main algorithmic components and a few descriptive results of our Cardio Vascular Modeling from Imaging software.

1.2 Data Processing

The Imaging-to-Modeling software system for cardiovascular data employs both Image Processing and Geometry processing functionalities to produce a suitable linear or higher order meshed model of the anatomy. Figure 1.2 describes the data-flow layout. We describe the major algorithmic components of each of the processing

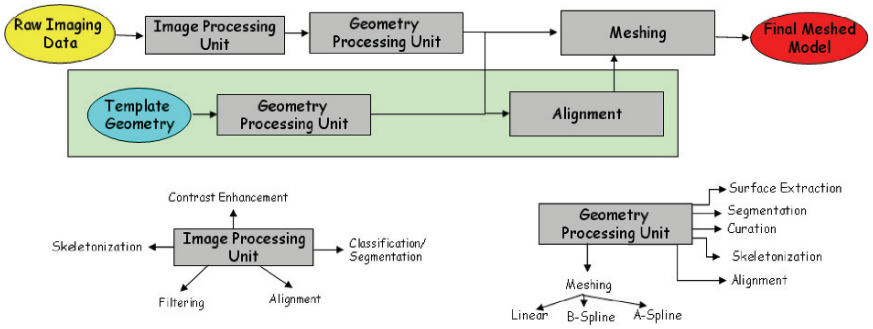


Fig. 1.2 Data Flow of Cardio-Vascular Modeling from 3D Imaging Data: There are two major data processing algorithmic modules – Image Processing and Geometry Processing. The Image Processing module consists of sub-modules for *Contrast Enhancement*, *Classification/Segmentation*, *Filtering*, *Skeletonization* and *Alignment*. The Geometry Processing module consists of *Surface Extraction*, *Curation*, *Segmentation*, *Skeletonization*, *Alignment* and *Meshing*, which itself is further subdivided into linear and higher order boundary and finite element mesh generation components. The 3D CT Imaging data is first passed through the Image Processing module for improvement of image quality which is then processed by the Geometry Processing module for extraction of a clean geometry annotated with the present features. Finally the clean geometry is converted to a linear or higher-order mesh. Occasionally, to deal with incomplete or low quality imaging data, a twin data processing pipeline is employed where a template geometry is processed to extract vital geometric information which is then passed to robustly model patient specific anatomical model from low-quality imaging data

modules in Sections 1.2.1 and 1.2.2. The reader must note that, the modules are selectively used depending on the nature and quality of the input imaging data (Section 1.3).

1.2.1 Image Processing

(A) Contrast Enhancement:

The three dimensional intensity data often possesses a low contrast between structural features and the background, thereby making further processing all the more difficult. Image contrast enhancement is a process used to improve the image quality for better visual appearance for subsequent operations. The most commonly used methods utilize global contrast manipulation based on global [31, 32] or local histogram equalization [31–34], *retinex model* [35, 36] and wavelet decomposition [37, 38].

We have developed a fast method for image contrast enhancement [39] based on a localized version of the classical contrast manipulation [31, 32]. In this method, we design an adaptive transfer function for each individual voxel, based on the intensities in a suitable local neighborhood. First, we compute the local statistics (local average, local minimum, and local maximum) for each voxel using a fast propagation scheme [40, 41]. Then a transfer function is

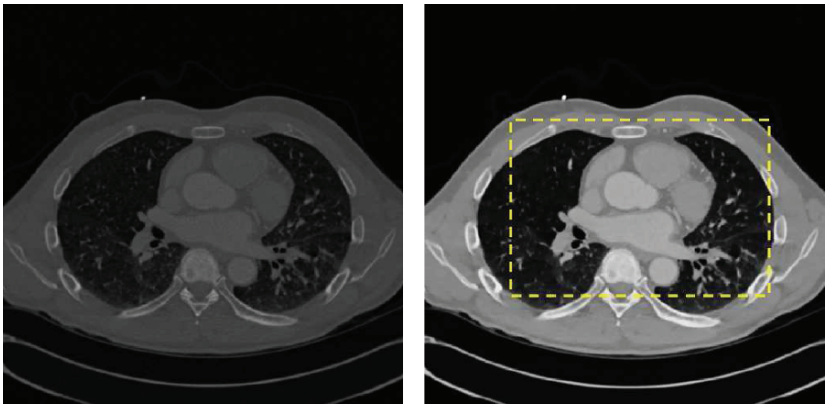


Fig. 1.3 The performance of the contrast enhancement algorithm is shown for one slice of the CTA data

designed based upon the calculated local statistics. Various linear or nonlinear functions can be used here to stretch the contrast profile. We build a transfer function which consists of two pieces: a convex curve in the dark-intensity range and a concave curve in the bright-intensity range. The overall function is C^1 continuous. Finally, we map the intensity of each voxel to a new one using the calculated transfer function. The performance of this algorithm is shown in Fig. 1.3.

(B) Filtering:

The input images are often contaminated with noise and are therefore need to be filtered. Traditional image filters include Gaussian filtering, median filtering, and frequency domain filtering [31]. Compared to these, anisotropic filters are preferred as they tend to preserve the features better. Bilateral filtering [42–45] is a straightforward extension of Gaussian filtering by simply multiplying an additional term in the weighting function. Partial differential equation (PDE) based techniques, known as anisotropic geometric diffusion, have also been studied [46, 47]. Another popular technique for anisotropic filtering is by wavelet transformation [48]. By carefully designing the filter, one can smooth image noise while maintaining the sharpness of the edges in an image [49]. Finally, the development of nonlinear median-based filters in recent years has also produced promising results [50, 51]. Among the aforementioned techniques, two methods for noise reduction have been suggested for tomographic data sets, namely wavelet filtering [52] and non-linear anisotropic diffusion [53] (Fig. 1.4).

Our approach, utilizing a bilateral pre-filtering coupled with an evolution driven anisotropic geometric diffusion PDE (partial differential equation), has shown significant results in enhancing the features of intensity maps. The PDE model is :

$$\partial_t \phi - \|\nabla\phi\| div\left(D^{\sigma}\frac{\nabla\phi}{\|\nabla\phi\|}\right) = 0$$

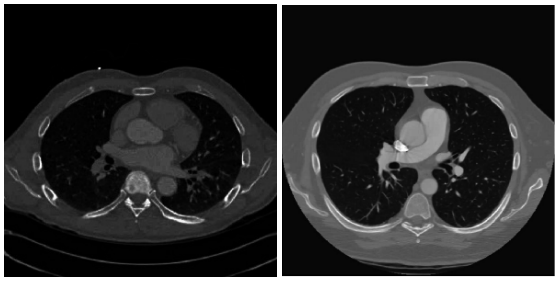


Fig. 1.4 The performance of the bilaterally pre-filtered (left) and anistropic diffusion filtered (right) algorithm is shown for one slice of the CTA data

The efficacy of our method is based on a careful selection of the anisotropic diffusion tensor D^{σ} based on estimates of the normal and two principal curvatures and curvature directions of a feature isosurface (level-set) in three dimensions [54–56]. The diffusivities along the three independent directions of the feature isosurface are determined by the local second order variation of the intensity function at each voxel. In order to estimate continuous first and second order partial derivatives, a tricubic B-spline basis is used to locally approximate the original intensity.

(C) Classification/Segmentation:

Voxel Classification:

The Fuzzy C-Means (FCM) algorithm [57] and the Expectation Maximization (EM) algorithm [58] have been used for soft clustering in data-mining and image classification. Pham et al. proposed an Adaptive Fuzzy C-Means (AFCM) algorithm to classify inhomogeneous medical images and volume datasets [59, 60]. Ahmed et al. proposed a bias corrected FCM algorithm to compensate inhomogeneities of images of volume datasets [61, 62]. Each of these algorithms minimizes an objective function through iterative methods. Gopal et al. proposed a maximum likelihood estimate algorithm with a Spatially Variant Finite Mixture model (SVFMM) for image classification [63]. Laidlaw suggested a partial-volume Bayesian classification algorithm based on Bayes theorem [64].

Our goal of 3D map segmentation is to partition the map into a number of connected regions of interest. We have compared and implemented several classification algorithms [64–67]. We first locate the seed points by gradient vector diffusion [68]. We then compute the min-max range of every seed point's neighbors and cluster the seed points to belong to the same region if the min-max ranges overlap. We then apply GVF snake [69, 70] to cluster the voxels falling into separate regions. In addition, the contour spectrum is used to identify the number of materials in the image and is also used to select critical isovalues based on volume fraction of the material [71]. We have also developed a multi-dimensional signature based voxel classification

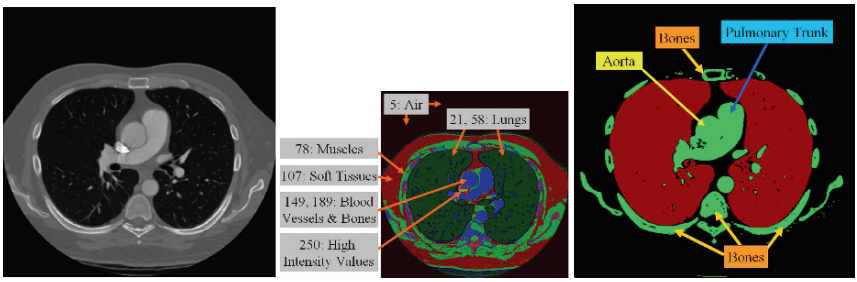


Fig. 1.5 Voxel Classification: a single slice of CTA image of a patient (a) is classified to tag voxels belonging to different anatomical regions (b). The intensity values are also marked (b). The final classified voxelized image is shown with most important regions (c)

scheme [72] appropriate for medical imaging data. Figure 1.5 shows the results of the classification on a single slice of the patient scan where the voxels have been classified into the background, lungs and vasculature.

Segmentation via Fast Marching Method:

Segmentation is a way to electronically dissect the significant biological components, and thereby obtain a clear view into the machinery's architectural organization [73]. Segmentation is usually carried out either manually [74–78] or semi-automatically [79, 80]. Current efforts on the decomposition still largely rely on manual work with an assistance of a graphical user interface [81,82]. Manual segmentation can be tedious and often subjective [76,83]. Automated segmentation is still recognized as one of the hardest tasks in the field of image processing although various techniques have been proposed for automated or semi-automated segmentation. Commonly used methods include segmentation based on edge detection, region growing and/or region merging, active curve/surface motion and model based segmentation. In particular, two techniques were discussed in details in the electron tomography community. One is called *water-shed immersion method* [79] and the other is based on normalized graph cut and eigenvector analysis [80].

In [84–86] we adopted a variant of the fast marching method [87–89]. In this method a contour is initialized from a pre-chosen seed point, and the contour is allowed to grow until a certain stopping condition is reached. Traditionally this method is designed for a single object segmentation. We present an approach based on an idea of "re-initialization" by simply regarding and classifying the critical points as seeds. Every seed initiates one contour and all contours start to grow simultaneously and independently. We further classify the critical points into clusters and merge the growing contours which are initiated by the critical points in the same cluster. This multi-label idea has been used elsewhere (e.g. [90]), but the detection and classification of seeds are different in our approach. Figure 1.6 shows the process of segmentation, namely the seed point

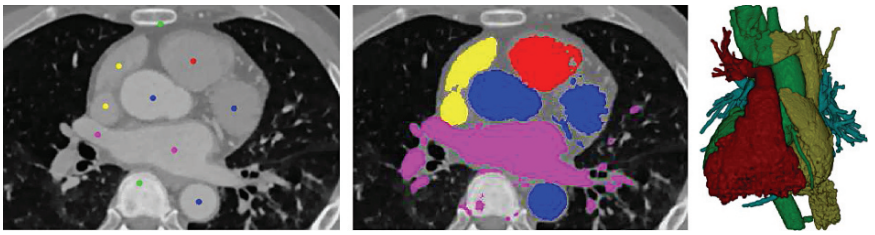
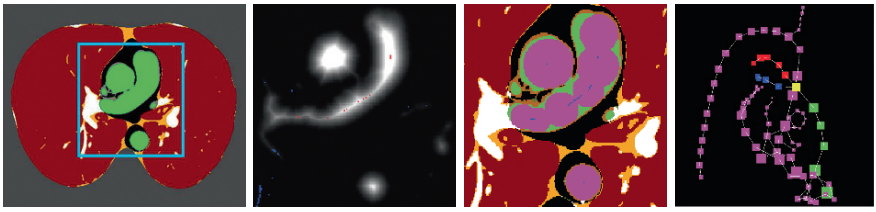


Fig. 1.6 Segmentation via Fast Marching Method: leftmost subfigure shows a single slice of the CTA image of a patient along with manually placed seeds for each of the main compartments of the human heart to be identified. The middle subfigure shows the segmentation on a single slice colored by segmented regions. The rightmost subfigure shows the three dimensional segmented model of heart via isosurface rendering. The different colors represent left and right ventricle and aorta

classification and region growing on a single slice of the image followed by the final segmented three dimensional model of human heart from patient imaging data.

(D) Skeleton Extraction:
Extraction of skeletal description often leads to a complexity-reducing better understanding of the image as it amplifies lower dimensional key features present in the data. Image based skeletonization algorithms are abound in the field of image processing. The previous efforts on this topic by other authors can be categorized into three approaches – one based on isotropic diffusion, governed by a set of linear PDEs [91,92], one using scale-space theory [93–95] and one based on pseudo-distance map [96].
We have developed two distinct approaches to extract skeletons from imaging data which offer robustness and efficiency. First approach is based on the critical point structure of the imaging data. We first compute the gradient vector field at every voxel of the imaging data. Because of the noise, the vector field is somewhat arbitrary and do not carry much useful information. To bring out the underlying structure, we then apply gradient vector diffusion (GVF). The resulting vector field is then analyzed to detect the critical points and these critical points are joined to form a skeletal structure of the foreground of the image. The details of the process is given in [56].
The second approach for skeletonization starts with an isosurface and the distance function induced by it [28]. It then places a sequence of inner medial balls in a greedy fashion, capturing as much inner volume as possible. A neighborhood graph, based on the intersection pattern of these inner medial balls, is then constructed which provides the one dimensional skeletal structure. Figure 1.7 shows an example result of this algorithm applied on the CTA data of human heart.

(E) Flexible Alignment:
Image registration is a commonly employed to flexibly match two different instances of a biological structure. In the context of cardiovascular modeling,



A.1 A.2 A.3 A.4

Fig. 1.7 Skeletonization: a slice of the imaging data, its distance map and the extracted skeletal graph using the algorithm in [28] are shown

the problem is to fit one image of the anatomy in question with its counterpart observed at a different time, or imaged from a different patient. Based on the transformation model that is to be applied on one instance (source) to fit the other (target), image resgistration is primarily of two types – Affine and Elastic/Deformable.

In case of affine registration, the relationship between the source and target image is established via a set of transformation parameters, and then those parameters are estimated by minimizing a quadratic error function. Algoithms under such approaches can be found in [97–99]. We have developed an algorithm in [100] which exploits the non-equispaced Fourier transformation techniques [101, 102] to speed up the affine image registration process.

The task is far more challenging when deformation of one image needs to be taken into account to properly fit it into the other image. For a nice survey on affine and deformable image registration algorithms, see [103]. However this survey is old, and since its publication many other algorithms have been published. Bajcsy and Kovacic modeled the elastic image registration problem by the deformation of elastic plates [104]. Christensen et al. considered the deforming image to be embedded in a viscous fluid whose motion is governed by Navier-Stokes equation [105]. Based on a similar viscous fluid registration scheme, Yanovsky et al. designed a new energy function introducing Jacobian maps [106] and this method was shown to perform better than [105] in terms of converegence and stability. There are other level-set based methods, e.g. by Clarenz et al. [107], and via edge matching technique by Mumford and Shah [108].

1.2.2 Geometry Processing

(A) Surface Extraction:

Geometry extraction from three dimensional volumetric data is a primary step toward further analysis. There are several approaches for accomplishing this task.

Contouring:

Isosurfacing is a popular method to extract surface geometry from scalar volume containing intensity values of the scanned anatomy. There are typically two types of contouring method frequently used in the literature – Primal Contouring and Dual Contouring. The most widely used primal contouring technique is the Marching Cubes Method [109] which extracts the geometry in a piecewise fashion by visiting every voxel of the volume data. There are several improvements of this technique that has been reported since the first appearance of the algorithm [110]. Dual contouring technique is similar to primal contouring in a sense that it also extracts the isosurface in a piecewise fashion. However it samples every voxel, instead of every edge of the voxel, to better approximate possible sharp features in the extracted geometry. We have experimented with both techniques for extracting a geometry from Cardiac CT data and we have seen that they perform similarly without rendering any significant advantage of any technique over the other. Figure 1.8 shows sample isocontours extracted from CTA imaging data of human heart.
Other than isocontouring, one can also apply level set based methods where a seed is grown to capture the boundary of a region in the image based on the intensity values. In literature, such technique is commonly known as *snake* [69, 70]. Note, this is similar to the image segmentation technique described earlier.

Point Cloud Reconstruction:

Both of these approaches are very sensitive to the noise present in the data and especially isosurfacing techniques suffer when the imaging data is inhomogeneous. To circumvent these problems we adopt a third approach which is based on scattered data interpolation. After performing image segmentation on

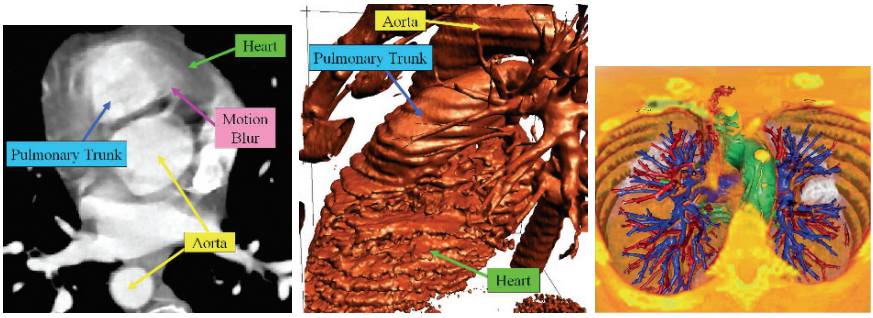


Fig. 1.8 Contouring: a single annotated slice of human cardio-vasculature and the geometry extracted from the imaging data via contouring (isosurfacing) are shown from left to right. In the rightmost subfigure, the heart of the patient is shown in green while the blood vessel tree is color coded by red and blue depending on if a branch is an artery or a vein

the CT image, we obtain a set of voxels belonging to every region of interest. We then apply the point based reconstruction technique to extract the geometry from the cloud of boundary voxels.

The point based reconstruction technique has been researched extensively in the last decade. We refer the readers to some recent surveys for prior work in this area, e.g. [111]. We have adopted two recent techniques for our purpose of reconstruction – *TightCocone* and *RobustCocone*. *TightCocone* algorithm by Dey and Goswami [112] reconstructs a watertight triangulated surface from possibly undersampled input point cloud. A variant of this algorithm, called *RobustCocone*, was also developed in 2004. This algorithm is particularly suitable for noisy data. In our case, we often encounter noise in the segmented image even after applying image segmentation techniques, and to tackle such cases, we use *RobustCocone* for a reconstruction of the geometry. Figure 1.9 shows the results of surface reconstruction on the pointsets sampling compartments of human heart.

Surface reconstruction from scattered data has also been approached using variational approach. These techniques typically formulate an energy function based on the input data points and try to extract a surface that minimizes that energy. Such approach was first advocated by Zhao et al. in [113] who formulated the energy function as the integral of the distance function weighted by the area element of the input set of primitives for which a surface needs to be fit. Then they evolved an initial guess using a convection based approach.

We adopt an approach based on higher order level set spline (HLS) method. Given a non-negative energy function $g(\mathbf{x})$, the surface Γ is defined to be the one that minimizes the energy function $E(\Gamma) = \int_\Gamma g(\mathbf{x})d\mathbf{x} + \varepsilon \int_\Gamma h(\mathbf{x},\mathbf{n})d\mathbf{x}$. Given a input set of points, one then formulates this energy using the distance function and evolves an initial approximation to guide the evolution so that the resulting deformation minimizes the energy. Details of this method can be found in [114].

(B) Curation/Filtering:

The cardiovascular geometry extracted from imaging data typically has topological anomalies, namely small components, spurious noisy features etc. Therefore a careful investigation and subsequent removal of the spurious features present in the data is essential. Following are different scenarios.

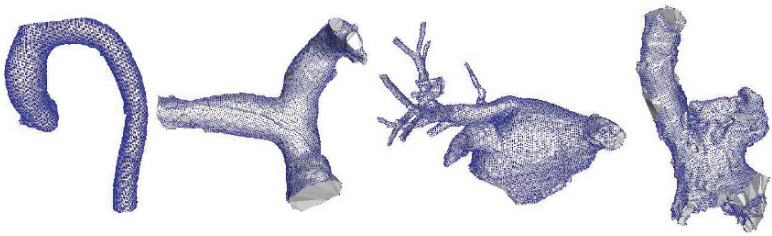


Fig. 1.9 Surface Reconstruction from Point Cloud Data: major components of the human heart are reconstructed using the voxels surrounding the boundaries of the individual regions

Regularization:

Geometry from volume data is often reconstructed via image segmentation. However the set of voxels segregated from the imaging data does not always conform with the true surface topology. As a result, we encounter subsets of voxels which do not sample a two dimensional manifold. Therefore it is important to recognize the dimension of the underlying space of the voxels marked by the segmentation process and remove the spurious ones. To perform this task, we use the technique described in [115]. A similar Voronoi-based approach was also reported in [116]. Following [115] we first construct a k-neighborhood graph. Then for every point we collect the neighboring points and perform Principal Component Analysis (PCA) on that subset. The eigenvalues of the covariance matrix determines the underlying dimension of the manifold. More precisely if all the eigenvalues are almost equal, the voxel is inside the segmented region and the point samples a three dimensional manifold. If two eigenvalues are almost equal and one is much smaller than the other two, the voxel lies on the boundary surface of the segmented region and is a true candidate for subsequent geometry reconstruction. Finally if two of the eigenvalues are much smaller than the third one, then the voxel samples a dangling one dimensional strand and such voxels must be removed.

Volumetric Feature Quantification:

Given a set of points P sampling the entire shape, possibly contaminated with topological artifacts like small connected components and thin tunnels, we synthesize a distance function h_P which assigns every point in $\mathbb{R}^3$ the distance to the nearest sample point in P. There are four types of critical points of h_P, namely maxima, index 2 saddles, index 1 saddles and minima. It was shown that these critical points can be detected efficiently using the duality of Voronoi and Delaunay diagram of the original pointset P [117]. It was further shown that the stable manifolds of the maxima interior and exterior to the shape contains useful information connecting to the primal and complementary feature space of the shape [118, 119]. Figure 1.10 shows how the bone, ribs and thin blood vessels are removed via volumetric feature quantification process since they are not essential for creating a suitable model of human heart.

(C) Segmentation:

Based on the critical points of the distance function induced by the input geometry, we perform the geometry segmentation as follows. The detail of the algorithm is given in [118].

The geometric shape is given by a set of points P sampling the shape. The *feature* of the shape is then defined in terms of the stable manifold of the maxima of the distance function h_P. The maxima are first computed by identifying the Voronoi vertices which lie inside their dual tetrahedra. Applying a Delaunay-based reconstruction technique on the pointset, one can further classify the tetrahedra holding the maxima into *inside* or *outside*. For our purpose we use only the *interior* maxima. We compute the stable manifold of these

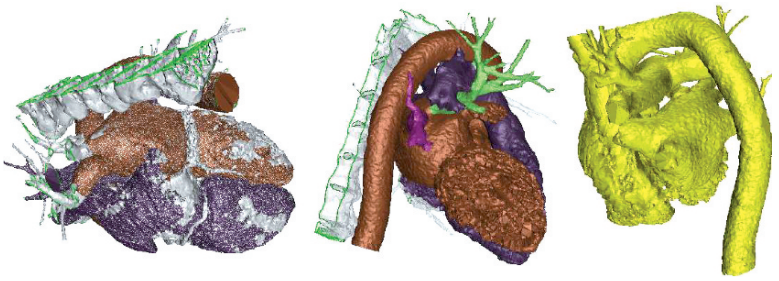


Fig. 1.10 Geometry Curation: geometry of human heart extracted from the imaging data, which is cluttered with bone and other unnecessary parts, is cleaned up using the curation process

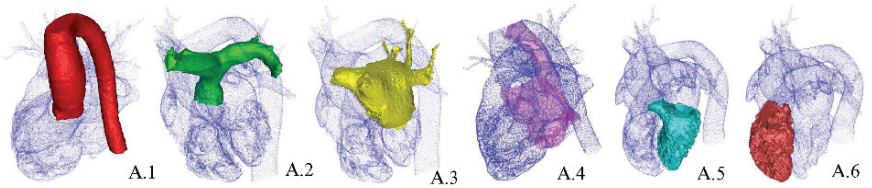


Fig. 1.11 Geometry Segmentation: different components – Aorta (A.1), Pulmonary Artery (A.2), Left Atrium (A.3), Right Atrium (A.4), Left Ventricle (A.5) and Right Ventricle (A.6) are extracted from the point sample of the whole heart

maxima using the algorithm in [118]. The adjacent stable manifolds are then merged if the generating maxima have almost same value of h_P as measured by a parameter δ. Figure 1.11 shows the performance of this algorithm on the cardiovascular geometries. In this figure, the six main components of a human heart, namely aorta, pulmonary artery, left and right atrium and ventricle are segmented out from a set of points sampling the boundary of heart.

(D) Skeletonization:

Computing skeletons of a geometric shape is a research issue that has been around for a long time. Medial axis cite (Blum) is considered a standard skeletal description of a shape and there are algorithms [120, 121] and publicly available software to compute the Medial Axis Transform (MAT) of a shape from its pointsample [122]. However, medial axis is composed of planar (two dimensional) and linear (one dimensional) parts. In order to compute such dimension-dependent decomposition of the medial axis, as well as pruning away hairy branches from the medial axis one requires some extra gadgets which we describe in the next paragraph. There are some previous works which focussed on computing a linear skeleton of an arbitrary shape. Some of these are topological thinning [123], distance field based methods [124–127], potential field based methods [128], thinning via medial geodesic function [129] and others [130–132]. Cornea et al. [133] give a comprehensive survey of these techniques.

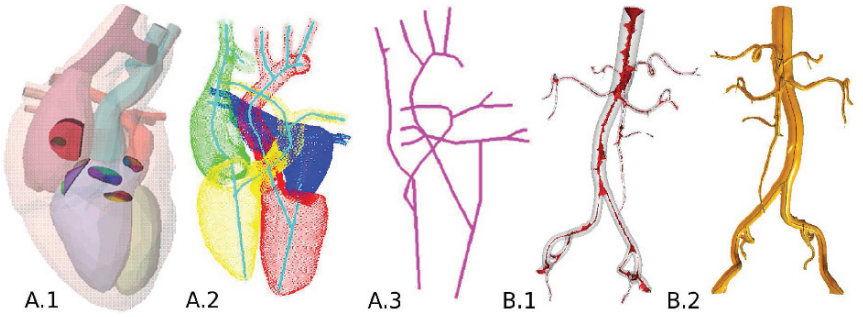


Fig. 1.12 Skeletonization: (A). Skeleton of the template geometry is extracted. (B). The skeletal structure of the abdominal aorta is extracted. Note, compared to the medial axis (B.1), the unstable manifold of index 2 saddles is a much cleaner one dimensional skeletal structure

Once a suitable description of the geometry is obtained either by reconstruction or by contouring, the distance function based approach is used to compute a skeletal structure of the shape. As for segmentation, the critical points of the function h_P is computed for a set P of input points sampling the shape. The index 1 and index 2 saddle points are then detected using the Voronoi-Delaunay duality [117]. To generate the skeletal structure, we then compute the unstable manifold of these critical points. The unstable manifold of an index 2 saddle point (U_2) is one dimensional and the unstable manifold of an index 1 saddle point (U_1) is two dimensional. Moreover, every U_1 is bounded by some U_2's. The details of the computation of U_1 and U_2 are given in [134]. Figure 1.12 shows two instances where the skeletal structures have been constructed using this method.

(E) Alignment:

Alignment of two similar but not identical geometric objects is a difficult problem. There are relatively few papers in the geometric modeling community that address this problem. Recently, an interesting technique was reported by Eckstein et al. [135] where generalized surface flow was used for non-rigid alignment of a template geometry into the patient data. The authors design an energy function based on pseudo-Hausdorff distance between the two geometries and evolve the template geometry to fit the patient geometry following the gradient of the energy function.

We experimented with a different approach for non-rigid fitting of the segments of the template heart into the patient data in order to inherit the correct topological structure from the template geometry as well as retrieve the missing information in the patient data. We construct a skeletal description of different parts of the template geometry as described earlier. Every segment is then described as a union of balls centered at the one dimensional skeleton following a popular approach due to [136]. A mass-spring network is then built where each ball's mass is proportional to its radius and the spring constant is taken to be constant. The normal mode analysis (NMA), which is a popular way to

depict the vibrational nature of a molecule [137], is then applied to the network which produces a spectrum of possible deformed shapes. We choose one deformed shape from the spectrum that best fits the patient geometry.

(F) Quality Meshing:
The final goal of this Imaging to Modeling software system is to produce suitably discretized meshed model of the imaged biological entity. The task of meshing is primarily divided into two parts – Boundary Element and Finite Element. Each of Boundary Element and Finite Element meshing again has three sub-parts.

Boundary Elements:

Boundary Element meshing refers to the meshing of the surface geometry. Depending on the smoothness and shape of every patch forming the surface mesh.

(a) Triangle/Quadrilateral Elements: Given a CT image, the task is to compute a triangulated or quadrangulated discretization of the boundary of the cardio-vascular anatomy. In a sense the task is similar to contouring the zero-set of the original intensity function or a distance function induced by the reconstructed geometry. There are numerous algorithms to accomplish such tasks as mentioned in the previous subsection. However in this step our goal is to build a boundary element mesh of superior quality than what is typically output by the contouring routines. The mesh quality metrics are described in [138].
We approach the problem in two steps. First we apply the dual contouring method proposed by Ju et al. [139] and apply geometric flow to improve the quality of the surface mesh. Figure 1.13 shows the performance of the boundary element meshing algorithm.

(b) B-Spline Elements: Building B-Spline model for free-form geometric objects has been an active area of research in the past. Given a triangulated or quadrilateral surface mesh, there have been numerous approaches

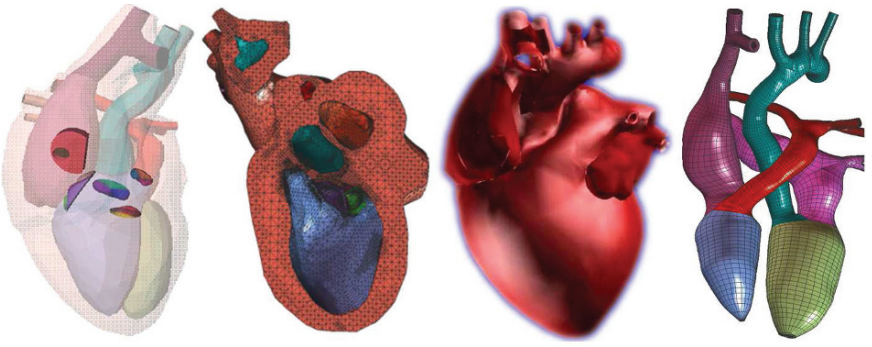


Fig. 1.13 Meshing: triangle-tetrahedral, A-patch, NURBS meshes

to build a smooth network of B-Spline patches to model the underlying surface [140]. Subdivision based methods, namely Catmull-Clark [141], Doo-Sabin [142, 143] and Loop subdivision [144] for producing quadratic and cubic B-Spline models from triangulated and quadrilateral control meshes have also gained popularity due to their simplicity. However in most of the cases, it is not straight-forward how to generate the initial control mesh for free-form shapes especially when the input is a set of scattered set of points or a set of voxels representing the boundary of a segmented region inside a three dimensional volume.

Until recently there were very works that dealt with the problem of building a control quadrilateral mesh from a free-form geometry of arbitrary topology [145, 146]. Recently there have been substantial research works in the computer graphics community that tackled the problem of quadrangulating a surface mesh following the intrinsic anisotropy of the geometry [147–149]. Following these approaches, one can build a quadrilateral base surface mesh on which any of the standard subdivision scheme can be applied to build the desired B-Spline model.

(c) A-Patch Elements: The linear meshes do not always provide the adequate smoothness for them to be effectively used further. This is the reason a higher order meshed description of the geometry is often desired. Bajaj et al. presented a solution for this problem in 1995 where they devised a scheme that takes a triangulated surface mesh and builds a higher order (cubic) patch complex to describe the same surface. Under certain conditions, they also showed that these patches meet in a C^1 continuous manner. They called this patched description of a two dimensional geometry an algebraic surface patch or an *A-patch* [150].

The construction of A-patches proceeds as follows. First a triangle mesh is created that linearly approximates the given geometry. A tetrahedral scaffolding is then built around it so that the triangle mesh lies inside it. By assigning a suitable weight at every node of the scaffold, a polynomial inside each tetrahedron is constructed in such a way that its zero set satisfies certain properties that makes the higher order patch inside the tetrahedron free of singularity. Moreover the pacthes are derivative-continuous across the tetrahedra. The patch complex then provides a higher order smooth description of the geometry. Details of this method can be found in [150]. This method of construction of surface A-patches has also been generalized to hexahedral elements, as well as to the multi-resolution construction of shell finite elements [151, 152]. Figure 1.13 shows a higher order geometric model of human heart.

3D Finite Elements:

Finite Element meshing refers to the techniques of producing the discretization of the volume enclosed by the surface geometry.

(a) Tetrahedral/Hexahedral Elements: There have been prior works done on building volumetric (finite element) tetrahedral and hexahedral meshes from imaging data. For a detailed survey on the prior works see [153].
We have developed two algorithms for such purposes – TetMesh and HexMesh. The approach is as follows. First we extract a correct boundary using the boundary element meshing. For tetrahedral mesh generation we use the triangulated boundary and for hexahedral mesh generation we use the quadrangulated boundary meshing scheme as described above. Next, we design a series of templates to build a solid tetrahedral or hexahedral mesh from that conforms with the boundary mesh. The details about the templates are given in [138, 154]. As with the boundary element mesh, we further improve the quality of the finite element mesh by applying geometric flow [155]. Figure 1.13 shows the tetrahedral mesh cut-away for a template human heart.

(b) B-Spline Elements: As the most highly developed and widely utilized technique, NURBS (Non Uniform Rational B-Splines) [156–158] has evolved into an essential tool for a semi-analytical representation of geometric entities. Sometimes NURBS solid models are taken as input for finite element mesh generation [159]. Anderson et al. proposed a fast generation of NURBS surfaces from polygonal mesh models of human anatomy [160]. An enhanced algorithm was developed for NURBS evaluation and utilization in grid generation [161]. In isogeometric analysis [162], NURBS basis functions are used to construct the exact geometry, as well as the corresponding solution space.
We have developed a skeleton-based approach for building NURBS model of vasculature. Using the skeletonization approach described in the earlier paragraph, a one dimensional polylinear skeletal structure is first extracted from the tubular geometry of vasculature. Then we design a set of templates that builds a hexahedral mesh around the skeleton. The details about the templates can be found in [24]. The hexahedral mesh is used as the control mesh for further NURBS mesh generation. Figure 1.13 shows the NURBS model of the inner blood volume of human heart.

(c) Shell A-Patch Elements: Shell structures appear frequently in biological entities. The muscle wall of heart and blood vessels are perfect examples of such surfaces. It is desired to model such *fat* surfaces with desired smoothness. Bajaj et al. presented algorithms that model smooth shell structures using shell A-patch finite elements [151, 152].
The algorithm takes a pair of triangle mesh as input where the correspondence between the triangle meshes is implicit. It then decimates both the meshes simultaneously to build a coarser representation and occasionally merges the adjacent triangles to form quadrilaterals wherever possible. Using the correspondence between the triangles and quadrilaterals in the two-sheeted surface, the interval volume is then filled with 3-sided (triangular) and 4-sided (quadrilateral) prisms. A C^1 piecewise trivariate function is then constructed over this collection of prisms. The range of the

function varies from -1 to 1. For any scalar $\alpha \in [-1,1]$, the 0-set thus gives the higher order approximation of the intermediate surface within the shell element. When α equals -1 or 1, the resulting patch gives a higher order approximation of the inner and outer boundaries of the shell.

1.3 Implementation Results

We exhibit the implementation results of our image and geometry processing algorithms on modeling the Heart and Vasculature (Coronary Artery, Pulmonary Arterry, Abdominal Aorta and Thoracic Aorta).

1.3.1 Heart

We have experimented with images of patient hearts. For illustration of the performance of the algorithms, we have selected two datasets one of which is a low quality image whereas the other is of relatively better quality.

The first heart dataset is of dimension $512 \times 512 \times 432$ and the spacing in x, y, z directions are respectively 0.390625, 0.390625, 0.3 mm. We first applied the contrast enhancement to the original image and then applied the fast marching based segmentation on the contrast-enhanced image to separate the subvolumes corresponding to the aorta, pulmonary artery, right atrium and left atrium. Because of the poor quality of the image, the ventricles could not be recovered well. The result of the image processing on this dataset is shown in Fig. 1.14B. After the four components of the patient heart are segmented from the volume, we took the boundary voxels of each of these four regions and applied surface reconstruction from scattered points to build the initial triangulated geometric models. As one can see these models have a reasonable amount of spurious parts as well as some missing information that could not be retrieved from the patient data. We analyzed each of the recovered geometries using the critical point structure of the distance function described in Section 1.2.2. Using curation and geometry segmentation, we identified the portions which most prominently correspond to the portions of a template heart model while pruning away the undesired portions. Figure 1.14C shows the relevant portions that have high correlation with the corresponding portions of the template geometry. Parallel to the processing of the patient heart, we also performed geometric analysis of the template heart model. We construct the one dimensional skeletal structure of all the six components of the template, as well as we segmented the template into aorta, pulmonary artery, right and left ventricle and atrium (Fig. 1.14D). From there we could draw a correspondence between the segmented portions of the patient heart with the template heart (Fig. 1.14E).

The second dataset was of better quality in terms of contrast and noise present. We first extracted the geometry via isocontouring. However using a single isovalue could not entirely serve the purpose as it also extracts the surrounding vasculature

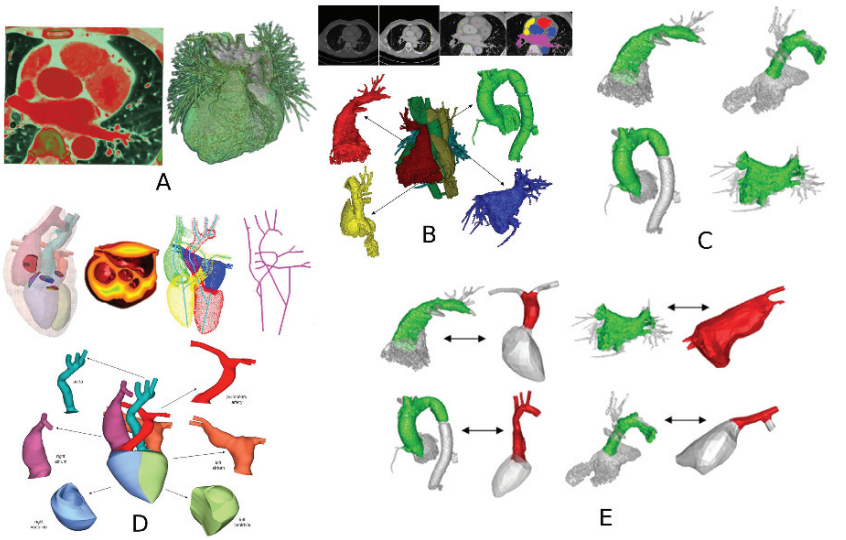


Fig. 1.14 Result: heart I. (A) One slice of the original data and isocontour-enhanced volume rendering of the input. (B) Due to relatively poor quality, the imaging data is first passed through the image processing unit that enhances the contrast and segments the volume into major components using fast marching based image segmentation. (C) The extracted geometry from the segmented image is then curated to keep only the (green) portions which can be matched with the corresponding portions from the template heart model (D, E)

as shown in Fig. 1.15A. We therefore resorted on curation to extract just the geometry of heart by pruning away the surrounding thin blood vessels using curation (Fig. 1.15B). After this step we were left with the geometry of heart which we further segmented using the stable manifold of the maxima of the distance function induced by the geometry. The segmentation step was crucial as it was able to separate the six main components of the patient heart, namely aorta, pulmonary artery, right and left ventricle and atrium as shown in Fig. 1.15C. We were then able to draw the correspondence between each of the segmented parts with the corresponding ones from the template geometry as shown in Fig. 1.15D.

1.3.2 Vasculature

1.3.2.1 Pulmonary Artery

The primary objective behind modeling pulmonary artery was to detect pulmonary embolism (PE) automatically [29]. This required an initial artery-vein separation from the CT scans of the vasculature which was performed directly from the input imaging data by the image skeletonization technique described in Section 1.2.1. The skeletons are traced from the end of the branches and traversed upward toward

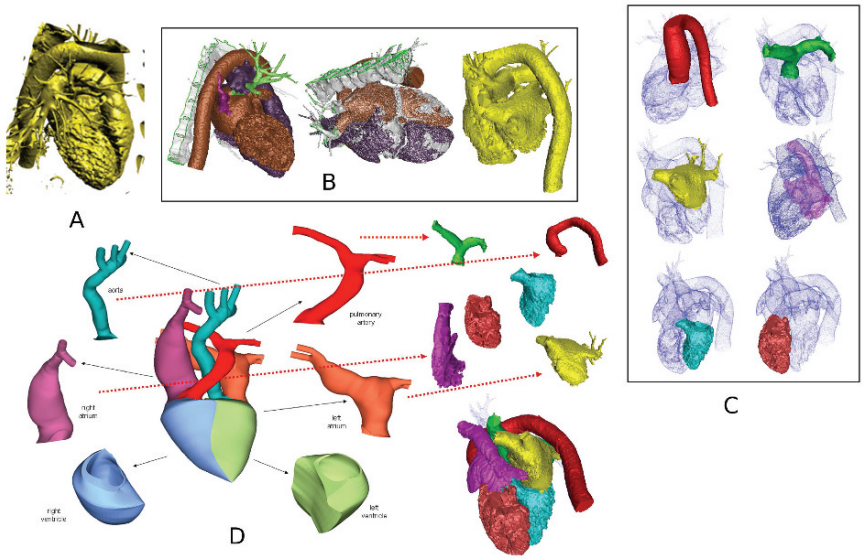


Fig. 1.15 Result: heart II. Starting with an isocontour extracted from the raw imaging data (A), the model is first curated to extract a clean geometry without the nearby bone structure or the thin blood vessels which are otherwise irrelevant for the specific task of modeling heart (B). This initial model is then geometrically segmented into six main components (C) and further a correspondence is drawn with the segmented template. The processing of the template heart model, namely, skeletonization and annotated segmentation is shown in (D)

the heart. Once the trace reached the patient heart through a series of disambiguation, as needed because of the poor image quality, one of the branches was tagged arterial while the other is tagged venous. At the same time, the rest of the volume was classified using the voxel classification technique as described in Section 1.2.1 (Fig. 1.16b, c). This led to a complete characterization of the CT data into the major components along with the artery and vein separated (Fig. 1.16d–f).

1.3.2.2 Abdominal Aorta

Starting with the CT scan of the abdominal section of the patient, we first extracted the geometry using an isovalue that best captures the geometry of the abdominal aorta along with the surrounding bone structure and other anatomical parts which are however irrelevant for the modeling of the aorta itself. To separate the aorta, we performed geometry segmentation, using the stable manifold of the maxima of the distance function. The segmented aorta is shown in green in Fig. 1.17a, b. We then performed geometry skeletonization on this segmented geometry and as a result a one-dimensional skeletal structure was produced (Fig. 1.17c, d). This skeleton was then used for building a swept hexahedral volume that best represented the geometry of the aorta, as well as served the purpose of a control mesh which was then used to build a solid NURBS model of the abdominal aorta (Fig. 1.17e, f).

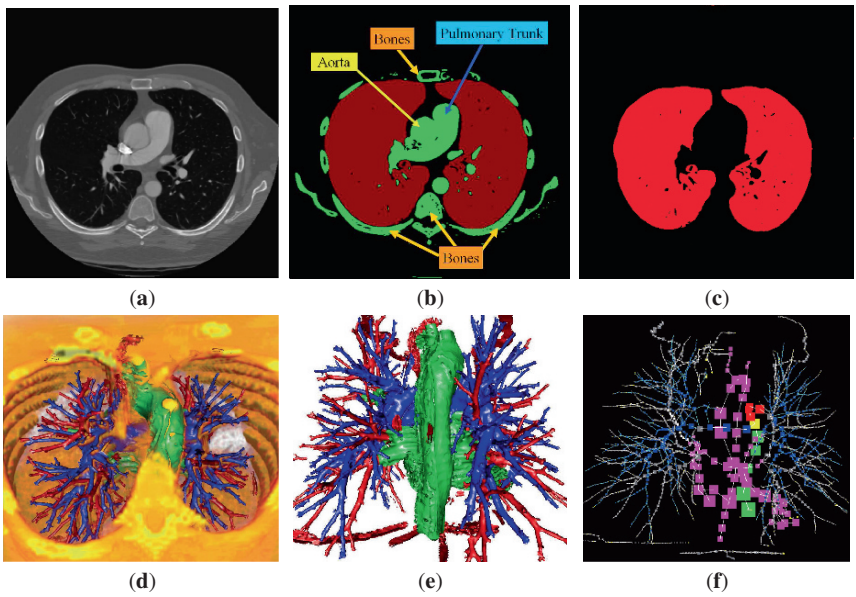


Fig. 1.16 Result: pulmonary artery. The imaging data (one slice is shown in (a)), is first classified to identify the voxels belonging to lungs and the vasculature (b, c). Using the skeletonization technique from Section 1.2.1, the arterial (red) and venous (blue) trees are detected. The vasculature and heart (green) are shown in (d–f) superimposed with the volume rendering of the CTA image

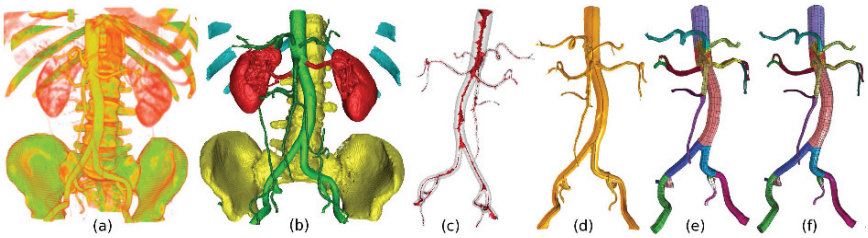


Fig. 1.17 Result: abdominal aorta. Volume rendering of the original imaging data is shown in (a). An isocontour is extracted from the imaging data from which the abdominal aorta (green) is segmented (b). The medial axis and the linear skeleton extracted from it are shown in (c, d) from which an initial control polyhedron (e) and the final NURBS model (f) are created

1.3.2.3 Thoracic Aorta

Starting with the scan of the patient heart, we first segmented out and extracted the geometry of the thoracic aorta via isocontouring. We then performed the skeletonization of this geometry in order to obtain an one-dimensional polylinear skeletal (Fig. 1.18 A.1). An external pathway is added to the skeletal structure in order to simulate the LVAD used in time of open-heart surgery. Using the skeleton as a sweeping path, we then built a hexahedral control mesh that best represented the

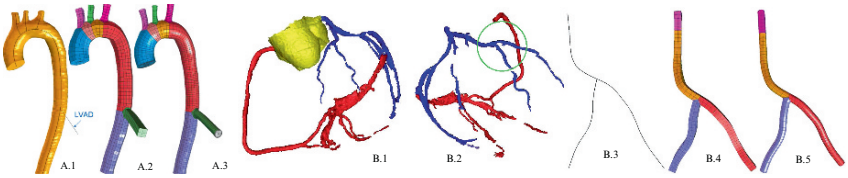


Fig. 1.18 Result: (A) Thoracic aorta. Skeleton is extracted from the geometric model (A.1). An artificial pathway is added to simulate the LVAD. The hexahedral control mesh and the resulting NURBS model are shown in (A.2, 3). (B) Coronary artery. The coronary artery is segmented into left and right subtrees (B.1, 2). NURBS mesh of a small portion of the tree is then built using the extracted skeleton (B.3–5)

aorta along with its inner volume (Fig. 1.18 A.2). This control mesh was then used to construct a solid NURBS mesh of the thoracic aorta as shown in Fig. 1.18 A.3.

1.3.2.4 Coronary Artery

Starting with a CT scan of the patient heart, first the coronary artery was extracted as shown in Fig. 1.18B.1. The coronary artery has two main branches – right and left, which were then geometrically segmented from the whole vascular structure. Figure 1.18B.2 shows both the vasculature trees colored in red and blue. We then computed the skeletal structure for each of these substructures using the skeletonization method described earlier (Fig. 1.18B.3) and subsequently a NURBS mesh model was constructed using the swept volume technique as described in the meshing subsection earlier (Fig. 1.18B.4, B.5).

1.4 Conclusions

We have developed a comprehensive software collection (http://cvcweb.ices.utexas.edu/software/) of data processing tools (http://cvcweb.ices.utexas.edu/cvc/projects/medx/pipeline.php) that can be utilized in producing patient specific, and spatially realistic, linear and curved, boundary and finite element models of human cardio-vasculature. Such a combination of geometry extraction and geometric modeling are the necessary enablers for quantitative and interrogative querying, analysis and visualization. These boundary and finite elements are additionally useful for a variety of physiological, bio-chemical modeling and simulation of normal or diseased conditions. They are also useful for virtual surgical training, treatment planning and drug or dosage delivery. Current imaging modalities we have successfully processed through our software include Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), and their blood perfused variations for cardio-vasculature modeling and micro-CT for osteoporotic bone modeling.

Acknowledgements The research was NSF grant CNS-0540033 and NIH contracts: P20-RR020647, R01-EB00487, R01-GM074258, R01-GM07308. We also thank Dr. Charlie Walvaert of the Austin Heart Hospital, USA for providing us with the CT64 thoracic scan. Thanks are also due to Joe Rivera of CVC for his immense help with data processing. We would like to additionally thank Dr. Tom Hughes, Dr. Yuri Bazilevs for helpful discussions, and Dr. Sangmin Park and Dr. Yongjie Zhang for their help with image processing and mesh generation parts of this paper.

References

1. Winslow, R.L., Scollan, D.F., Greenstein, J.L., Yung, C.K., Jr., W.B., Bhanot, G., Gresh, D.L., Rogowitz, B.E.: Mapping, modeling, and visual exploration of structure-function relationships in the heart. Deep Computing for the Life Sciences **40**(2) (2001)
2. Luo, C., Rudy, Y.: A model of the ventricular cardiac action potential: Depolarization, repolarization and their interaction. Circulation Research **68**(6) (1991) 1501–1526
3. Luo, C., Rudy, Y.: A dynamic model of the cardiac ventricular action potential: I. simulations of ionic currents and concentration changes. Circulation Research **74**(6) (1994) 1071–1096
4. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal of Physiology **117**(1952) 500–544
5. Hille, B.: Ionic Channels of Excitable Membranes, 2nd edn. Sinauer Associates, Sunderland, MA (1992)
6. Winslow, R.L., Scollan, D.F., Holmes, A., Yung, C.K., Zhang, J., Jafri, M.S.: Electrophysiological modeling of cardiac ventricular function: from cell to organ. Annual Reviews in Biomedical Engineering **2**(2000) 119–155
7. Vetter, F., McCulloch, A., Rogers, J.: A finite element model of passive mechanics and electrical propagation in the rabbit ventricles. Computers in Cardiology (1998) 705–708
8. Rogers, J.M., McCulloch, A.D.: A collocation-galerkin finite element model of cardiac action potential propagation. IEEE Transactions on Biomedical Engineering **41**(1994) 743–757
9. Rudy, Y., Plonsey, R.: A comparison of volume conductor and source geometry effects on body surface and epicardial potentials. Circulation Research **46**(1980) 283–291
10. Costa, K.D., Hunter, P.J., Wayne, J.S., Waldmann, L.K., Guccione, J.M., McCulloch, A.D.: A three-dimensional finite element method for large elastic deformations of ventricular myocardium: Ii - prolate spheroidal coordinates. Journal of Biomedical Engineering **118**(4) (1996) 464–472
11. Hunter, P., McCulloch, A., Nielsen, P., Smaill, B.: A finite element model of passive ventricular mechanics. ASMS BED **9**(1988) 387–397
12. Sachse, F.B.: Computational Cardiology: Modeling of Anatomy, Electrophysiology, and Mechanics. LNCS 2966. Springer, Berlin, Heidelberg, New York (2004)
13. Taylor, C., Hughes, T., Zarins, C.: Finite element modeling of blood flow in arteries. Computer Methods in Applied Mechanics and Engineering **158**(1–2) (1998) 155–196
14. Taylor, C., Hughes, T., Zarins, C.: Finite element modeling of 3-dimensional pulsatile flow in the abdominal aorta: relevance to atherosclerosis. Annals of Biomedical Engineering **26**(6) (1998) 1–13
15. Taylor, C., Hughes, T., Zarins, C.: Effect of exercise on hemodynamic conditions in the abdominal aorta. Journal of Vascular Surgery **29**(1999) 1077–89
16. Sahni, O.: Adaptive procedure for efficient blood-flow simulations. PhD thesis, RPI (2005)
17. Sahni, O., Mueller, J., Jansen, K.E., Shephard, M.S., Taylor, C.A.: Efficient anisotropic adaptive discretization of the cardiovascular system. Technical report, RPI (2005)
18. Yin, L., Luo, X., Shephard, M.S.: Identifying and meshing thin sections of 3-d curved domains. Technical report, RPI (2005)

19. Hackbusch, W.: Multi-Grid Methods and Applications. Springer Verlag, Berlin, Heidelberg, New York, Tokyo (1985)
20. Braess, D.: Towards algebraic multigrid for elliptic problems of second order. Computing **55**(1995) 379–393
21. Brown, P., Byrne, G., Hindmarsh, A.: VODE: a variable-coefficient ode solver. SIAM Journal on Scientific Computation **10**(1989) 1038–1057
22. de Munck, J.: A linear discretization of the volume conductor boundary integral equation using analytically integrated elements. IEEE Transactions on Biomedical Engineering **39**(9) (1992) 986–990
23. Team, P.E.: Essential Atlas of Anatomy, English edn. Parramon Ediciones, Barcelona, Spain (2001)
24. Zhang, Y., Bazilevs, Y., Goswami, S., Bajaj, C.L., Hughes, T.J.R.: Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow. Computer Methods in Applied Mechanics and Engineering (CMAME) **196**(29–30) (2007) 2943–2959
25. Gady Agam, Samuel G. Armato, I., Wu, C.: Vessel tree reconstruction in thoracic ct scans with application to nodule detection. IEEE Transaction on Medical Imaging **24**(4) (2005) 486–499
26. Dehmeshki, J., Ye, X., Wang, F., Lin, X.Y., Abaei, M., Siddique, M., Qanadli, S.: An accurate and reproducible scheme for quantification of coronary artery calcification in ct scans. In: Proceedings of the 26th Annual International Conference of IEEE EMBS, IEEE, The International Society for Optical Engineering (2004) 1918–1921
27. Yoshitaka Masutani, H.M., Doi, J.: Computerized detection of pulmonary embolism in spiral ct angiography based on volumetric image analysis. IEEE Transaction on Medical Imaging **21**(12) (2002) 1517–1523
28. Park, S.M., Gladish, G.W., Bajaj, C.L.: Artery-vein separation from thoracic CTA scans. IEEE Transactions on Medical Imaging (Submitted)
29. Park, S.M., Gladish, G.W., Bajaj, C.L.: Automatic pulmonary embolism detection from thoracic CTA scans. IEEE Transactions on Medical Imaging (Submitted)
30. Schoepf, U.J., Costello, P.: Ct angiography for diagnosis of pulmonary embolism: state of the art. Radiology **230**(2) (2004) 329–337
31. Gonzalez, R., Woods, R.: Digital image processing. Addison-Wesley, New York (1992)
32. Pratt, W.: Digital Image Processing, 2nd edn. A Wiley-Interscience, New York (1991)
33. Caselles, V., Lisani, J., Morel, J., Sapiro, G.: Shape preserving local histogram modification. IEEE Transactions on Image Processing **8**(2) (1998) 220–230
34. Stark, J.: Adaptive contrast enhancement using generalization of histogram equalization. IEEE Transactions on Image Processing **9**(5) (2000) 889–906
35. Jobson, D., Rahman, Z., Woodell, G.: Properties and performance of a center/surround retinex. IEEE Transactions on Image Processing **6**(3) (1997) 451–462
36. Jobson, D., Rahman, Z., Woodell, G.: A multiscale retinex for bridging the gap between color images and the human observation of scenes. IEEE Transactions on Image Processing **6**(7) (1997) 965–976
37. Lu, J., Healy, D., Weaver, J.: Contrast enhancement of medical images using multiscale edge representation. Optical Engineering **33**(7) (1994) 2151–2161
38. Laine, A., Schuler, S., Fan, J., Huda, W.: Mammographic feature enhancement by multiscale analysis. IEEE Transactions on Medical Imaging **13**(4) (1994) 725–738
39. Yu, Z., Bajaj, C.: A fast and adaptive algorithm for image contrast enhancement. In: Proceedings of IEEE International Conference on Image Processing. (2004) 1001–1004
40. Deriche, R.: Fast algorithm for low-level vision. IEEE Transactions on Pattern Recognition and Machine Intelligence **12**(1) (1990) 78–87
41. Young, I., Vliet, L.: Recursive implementation of the gaussian filter. Signal Processing **44**(1995) 139–151
42. Barash, D.: A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(6) (2002) 844–847

43. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. In: ACM Conference on Computer Graphics (SIGGRAPH) (2002) 257–266
44. Elad, M.: On the bilateral filter and ways to improve it. IEEE Transactions On Image Processing **11**(10) (2002) 1141–1151
45. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: 1998 IEEE International Conference on Computer Vision (1998) 836–846
46. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(7) (1990) 629–639
47. Weickert, J.: Anisotropic Diffusion In Image Processing. ECMI Series, Teubner, Stuttgart, ISBN 3-519-02606-6 (1998)
48. Donoho, D., Johnson, I.: Ideal spatial adaptation via wavelet shrinkage. Biometrika **81**(1994) 425–455
49. Xu, Y., Weaver, J.B., Healy, D.M., Lu, J.: Wavelet transform domain filters: A spatially selective noise filtration technique. IEEE Transactions on Image Processing **3**(6) (1994) 747–758
50. Hamza, A.B., Luque, P., Martinez, J., Roman, R.: Removing noise and preserving details with relaxed median filters. Journal of Mathematical Imaging and Vision **11**(2) (1999) 161–177
51. Hamza, A.B., Krim, H.: Image denoising: A nonlinear robust statistical approach. IEEE Transactions on Signal Processing **49**(12) (2001) 3045–3054
52. Stoschek, A., Hegerl, R.: Denoising of electron tomographic reconstructions using multiscale transformations. Journal of Structural Biology **120**(1997) 257–265
53. Frangakis, A., Hegerl, R.: Noise reduction in electron tomographic reconstructions using nonlinear anisotropic diffusion. Journal of Structural Biology **135**(2001) 239–250
54. Bajaj, C., Wu, Q., Xu, G.: Level set based volumetric anisotropic diffusion. In: ICES Technical Report 301, The University of Texas at Austin (2003)
55. Jiang, W., Baker, M., Wu, Q., Bajaj, C., Chiu, W.: Applications of bilateral denoising filter in biological electron microscopy. Journal of Structural Biology **144**(2003) 114–122
56. Yu, Z., Bajaj, C.: A segmentation-free approach for skeletonization of gray-scale images via anisotropic vector diffusion. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'04). Volume 1. (2004) 415–420
57. Bezdek, J.: A convergence theorem for the fuzzy ISODATA clustering algorithm. IEEE Transactions on Pattern Analysis Machine Intelligence **2**(1) (1980) 1–8
58. Titterington, D.M., Smith, A.F.M., Makov, U.E.: Statistical Analysis of Finite Mixture Distrubutions. J. Wiley, Chichester (1985)
59. Pham, D.L., Prince, J.L.: Adaptive fuzzy segmentation of magnetic resonance images. IEEE Transactions on Medical Imaging **18**(9) (1998) 737–752
60. Pham, D.L., Prince, J.L.: An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities. Pattern Recognition Letters **20**(1) (1999) 57–68
61. Ahmed, M.N., Yamany, S.M., Farag, A.A., Moriarty, T.: A bias field estimation and adaptive segmentation of MRI data using a modified Fuzzy C-Means algorithm. In: Proceedings of 13th International Conference on Computer Assisted Radiology and Surgery (1999)
62. Ahmed, M.N., Yamany, S.M., Mohamed, N., Farag, A.A.: A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. IEEE Transactions on Medical Imaging **21**(3) (2002) 193–199
63. Gopal, S.S., Hebert, T.J.: Maximum likelihood pixel labeling using a spatially variant finite. IEEE Transaction on Nuclear Science **44**(4) (1999) 1578–1582
64. Laidlaw, D.H., Fleischer, K.W., Barr, A.H.: Bayesian mixture classification of mri data for geometric modeling and visualization. A poster presented at the First International Workshop on Statistical Mixture Modeling, Aussois, France (1995)
65. Ahmed, M.N., Yamany, S.M., Mohamed, N., Farag, A.A., Moriarty, T.: A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data. IEEE Transactions on Medical Imaging **21**(3) (2002)

66. Kindlmann, G., Darkin, J.W.: Semi-automatic generation of transfer functions for direct volume rendering. In: Proceedings of 1998 Symposium on Volume Visualization (1998) 79–86
67. Laidlaw, D.: Geometric Model Extraction from Magnetic Resonance Volume Data (PhD thesis). PhD thesis, CalTech University, Arizona (1995)
68. Tomasi, C., R.Madcuchi: Bilateral filtering for gray and color images. In: Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India (1998) 839–846
69. Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing **7**(3) (1998) 359–369
70. Yu, Z., Bajaj, C.: Normalized gradient vector diffusion and image segmentation. In: Proceedings of European Conference on Computer Vision (2002) 517–530
71. Bajaj, C., Pascucci, V., Schikore, D.: The contour spectrum. In: Proceedings of IEEE Visualization Conference (1997) 167–173
72. Park, S., Bajaj, C.: Feature selection of 3d volume data through multi-dimensional transfer functions. Pattern Recognition Letters **28**(3) (2007) 367–374
73. Ellis, R.: Macromolecular crowding: obvious but underappreciated. Trends in Biochemical Sciences **26**(10) (2001) 597–604
74. Hessler, D., Young, S.J., Ellisman, M.H.: A flexible environment for the visualization of three-dimensional biological structures. Journal of Structural Biology **116**(1) (1996) 113–119
75. Kremer, J., Mastronarde, D., McIntosh, J.: Computer visualization of three-dimensional image data using imod. Journal of Structural Biology **116**(1996) 71–76
76. Li, Y., Leith, A., Frank, J.: Tinkerbell-a tool for interactive segmentation of 3d data. Journal of Structural Bioloby **120**(3) (1997) 266–275
77. McEwen, B., Marko, M.: Three-dimensional electron micros-copy and its application to mitosis research. Methods in Cell Biology **61**(1999) 81–111
78. Harlow, M., Ress, D., Stoschek, A., Marshall, R., McMahan, U.: The architecture of active zone material at the frog's neuromuscular junction. Nature **409**(2001) 479–484
79. Volkmann, N.: A novel three-dimensional variant of the watershed transform for segmentation of electron density maps. Journal of Structural Biology **138**(1–2) (2002) 123–129
80. Frangakis, A.S., Hegerl, R.: Segmentation of two- and three-dimensional data from electron microscopy using eigenvector analysis. Journal of Structural Biology **138**(1–2) (2002) 105–113
81. Zhou, Z.H., Baker, M.L., Jiang, W., Dougherty, M., Jakana, J., Dong, G., Lu, G., Chiu, W.: Electron cryomicroscopy and bioinformatics suggest protein fold models for rice dwarf virus. Nature Structural Biology **8**(10) (2001) 868–873
82. Jiang, W., Li, Z., Baker, M.L., Prevelige, P.E., Chiu, W.: Coat protein fold and maturation transition of bacteriophage p22 seen at subnanometer resolution. Nature Structural Biology **10**(2) (2003) 131–135
83. Marko, M., Leith, A.: Sterecon - three-dimensional reconstructions from stereoscopic contouring. Journal of Structural Biology **116**(1) (1996) 93–98
84. Bajaj, C., Yu, Z., Auer, M.: Volumetric feature extraction and visualization of tomographic molecular imaging. Journal of Structural Biology **145**(1) (2004) 168–180
85. Yu, Z., Bajaj, C.: Automatic ultrastructure segmentation of reconstructed cryoem maps of icosahedral viruses. IEEE Transactions on Image Processing: Special Issue on Molecular and Cellular Bioimaging **14**(9) (2005) 1324–1337
86. Baker, M., Yu, Z., Chiu, W., Bajaj, C.: Automated Segmentation of Molecular Subunits in Electron Cryomicroscopy Density Maps. Journal of Structural Biology (2006) online–version
87. Malladi, R., Sethian, J.: A real-time algorithm for medical shape recovery. In: IEEE International Conference on Computer Vision (1998) 304–310
88. Sethian, J.: A marching level set method for monotonically advancing fronts. Proceedings of the National Academy Science **93**(4) (1996) 1591–1595
89. Sethian, J.: Level Set Methods and Fast Marching Methods, 2nd edn. Cambridge University Press, Cambridge (1999)

90. Sifakis, E., Tziritas, G.: Moving object localization using a multi-label fast marching algorithm. Signal Processing: Image Communication **16**(10) (2001) 963–976
91. Tari, S., Shah, J., Pien, H.: Extraction of shape skeletons from gray-scale images. Computer Vision and Image Understanding **66**(2) (1997) 133–146
92. Chung, D.H., Sapiro, G.: Segmentation-free skeletonization of gray-scale images via pde's. In: International Conference on Image Processing. (2000) 927–930
93. Lindeberg, T.: Scale-space Theory in Computer Vision. The Kluwer International Series in Engineering and Computer Science, Kluwer, Netherlands (1994)
94. Pizer, S.M., Eberly, D., Fritsch, D.S., Morse, B.S.: Zoom invariant vision of figural shape: The mathematics of cores. Computer Vision and Image Understanding **69**(1) (1998) 55–71
95. Morse, B.S., Pizer, S.M., Puff, D.T., Gu, C.: Zoominvariant vision of figural shape: Effects on cores of images disturbances. Computer Vision and Image Understanding **69**(1) (1998) 72–86
96. Jang, J.H., Hong, K.S.: A pseudo-distance map for the segmentation-free skeletonization of gray-scale images. In: Proceedings of the International Conference on Computer Vision. (2001) 18–23
97. Castro, E.D., Morandi, C.: Registartion of translated and rotated images using finite fourier transforms. IEEE Transactions on Patten Analysis and Machine Intelligence **9**(5) (1986) 700–703
98. Kuglin, C.D., Hines, D.C.: The phase correlation image alignment method. In: Proceedings of IEEE International Conference on Cybrnetics and Society. (1975) 163–165
99. Lehmann, T.M.: A two stage algorithm for model-based registration of medical images. In: Proceedings of the International Conference on Pattern Recognition ICPR'98. (1998) 344–352
100. Araiza, R., Averill, M., Keller, G., Starks, S., Bajaj, C.: 3D image registration using Fast Fourier Transform, with potential applications to geoinformatics and bioinformatics. In: Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU06. (2006) 817–824
101. Dutt, A., Rokhlin, V.: Fast fourier transform for nonequispaced data. SIAM Journal of Scientific Computing **14**(1993) 1368–1393
102. Dutt, A., Rokhlin, V.: Fast fourier transform for nonequispaced data ii. Applied and Computational Harmonic Analysis **2**(1995) 85–100
103. Brown, L.G.: A survey of image registration techniques. ACM Computing Surveys **24**(4) (1992) 325–376
104. Bajcsy, R., Kovacic, S.: Multiresolution elastic matching. Computer Vision Graphics and Image Processing **46**(1–21) (1989) 1–21
105. Christensen, G.E., Rabbitt, R.D., Miller, M.I.: Deformable templates using large deformation kinematics. IEEE Transaction on Image Processing **5**(10) (1996) 1435–1447
106. Yanovsky, I., Thompson, P., Osher, S., Leow, A.: Large deformation unbiased diffeomorphic nonlinear image registration: Theory and implementation. Technical report, UCLA CAM (2006)
107. Clarenz, U., Droske, M., Rumpf, M.: Towards fast non-rigid registration. Proceedings of the AMS **313**(2002) 67–84
108. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Communications on and Pure Applied Mathematics **42**(4) (1989)
109. Lorensen, W., Cline, H.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In: SIGGRAPH. (1987) 163–169
110. Lopes, A., Brodlie, K.: Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. In: IEEE Transactions on Visualization and Computer Graphics. Volume 9. (2003) 16–29
111. Dey, T.K.: Curve and Surface Reconstruction: Algorithms with Mathematical Analysis. Cambridge Monographs on Applied and Computational Mathematics (2006)
112. Dey, T.K., Goswami, S.: Tight cocone: A water-tight surface reconstructor. In: Proceedings of the 8th ACM Symposium on Solid Modeling and Applications. (2003) 127–134

113. Zhao, H., Osher, S., Fedkiw, R.: Fast surface reconstruction using the level set method. In: 1st IEEE Workshop on Variational and Level Set Methods. (2001) 194–202
114. Bajaj, C., Xu, G., Zhang, X.: Bio-Molecular surface constructions via a higher-order level-set method. In: Proceedings of the 14th CAD/CG International Conference, 2007, Beijing, China
115. Cheng, S.W., Wang, Y., Wu, Z.: Provable dimension detection using principal component analysis. In: Proceedings of the Symposium on Computational Geometry (2005) 208–217
116. Dey, T.K., Giesen, J., Goswami, S., Zhao, W.: Shape dimension and approximation from samples. Discrete and Computaional Geometry **29**(2003) 419–434
117. Siersma, D.: Voronoi diagrams and morse theory of the distance function (1999)
118. Dey, T.K., Giesen, J., Goswami, S.: Shape segmentation and matching with flow discretization. In: Dehne, F., Sack, J.R., Smid, M., eds.: Proceedings of Workshop Algorithms Data Structures (WADS 03). LNCS 2748, Berlin, Germany (2003) 25–36
119. Bajaj, C., Gillette, A., Goswami, S.: Topology based selection and curation of level sets. In: TopoInVis 2007, edited by A. Wiebel and H. Hege and K. Polthier and G. Scheuermann (Accepted)
120. Dey, T.K., Zhao, W.: Approximate medial axis as a Voronoi subcomplex. Computer Aided Design **36**(2) (2003) 195–202
121. Chazal, F., Lieutier, A.: The λ-medial axis. Graphical models **67**(4) (2005) 304–331
122. Cocone: Tight Cocone Software for surface reconstruction and medial axis approximation. http://www.cse.ohio-state.edu/∼tamaldey/cocone.html (2003)
123. Borgefors, G., Nystrom, I., Baja, G.D.: Computing skeletons in three dimensions. Pattern Recognition **32**(7) (1999)
124. Bitter, I., Kaufman, A., Sato, M.: Penalized distance volumetric skeleton algorithm. IEEE TVCG **7**(3) (2001)
125. Bouix, S., Siddiqi, K., Tannenbaum, A.: Flux driven fly throughs. In: IEEE Conference on Computer Vision and Pattern Recognition. (2003) 449–454
126. Hassouna, M.S., Farag, A.A.: Robust centerline extraction framework using level sets. In: IEEE Conference on Computer Vision and Pattern Recognition. (2005) 458–465
127. Zhou, Y., Toga, A.: Efficient skeletonization of volumetric objects. IEEE Transactions on Visualization and Computer Graphics **5**(3) (1999) 196–209
128. Cornea, N., Silver, D., Yuan, X., Balasubramaniam, R.: Computing hierarchical curveskeletons of 3D objects. The Visual Computer **21**(11) (2005) 945–955
129. Dey, T.K., Sun, J.: Defining and computing curve-skeletons with medial geodesic functions. In: Symposium on Geometry Processing. (2006) 143–152
130. Costa, L.: Multidimensional scale space shape analysis. In: IWSNHC3DI. (1999) 214–217
131. Ogniewicz, R.L., Kubler, O.: Hierachic voronoi skeletons. Pattern Recognition **28**(3) (1995) 343–359
132. Verroust, A., Lazarus, F.: Extracting skeletal curves from 3D scattered data. The Visual Computer **16**(2000) 15–25
133. Cornea, N., Silver, D., Min, P.: Curve skeleton applications. In: IEEE Visualization. (2005) 95–102
134. Goswami, S., Dey, T.K., Bajaj, C.L.: Identifying flat and tubular regions of a shape by unstable manifolds. In: Proceedings of the 11th Symposium Solid and Physical Modeling. (2006) 27–37
135. Eckstein, I., Joshi, A.A., Kuo, C.J., Leahy, R., Desbrun, M.: Generalized surface flows for deformable registration and cortical matching. In: MICCAI. (2007) 183–192
136. Amenta, N., Choi, S., Kolluri, R.: The power crust, unions of balls, and the medial axis transform. Computational Geometry: Theory and Applications **19**(2–3) (2001) 127–153
137. Tama, F., Miyashita, O., Brooks, C.: Flexible multi-scale fitting of atomic structures into low-resolution electron density maps with elastic network normal mode analysis. Journal of Molecular Biology **337**(2004) 985–999
138. Zhang, Y., Bajaj, C.: Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. Computer Methods in Applied Mechanics and Engineering (CMAME) **195**(9–12) (2006) 942–960

139. Ju, T., Losasso, F., Schaefer, S., Warren, J.: Dual contouring of hermite data. In: SIGGRAPH 2002, Computer Graphics Proceedings, ACM Press / ACM SIGGRAPH. (2002) 339–346
140. Farin, G.: Curves and Surfaces for CAGD: A Practical Guide. 5th edn. Morgan-Kaufmann, San Francisco, CA (2002)
141. Catmull, E., Clark, J.: Recursively generated b-spline surfaces on arbitrary topological surfaces. Computer-Aided Design **10**(6) (1978) 350–355
142. Doo, D.: A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In: Proceedings on Interactive Techniques in Computer Aided Design. (1978) 157–165
143. Doo, D., Sabin, M.: Behavior of recursive division surfaces near extraordinary points. Computer-Aided Design **10**(6) (1978) 356–360
144. Loop, C.: A g^1 triangular spline surface of arbitrary topological type. Computer Aided Geometric Design **11**(3) (1994) 303–330
145. Krishnamurthy, V., Levoy, M.: Fitting smooth surfaces to dense polygon meshes. In: Proceedings of SIGGRAPH. (1996) 313–324
146. Eck, M., Hoppe, H.: Automatic reconstruction of b-spline surfaces of arbitrary topological type. In: Proceedings of SIGGRAPH. (1996) 325–334
147. Cohen-Steiner, D., Alliez, P., Desbrun, M.: Variational shape approximation. In: Proceedings of SIGGRAPH. (2004) 905–914
148. Dong, S., Bremer, P.T., Garland, M., Pascucci, V., Hart, J.: Spectral surface quadrangulation. ACM Transactions on Graphics **25**(3) (2006) 1057–1066
149. Ying, L., Zorin, D.: A simple manifold-based construction of surfaces of arbitrary smoothness. ACM Transactions on Graphics **23**(3) (2004) 271–275
150. Bajaj, C., Chen, J., Xu, G.: Modeling with cubic A-patches. ACM Transactions on Graphics **14**(2) (1995) 103–133
151. Bajaj, C., Xu, G.: Smooth shell construction with mixed prism fat surfaces. Brunett, G., Bieri,H., Farin, G. (eds.), Geometric Modeling Computing Supplement **14**(2001) 19–35
152. Bajaj, C., Xu, G., Holt, R., Netravali, A.: Hierarchical multiresolution reconstruction of shell surfaces. Computer Aided Geometric Design **19**(2002) 89–112
153. Goswami, S., Gillette, A., Bajaj, C.: Efficient Delaunay mesh generation from sampled scalar function. In: Proceedings of the 16th International Meshing Roundtable. (2007) 495–511
154. Zhang, Y., Bajaj, C., Sohn, B.S.: 3D finite element meshing from imaging data. The special issue of Computer Methods in Applied Mechanics and Engineering (CMAME) on Unstructured Mesh Generation **194**(48–49) (2005) 5083–5106
155. Zhang, Y., Bajaj, C., Xu, G.: Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. In: Proceedings of 14th International Meshing Roundtable. (2005) 449–468
156. Rogers, D.F.: An Introduction to NURBS With Historical Perspective. Academic, San Diego, CA (2001)
157. Piegl, L., Tiller, W.: The NURBS Book (Monographs in Visual Communication), 2nd edn. Springer, New York (1997)
158. Thompson, J.F., Soni, B.K., Weatherill, N.P.: Grid Generation. CRC Press LLC, Boca Raton, FL (1999)
159. Gursoy, H.N.: Tetrahedral finite element mesh generation from nurbs solid models. Engineering with Computers **12**(19) (1996) 211–223
160. Anderson, C.W., Crawford-Hines, S.: Fast generation of nurbs surfaces from polygonal mesh models of human anatomy. In: Technical Report CS-99-101, Colorado State University. (2000)
161. Yu, T.Y., Soni, B.K.: Nurbs evaluation and utilization for grid generation. In: 5th International Conference on Numerical Grid Generation in Computational Field Simulations. (1996) 323–332
162. Hughes, T.J., Cottrell, J.A., Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. CMAME **194**(2005) 4135–4195

Chapter 2
Geodesic Methods for Shape and Surface Processing

Gabriel Peyré and Laurent D. Cohen

Abstract This paper reviews both the theory and practice of the numerical computation of geodesic distances on Riemannian manifolds. The notion of Riemannian manifold allows to define a local metric (a symmetric positive tensor field) that encodes the information about the problem one wishes to solve. This takes into account a local isotropic cost (whether some point should be avoided or not) and a local anisotropy (which direction should be preferred). Using this local tensor field, the geodesic distance is used to solve many problems of practical interest such as segmentation using geodesic balls and Voronoi regions, sampling points at regular geodesic distance or meshing a domain with geodesic Delaunay triangles. The shortest path for this Riemannian distance, the so-called geodesics, are also important because they follow salient curvilinear structures in the domain. We show several applications of the numerical computation of geodesic distances and shortest paths to problems in surface and shape processing, in particular segmentation, sampling, meshing and comparison of shapes.

2.1 Manifold Geometry of Surfaces

In [1], it was shown that finding the weighted distance and geodesic paths to a point leads to fast algorithms for image segmentation. In this chapter, we give a more general framework that is illustrated by different important applications.

This section introduces some basic definitions about local metric (a tensor field) on a Riemannian manifold and the associated notion of geodesic distance and minimal paths. The important point is that the geodesic distance to a set of starting points

G. Peyré and L.D. Cohen
Ceremade, UMR CNRS 7534, Université Paris-Dauphine,
75775 Paris Cedex 16, France
e-mail: {peyre,cohen}@ceremade.dauphine.fr

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

satisfies a non-linear differential equation, the Eikonal equation, which is solved to compute numerically the geodesic distance.

2.1.1 Riemannian Manifold

Parametric surface. A parameterized surface embedded in Euclidean space $\mathcal{M} \subset \mathbb{R}^k$ is a mapping

$$u \in \mathcal{D} \subset \mathbb{R}^2 \mapsto \phi(u) \in \mathcal{M}.$$

This definition can be extended to include surfaces not topologically equivalent to a disk, by considering a set of charts $\{\mathcal{D}_i\}_i$ that overlap in a smooth manner.

A curve is defined in parameter domain as a 1D mapping $t \in [0,1] \mapsto \gamma(t) \in \mathcal{D}$. This curve can be traced over the surface and its geometric realization is $\bar{\gamma}(t) \stackrel{\text{def.}}{=} \phi(\gamma(t)) \in \mathcal{M}$. The computation of the length of γ in ambient k-dimensional space $\mathbb{R}^k$ follows the usual definition, but to do the computation over the parametric domain, one needs to use a local metric (the first fundamental form) defined as follow.

Definition 1 (First fundamental form)**.** *For a parametric surface ϕ, one defines*

$$I_\phi = \left(\langle \frac{\partial \phi}{\partial u_i}, \frac{\partial \phi}{\partial u_j} \rangle \right)_{i,j=1,2}.$$

This local metric I_ϕ defines at each point the infinitesimal length of a curve

$$L(\gamma) \stackrel{\text{def.}}{=} \int_0^1 \|\bar{\gamma}'(t)\| \mathrm{d}t = \int_0^1 \sqrt{\gamma'(t)^{\mathrm{T}} I_\phi(\gamma(t)) \gamma'(t)} \mathrm{d}t.$$

This fundamental form is an intrinsic invariant that does not depend on how the surface is isometrically embedded in space (since the lengths depend only on this tensor field I_ϕ). In contrast, higher order differential quantities such as curvature might depend on the bending of the surface and are thus usually not intrinsic (with the notable exception of invariants such as the gaussian curvature).

Riemannian manifold. A parameterized surface is embedded into some Euclidean domain $\mathbb{R}^k$, which allows to define a local metric thanks to the first fundamental form I_ϕ. It is however possible to consider directly a field of positive definite tensors on a parametric domain $\mathcal{D} = \mathbb{R}^s$ (in practice here $s = 2$ for surfaces or $s = 3$ for volumes). With a slight abuse in notations, we assimilate the resulting abstract surface $\mathcal{M}$ with $\mathcal{D}$. Once again, we consider only surfaces globally parameterized by some Euclidean domain $\mathcal{D}$ and handling generic surfaces requires to split the manifold into overlapping charts.

Definition 2 (Riemannian manifold)**.** *A Riemannian manifold is an abstract parametric space $\mathcal{M} \subset \mathbb{R}^s$ equipped with a metric $x \in \mathcal{M} \mapsto H(x) \in \mathbb{R}^{s \times s}$ positive definite.*

Using the Riemannian metric, one can compute the length of a piecewise smooth curve $\gamma : [0,1] \to \mathcal{M}$

$$L(\gamma) \stackrel{\text{def.}}{=} \int_0^1 \sqrt{\gamma'(t)^{\mathrm{T}} H(\gamma(t))\gamma'(t)}\mathrm{d}t.$$

At each location x, the Riemannian tensor can be diagonalized as follow

$$H(x) = \lambda_1(x)e_1(x)e_1(x)^{\mathrm{T}} + \lambda_2(x)e_2(x)e_2(x)^{\mathrm{T}} \quad \text{with} \quad 0 \leq \lambda_1 \leq \lambda_2, \tag{2.1}$$

and e_1, e_2 are two orthogonal eigenvector fields. In fact, e_i should be understood as direction (un-oriented) field since both e_i and $-e_i$ are eigenvectors of the tensor. A curve γ passing at location $\gamma(t) = x$ with speed $\gamma'(t)$ has a shorter local length if $\gamma'(t)$ is colinear to $e_1(x)$ rather than any another direction. Hence shortest paths (to be defined in the next section) tend to be tangent to the direction field e_1.

In practice, the Riemannian metric H is given by the problem one wishes to solve. In image processing, the manifold is the image domain $\mathcal{M} = [0,1]^2$ equipped with a metric derived from the image (for instance its gradient). Figure 2.1 shows some some frequently used geodesic metric spaces:

- *Euclidean space:* $\mathcal{M} = \mathbb{R}^s$ and $H(x) = \mathrm{Id}_s$.
- *2D shape:* $\mathcal{M} \subset \mathbb{R}^2$ and $H(x) = \mathrm{Id}_2$.
- *Isotropic metric:* $H(x) = W(x)\mathrm{Id}_s$, $W(x) > 0$ being some weight function.
- *Parametric surface:* $H(x) = I_\phi(x)$ is the first fundamental form.
- *Image processing:* given an image $I : [0,1]^2 \to \mathbb{R}$, one can use an edge-stopping weight $W(x) = (\varepsilon + \|\nabla_x I\|)^{-1}$. This way, geodesic curves can be used to perform segmentation since they will not cross boundaries of the objects.

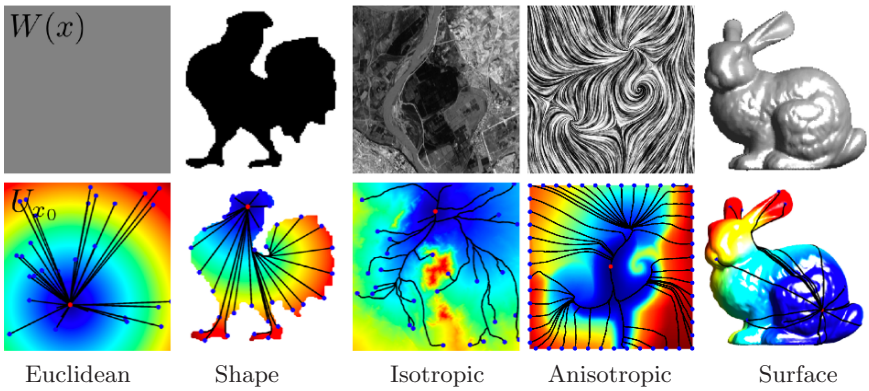


Fig. 2.1 Examples of Riemannian metrics (top row) and geodesic distances and curves (bottom row). The blue/red colormap indicates the geodesic distance to the starting point. From left to right: euclidean ($H(x) = \mathrm{Id}_2$ restricted to $\mathcal{M} = [0,1]^2$), planar domain ($H(x) = \mathrm{Id}_2$ restricted to $\mathcal{M} \neq [0,1]^2$), isotropic ($H(x) = W(x)^2\mathrm{Id}_2$ with W computed from the image using (3.6)), Riemannian manifold metric ($H(x)$ is the structure tensor of the image, see Eq. (2.8)) and 3D surface ($H(x)$ corresponds to the first fundamental form)

– *DTI imaging:* $\mathcal{M} = [0,1]^3$, and $H(x)$ is a field of diffusion tensors acquired during a scanning experiment. For DTI imaging, the direction field e_1 indicates the direction of elongated fibers of the white matter (see [2]).

The anisotropy of a metric $H(x)$ is defined as

$$\alpha(x) = \frac{\lambda_2 - \lambda_1}{\lambda_1 + \lambda_2} = 2\frac{\sqrt{ab-c^2}}{a+b} \in [0,1], \qquad \text{for} \qquad H(x) = \begin{pmatrix} a & c \\ c & b \end{pmatrix}. \tag{2.2}$$

A metric with $\alpha(x)$ close to 1 is highly directional near x, whereas a metric with $\alpha(x) = 0$ is locally isotropic near x.

2.1.2 Geodesic Distances

The local Riemannian metric $H(x)$ allows to define a global metric on the space $\mathcal{M}$ using shortest paths. This corresponds to the notion of geodesic curves.

Definition 3 (Geodesic distance)**.** *Given some Riemannian space $(\mathcal{M}, H)$ with $\mathcal{M} \subset \mathbb{R}^s$, the geodesic distance is defined as*

$$\forall (x,y) \in \mathcal{M}^2, \quad d_{\mathcal{M}}(x,y) \stackrel{\text{def.}}{=} \min_{\gamma \in \mathcal{P}(x,y)} L(\gamma)$$

where $\mathcal{P}(x,y)$ denotes the set of piecewise smooth curves joining x and y

$$\mathcal{P}(x,y) \stackrel{\text{def.}}{=} \{\gamma \setminus \gamma(0) = x \quad \textit{and} \quad \gamma(1) = y\}.$$

The shortest path between two points according to the Riemannian metric is called a geodesic. If the metric H is well chosen, then geodesic curves can be used to follow salient features on images and surfaces.

Definition 4 (Geodesic curve)**.** *A geodesic curve $\gamma \in \mathcal{P}(x,y)$ is such that $L(\gamma) = d_{\mathcal{M}}(x,y)$.*

A geodesic curve between two points might not be unique, think for instance about two anti-podal points on a sphere. In order to perform the numerical computation of geodesic distances, we fix a set of starting points $\mathcal{S} = (x_k)_k \subset \mathcal{M}$ and consider only distance and geodesic curves from this set of points.

Definition 5 (Distance map)**.** *The distance map to a set of starting points $\mathcal{S} = (x_k)_k \subset \mathcal{M}$ is defined as*

$$\forall x \in \mathcal{M}, \quad U_{\mathcal{S}}(x) \stackrel{\text{def.}}{=} \min_k d(x, x_k).$$

The main theorem that characterizes the geodesic distance is the following, that replaces the optimization problem of finding the minimum distance by a non-linear partial differential equation.

Theorem 1 (Eikonal equation)**.** *If the metric H is continuous, then for any $\mathcal{S} \subset \mathcal{M}$, the map $U_{\mathcal{S}}$ is the unique viscosity solution of the Hamilton-Jacobi equation*

$$\|\nabla_x U_{\mathcal{S}}\|_{H(x)^{-1}} = 1 \qquad \textit{with} \qquad \forall k, \quad U_{\mathcal{S}}(x_k) = 0, \tag{2.3}$$

where $\|v\|_A = \sqrt{v^{\mathrm{T}} A v}$.

It is important to notice that, even if the metric $x \mapsto H(x)$ is a smooth function, the distance function $U_{\mathcal{S}}$ might not be smooth (it exhibit gradient discontinuities). This is why the machinery of viscosity solution is needed to give a sense to the solution of the Hamilton-Jacobi equation. See for instance [3] for an introduction to viscosity solutions.

Once the distance map $U_{\mathcal{S}}$ has been computed by solving the Eikonal Eq. (2.3), one can extract a geodesic joining any point x to its closest point $x_k \in \mathcal{S}$ using a gradient descent on the function $U_{\mathcal{S}}$.

Theorem 2 (Gradient descent)**.** *The geodesic curve γ between x and its closest point in $\mathcal{S}$ solves*

$$\gamma'(t) = -\frac{H(\gamma(t))^{-1} \nabla_{\gamma(t)} U_{\mathcal{S}}}{\|H(\gamma(t))^{-1} \nabla_{\gamma(t)} U_{\mathcal{S}}\|} \qquad \textit{with} \qquad \gamma(0) = x.$$

The geodesic curve γ extracted using this gradient descent is parameterized with unit speed since $\|\gamma'\| = 1$, so that $\gamma : [0, T] \to \mathcal{M}$ where $T = d_{\mathcal{M}}(x, x_k)$.

Figure 2.2 shows examples of geodesic curves computed from a single starting point $\mathcal{S} = \{x_1\}$ in the center of the image $\mathcal{M} = [0,1]^2$ and a set of points on the boundary of $\mathcal{M}$. The geodesics are computed for a metric $H(x)$ whose anisotropy $\alpha(x)$ (defined in Eq. (2.2)) is decreasing, thus making the Riemannian space progressively closer to the Euclidean space.

For the particular case of an isotropic metric $H(x) = W(x)^2 \mathrm{Id}_2$, the geodesic distance and the shortest path satisfies

$$\|\nabla_x U_{\mathcal{S}}\| = W(x) \qquad \text{and} \qquad \gamma'(t) = -\frac{\nabla_x U_{\mathcal{S}}}{\|\nabla_x U_{\mathcal{S}}\|}. \tag{2.4}$$

This corresponds to the Eikonal equation, that has been used to compute minimal paths weighted by W [4].

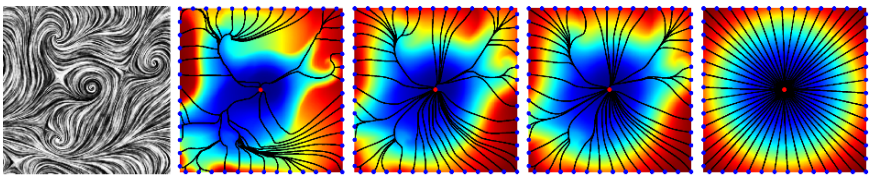


Fig. 2.2 Examples of geodesics for a tensor metric with an decreasing anisotropy α (see Eq. (2.2) for a definition of this parameter). The tensor field $H(x)$ is computed from the structure tensor of f as defined in Eq. (2.8), its eigenvalues fields $\lambda_i(x)$ are then modified to impose the anisotropy α

2.2 Numerical Computations of Geodesic Distances

In order to make all the previous definitions effective in practical situations, one needs a fast algorithm to compute the geodesic distance map $U_{\mathcal{S}}$. This section details Fast Marching algorithms based on front propagation that enable to compute the distance map by propagating the distance information from the starting points in $\mathcal{S}$.

The basic Fast Marching algorithm and several extensions are exposed in the book on Fast Marching methods [5]. For other applications to computer graphics and image processing one can see [6] and [7]. The recent book [8] treats all the details of the geometry of non-rigid surfaces, including geodesic distance computation and shape comparison. One can also see the two books [9,10] that contain review articles with some applications of Fast Marching and geodesic methods and in particular [1].

2.2.1 Front Propagation Algorithms

Depending on the properties of the metric, one needs to consider several algorithms, that all rely on the idea of front propagation. This family of algorithms allows to sort the computations in such a way that each point of the discretization grid is visited only once. This ordering is feasible for distance computation because the distance value of a grid point only depends (and can be computed) from a small number of points having only smaller distances. If one can sort the grid points with increasing distance, then one gets a coherent ordering of the computations. Of course, this is not that easy since this distance ordering would require the knowledge of the solution of the problem (the distance itself). But depending on the application, it is possible to devise a selection rule that actually select at each step the correct grid point.

A front propagation labels the points of the grid according to a state

$$S(x) \in \{Computed, Front, Far\}.$$

During the iterations of the algorithm, a point can change of label according to

$$Far \mapsto Front \mapsto Computed.$$

Computed points $S(x) = Computed$ are those that the algorithm will not consider any more (the computation of $U_{\mathcal{S}}(x)$ is done for these points). Front points $S(x) = Front$ are the points being processed (the value of $U(x) \approx U_{\mathcal{S}}(x)$ is well defined but might change in future iterations). Far points $S(x) = Far$ are points that have not been processed yet.

In practice, a front propagation algorithm requires three key ingredients:

- Given a point x in the grid, a local set of neighbors $Neigh(x)$ connected to x.
- A priority $\mathcal{P}(x)$ among points x in the front, that allows to select the point to process at a given iteration. In most application, this priority is computed as the

Table 2.1 Front propagation algorithm

1. *Initialization:* $\forall x \in \mathcal{S}, U(\mathcal{S}) \leftarrow 0$, $S(x) \leftarrow Front$, $\forall y \notin \mathcal{S}$, $S(y) \leftarrow Far$.
2. *Select point:* $x \longleftarrow \underset{S(z)=Front}{\operatorname{argmin}} \mathcal{P}(z)$.
3. *Tag:* $S(x) \leftarrow Computed$.
4. *Update neighbors:* for all $y \in Neigh(x)$,
   – If $S(y) = Far$, then $S(y) \leftarrow Front$ and $U(y) \leftarrow Update(y)$.
   – If $S(y) = Front$, then $U(y) \leftarrow \min(U(y), Update(y))$.
   – Recompute the priority $\mathcal{P}(y)$.
5. *Stop:* If $x \neq x_1$, go back to 2.

current value of the distance $\mathcal{P}(x) \overset{\text{def.}}{=} U(x)$. Section 2.3.2 shows how to change this priority in order to speed up computations.

– A procedure $x \mapsto Update(x) \in \mathbb{R}$ that computes the distance value $U(x)$ approximating $U_{\mathcal{S}}(x)$ knowing the value $U(x)$ for computed point and an approximate value for points in the front. This procedure usually solves some kind of equation that discretizes the Eikonal Eq. (2.3) one wishes to solve.

Table 2.1 gives the details of the front propagation algorithm that computes a distance map U approximating $U_{\mathcal{S}}(x)$ on a discrete grid. The following section details for actual implementations of the $Update$ procedure for different metrics (Table 2.1).

The numerical complexity of this scheme is $O(n\log(n))$ for a discrete set of n points. This is because all the points are visited (tagged *Computed*) once, and the selection of $\min \mathcal{P}$ from the front points takes at most $\log(n)$ operations with a special heap data structure (although in practice it takes much less and the algorithm is nearly linear in time).

2.2.2 Eikonal Equation Discretization

On a square grid. The classical Fast Marching algorithm, introduced by Sethian [5], is a fast procedure to solve the Eikonal Eq. (2.3) for an isotropic metric $H(x) = W(x)^2 \mathrm{Id}_s$ for a uniform regular grid that discretizes $[0,1]^s$. We recall this procedure for a planar domain $s = 2$ although it can be extended to any dimension.

In order to capture the viscosity solution of an Hamilton Jabobi equation, one cannot use standard finite differences because of the apparition of shocks and singularities in the solution of the equation. One needs to choose, at each grid point, the optimal finite difference scheme (differentiation on the left or on the right to approximate d/dx for instance). This optimal differentiation should be chosen in the direction where the solution of the equation decreases. This is called an upwind finite difference scheme, and on a 2D grid with spacing h it leads to find $u = Update(x)$ at a grid point $x = x_{i,j}$ that is the smallest solution of

$$\begin{aligned} &\max(u-U(x_{i-1,j}),u-U(x_{i+1,j}),0)^2+ \\ &\max(u-U(x_{i,j-1}),u-U(x_{i,j+1}),0)^2 = h^2W(x_{i,j})^2. \end{aligned} \tag{2.5}$$

The smallest solution of this equation leads to a stable and convergent scheme that can be used in the front propagation algorithm listing 2.1.

On a triangulation. The classical Fast Marching algorithm is restricted to isotropic metrics on a regular grid. This setting is useful for image and volumetric data processing, but in order to deal with arbitrary Riemmanian surfaces embedded in $\mathbb{R}^k$, one needs to modify Eq. (2.5).

Kimmel and Sethian [11] have developed a version of the Fast Marching algorithm for a surface $\mathcal{M} \subset \mathbb{R}^k$ with metric $W(x)$ for x in embedding space $\mathbb{R}^k$. In the continuous setting, a parametric surface $(\mathcal{M}, \phi)$ embedded with a metric $W(x)$ in ambient space corresponds to a Riemannian manifold with a metric $I_\phi(\bar{x})W(\phi(\bar{x}))$ in parameter space $\bar{x} \in \mathbb{R}^2$.

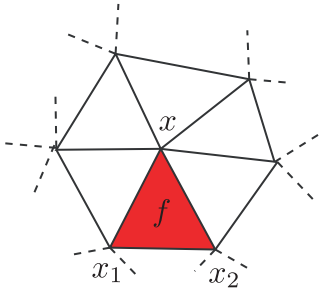
The algorithm of Kimmel and Sethian works on a triangulated mesh and treats the triangles of this mesh as locally flat and equipped with an isotropic metric $W(x)^2$. The same algorithm can be used to process an anisotropic metric $H(x) \in \mathbb{R}^{2\times 2}$ defined on a square lattice (an image), by locally connecting a pixel x to its four direct neighbors in order to create four adjacent triangles (that are flat). In order to describe the algorithm for these two settings (curved triangulated surface embedded in $\mathbb{R}^k$ and Riemannian manifold with arbitrary metric $H(x)$ for $x \in \mathbb{R}^2$), we solve the Eikonal equation

$$\|\nabla_x U\|_{H^{-1}(x)} = W(x)$$

locally on the triangle faces $f \in F_x$ adjacent to x in order to compute $Update(x) \approx U(x)$.

In order to compute the update value at a given vertex x, the algorithm computes an update value $Update_f(x)$ for each triangle $f \in F_x$ in the face 1-ring around x, $F_x = \{f_1, \ldots, f_k\}$. The resulting Fast Marching update step is defined as

$$Update(x) = \min_{f \in F_x} Update_f(x).$$



In order to derive the expression for $Update_f(x)$, one considers a planar triangle $f = (x, x_1, x_2)$ and denotes $X = (x_1 - x, x_2 - x) \in \mathbb{R}^{2\times 2}$. The known distances are $u = (U(x_1), U(x_2))^{\mathrm{T}} \in \mathbb{R}^2$ and one wishes to solve for $Update_f(x) = p = U(x)$.

The linear interpolation of $U_{\mathcal{S}}$ can be written at the point x_1 and x_2 as

$$\text{for } i \in \{1,2\}, \quad U_{\mathcal{S}}(x_i) \approx \langle g, x_i - x \rangle + p \qquad \text{where} \qquad g \approx \nabla_x U_{\mathcal{S}}.$$

With this approximation, Eq. (2.3) leads to a quadratic equation

$$\begin{cases} U = X^{\mathrm{T}} g + p\mathbb{I} \\ \|g\|^2_{H^{-1}(x)} = W(x)^2 \end{cases} \quad \Longrightarrow \quad \mathbb{I}^{\mathrm{T}} Q \mathbb{I} p^2 + 2(\mathbb{I}^{\mathrm{T}} Q u) p + (u^{\mathrm{T}} Q u - W(x)^2) = 0.$$

where $\mathbb{I} = (1,1)^{\mathrm{T}} \in \mathbb{R}^2$ and $Q = (XH(x)^{-1}X^{\mathrm{T}})^{-1} \in \mathbb{R}^{2\times 2}$. The only admissible solution to this problem is

$$Update_f(x) = p = \frac{\mathbb{I}^{\mathrm{T}} Q u + \sqrt{(\mathbb{I}^{\mathrm{T}} Q d)^2 + \mathbb{I}^{\mathrm{T}} Q \mathbb{I}(u^{\mathrm{T}} Q u - W(x)^2)}}{\mathbb{I}^{\mathrm{T}} Q \mathbb{I}}$$

There is some technical difficulties with this scheme on triangulations that contain obtuse angles or with metric $H(x)$ with a large anisotropy, because the update procedure might not be monotone anymore. More accurate monotone schemes have been developped, see for instance [2, 12, 13]. We shall ignore these difficulties here and focus on the application of the numerical computation of geodesic distances.

2.2.3 Examples of propagations

2D isotropic propagation on a square grid. Figure 2.3 shows some examples of front propagation with the Fast Marching, for an isotropic $H(x) = W(x)^2 \mathrm{Id}_2$. The colored area shows, at some given step of the algorithm, the set of computed points (its boundary being the set of front points). During the iterations, the front propagates outwards until all the grid points are visited. The numerical complexity of this scheme is $O(n\log(n))$ for a grid of n points.

Figure 2.4 shows examples of distance functions to a starting point x_0 with the corresponding geodesics $\gamma(t)$ extracted from some ending point x_1. The front propagation is stopped when $S(x_1) = Computed$ to avoid performing useless computations. The idea of using geodesics in order to extract salient curves in images as been introduced in [14].

In practice, the difficult task is to design a metric W in order have meaningful geodesics. Here are some examples of possible choices, for image processing with an input image f:

– *Pixel value based potential:* in many applications, one simply wishes to extract curves with a constant value c. In this case, one can use a potential like

$$W(x) = \frac{1}{\varepsilon + |f(x) - c|}. \tag{2.6}$$

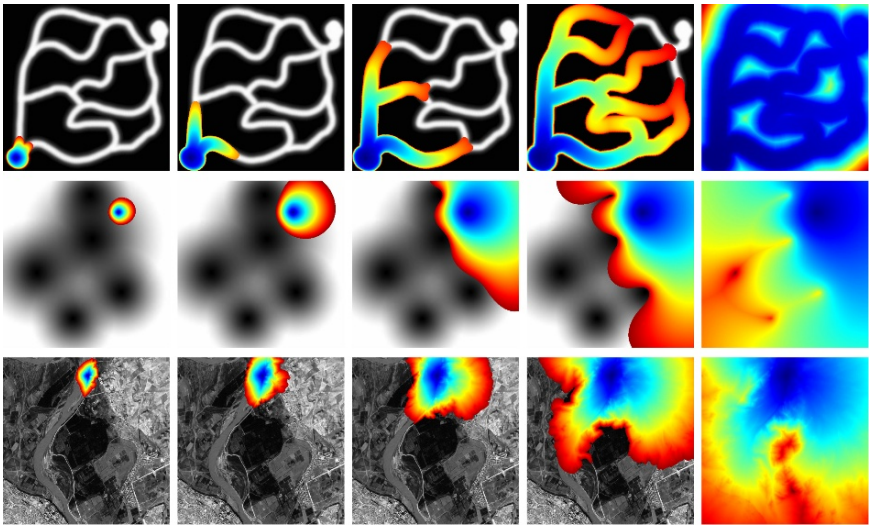


Fig. 2.3 Examples of isotropic front propagation. The colormap indicates the values of the distance functions at a given iteration of the algorithm. On rows 1 and 2, the potential W is computed using $W(x) = f(x)$ so that geodesics tend to follow bright regions. On row 3, the potential W is computed using (2.6) where c is chosen to match the intensity of the road to extract

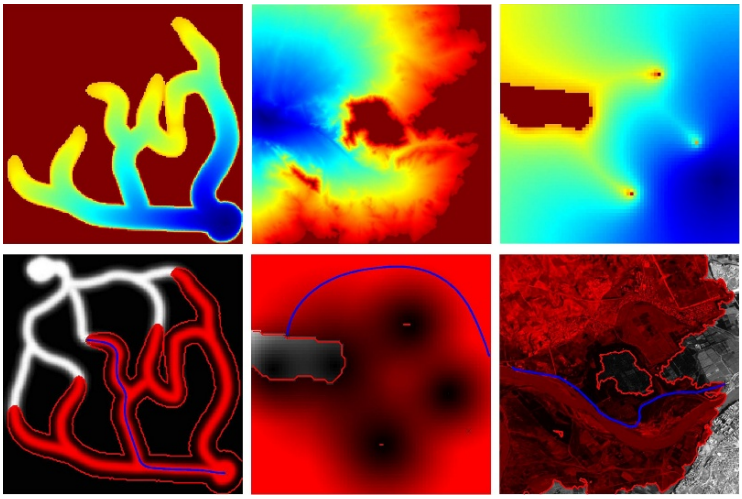


Fig. 2.4 Example of distance functions (top row) and geodesics (bottom row)

Figure 2.4, left and middle, shows examples of such curves extractions. Also in many applications related to segmentation of tubular shapes, like vessels, we are looking for curves that are located in brighter or darker regions. In this case the potential can be chosen respectively as

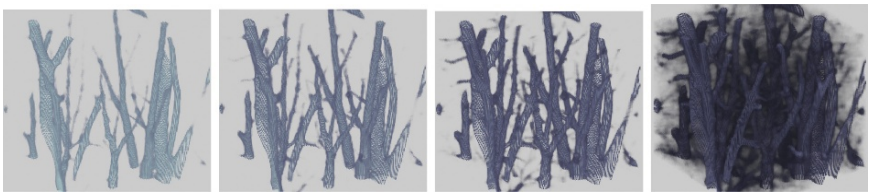


Fig. 2.5 Example of semi-transparent display of volumetric data

$$W(x) = f(x) \qquad \text{or} \qquad W(x) = -f(x) \tag{2.7}$$

and W should be rescaled to fill the range $[\varepsilon, 1]$.

– *Gradient-based potential:* for application such as edge detection one would like the geodesics to follow regions with high gradients. One can choose a potential such as

$$W(x) = \varepsilon + G_\sigma * \|\nabla_x f\|,$$

where G_σ is a smoothing kernel.

3D isotropic propagation on a square grid. The Fast Marching works the same way in any spacial dimension k and in particular can be used to extract shortest paths in 3D volumetric medical data. Such a volume is a discretization of a mapping $f : [0,1]^3 \mapsto \mathbb{R}$. Figure 2.5 shows a 3D display with a semi-transparent mapping that removes more or less parts of the data. The transparency at point (x,y,z) is defined as $\rho(f(x,y,z))$ where $\rho : [f_{\min}, f_{\max}] \to [0,1]$ is the α-mapping. Figure 2.6 shows in red the front of the Fast Marching propagation, displayed as an isosurface of $U_{\mathcal{S}}$.

Figure 2.6 shows some examples of geodesic extraction on a medical image that represents tubular structures (blood vessels) around the heart. The potential $W(x)$ is chosen as $W(x) = (|f(x) - f(x_0)| + \varepsilon)^{-1}$ where x_0 is a point given by the user and supposed to lie inside some vessel. A geodesic follows nicely a vessel since its density is constant and thus the value of f is approximately equal to $f(x_0)$ inside the vessel. Figure 2.7 shows other application of shortest path to extract tubular structures and centerlines in 3D medical data [15].

Isotropic propagation on a triangulated mesh. Figure 2.8 shows an example of propagation on a triangulated surface. The colored region corresponds to the points that are computed (its boundary being the front).

The propagation can be started from several starting points $\mathcal{S} = (x_k)_k$ in order to compute the geodesic distance map $U_{\mathcal{S}}$. Figure 2.9 shows examples of such distances to several points together with geodesics. A geodesic γ links a point x to its closest point in $\mathcal{S}$.

Anisotropic propagation on a square grid. In order to better follow the salient structures of an image f, one can replace the isotropic metric $H(x) = W(x)^2 \mathrm{Id}_s$ (examples are given here in $s = 2$ dimensions) by a fully anisotropic metric $H(x) \in \mathbb{R}^{2\times 2}$ which is a symmetric tensor field. This field might be given by the physical problem, such

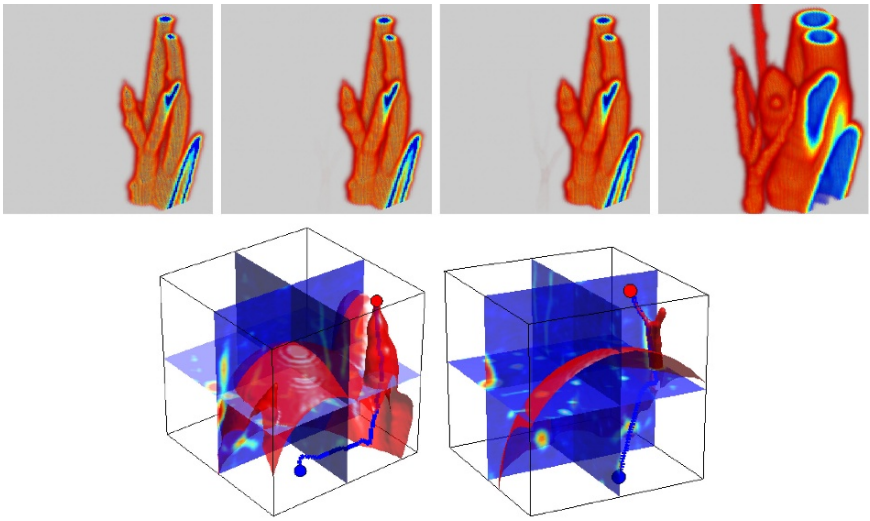


Fig. 2.6 Example of volumetric Fast Marching evolution (top row) and geodesic extractions (bottom row)

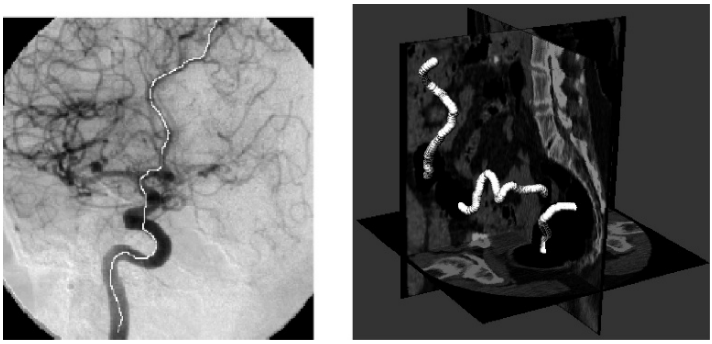


Fig. 2.7 Left: vessel extraction. Right: tubular structure extraction

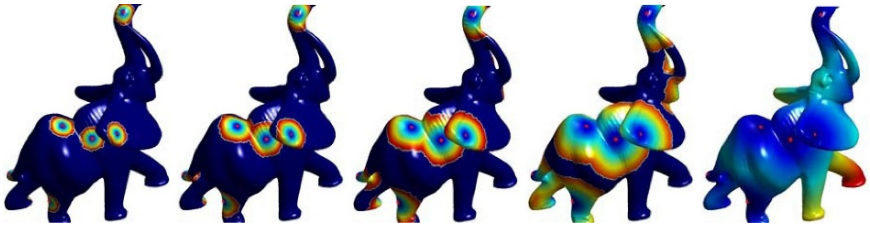


Fig. 2.8 Example of Fast Marching propagation on a triangulated mesh

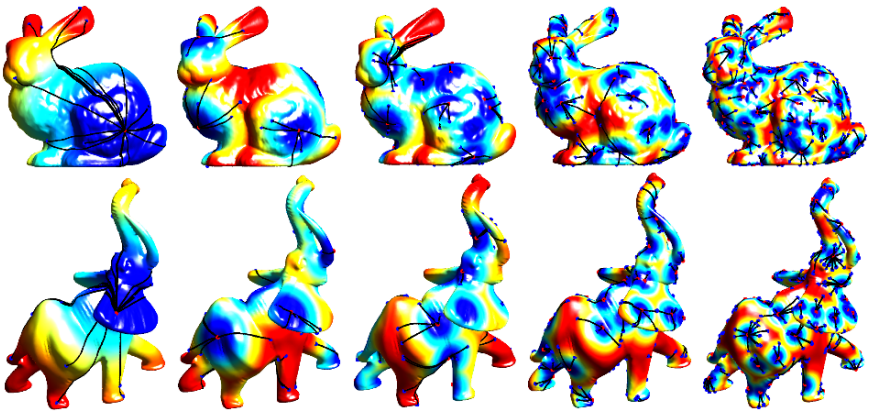


Fig. 2.9 Examples of geodesic extraction on a mesh with an increasing number of starting points

as the tensor field of DTI imaging [2]. Another option is to infer this field from some input image f.

The local orientation of a feature around a pixel x is given by the vector orthogonal to the gradient $v(x) = (\nabla_x f)^\perp$, which is computed numerically with finite differences (using maybe some little smoothing to cancel noise). This local direction information can be stored in a rank-1 tensor $T_0(x) = v(x)v(x)^{\mathrm{T}}$. In order to evaluate the local anisotropy of the image, one needs to average this tensor

$$T(x) = T_0 * G_\sigma(x) \tag{2.8}$$

where the four entries of the tensor are smoothed against a gaussian kernel G_σ of width $\sigma > 0$. The metric H corresponds to the so-called structure tensor, see for instance [16]. This local tensor T is able to extract both the local direction of edges and the local direction of textural patterns (see Fig. 2.11, left). Another option, that we do not pursue here, is to use the square of the Hessian matrix of f instead of the structure tensor.

In order to turn the structure tensor into a Riemannian metric, one can apply a non-linear mapping to the eigenvalues,

$$T(x) = \mu_1 e_1 {e_1}^{\mathrm{T}} + \mu_2 e_2 {e_2}^{\mathrm{T}} \quad \Longrightarrow \quad H(x) = \psi_1(\mu_1) e_1 {e_2}^{\mathrm{T}} + \psi_2(\mu_2) e_2 {e_2}^{\mathrm{T}}. \tag{2.9}$$

where ψ_i is a decreasing function, for instance $\psi_i(x) = (\varepsilon + |x|)^{-1}$ for a small value of ε.

Figure 2.10 shows an example of Fast Marching propagation using an anisotropic metric $H(x)$. The front propagates faster in the direction of the main eigenvector field $e_1(x)$. Figure 2.11 shows distance map for a tensor field $H(x)$ whose anisotropy α is progressively decreased, so that the geodesic distance becomes progressively Euclidean.

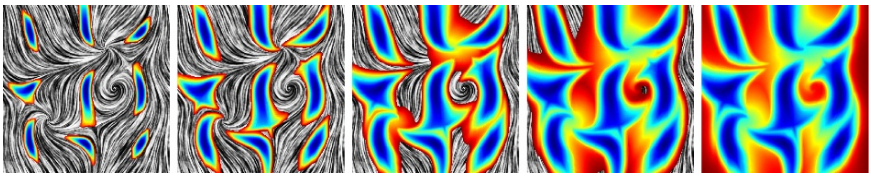


Fig. 2.10 Examples of anisotropic front propagation (from 9 starting points). The colormap indicates the values of the distance functions at a given iteration of the algorithm. The metric is computed using the structure tensor, Eq. (2.8), of the texture f shown in the background

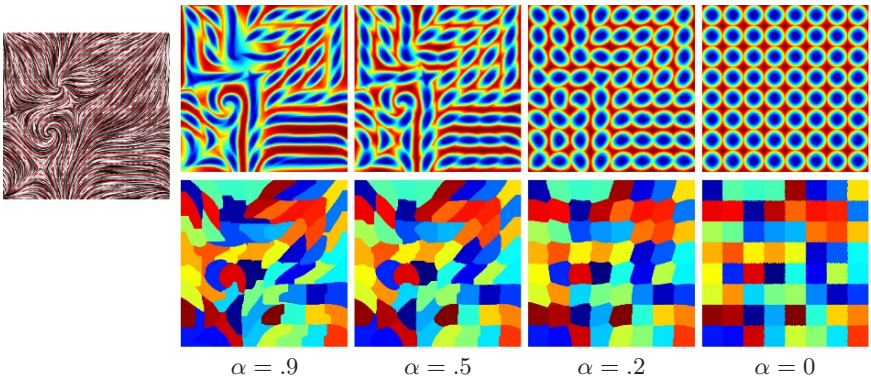


Fig. 2.11 Left image: example of texture together with the structure tensor field, computed using Eq. (2.9). Right: examples of anisotropic distances (top row) and Voronoi diagrams (bottom row) with a decreasing anisotropy α (see Eq. (2.2) for a definition of this parameter)

2.3 Applications and Extensions of Geodesic Distances

2.3.1 Shape Analysis

In order to analyze the shape of planar objects, one can consider the metric space obtained by restricting the plane to the inside of a planar domain.

Definition 6 (2D shape)**.** *A 2D shape S is a connected, closed compact set $S \subset \mathbb{R}^2$, with a piecewise-smooth boundary ∂S.*

The geodesic distance inside such a shape is obtained by constraining the curve to lie inside S.

Definition 7 (Geodesic distance in S)**.** *The geodesic distance in S for the uniform metric is*

$$d_S(x,y) \overset{\text{def.}}{=} \min_{\gamma \in \mathcal{P}(x,y)} L(\gamma) \qquad \textit{where} \qquad L(\gamma) \overset{\text{def.}}{=} \int_0^1 |\gamma'(t)| \mathrm{d}t.$$

where $\mathcal{P}(x,y) \subset S$ are the paths with starting point x and ending point y.

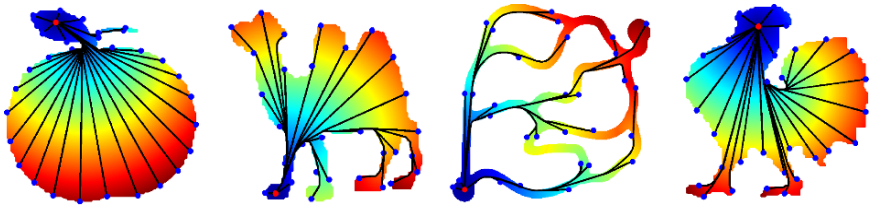


Fig. 2.12 Geodesics inside a 2D shape

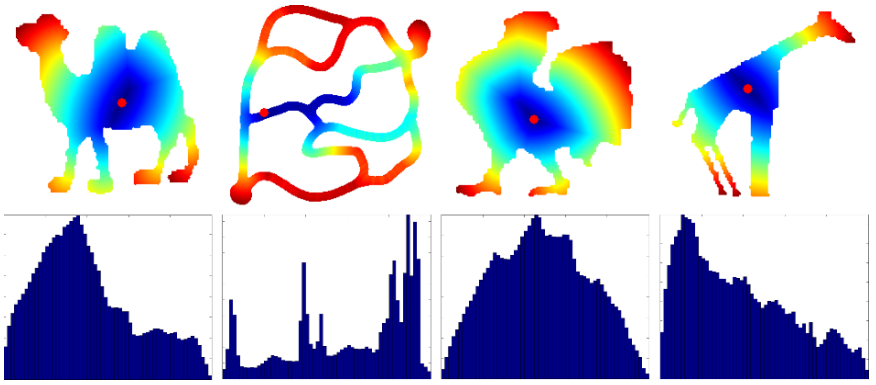


Fig. 2.13 Example of eccentricity E_S and corresponding histograms h_S

Figure 2.12 shows examples of shapes together with the geodesic distance to a starting point. The geodesic curve is the union of segments inside S and pieces of the boundary ∂S.

The geodesic distance can be used to define several functions on the 2D shape. This section studies the eccentricity of a shape, as introduced by [17] to perform shape recognition.

Definition 8 (Eccentricity)**.** *The eccentricity* $E_S : \mathcal{M} \mapsto \mathbb{R}$ *is*

$$E_S(x) \overset{\text{def.}}{=} \max_{y \in S} d_S(x,y) = \max_{y \in \partial S} d_S(x,y).$$

Figure 2.13 (top row) shows several examples of eccentricity. The colormap indicates in blue points with small eccentricity.

The points for which the minimum in the definition of E_S is obtained are called eccentric. The set of eccentric point is denoted as $\mathcal{E}(S)$.

Definition 9 (Eccentric points)**.** *An eccentric point* $x \in \mathcal{E}(S)$ *satisfies* $\exists y \in S$, $E_S(y) = d(x,y)$.

These eccentric points define regions of influence which perform a segmentation of the shape as follow

$$S = \bigcup_{x \in \mathcal{E}(S)} \{y \in S \setminus \mathrm{E}_S(y) = d(x,y)\}.$$

These eccentric points are in fact located along the boundary.

Theorem 3 (Location of eccentric points)**.** *One has* $\mathcal{E}(S) \subset \partial S$.

A more general definition of eccentricity allows to replace the maximum by a weighted average of geodesic distances.

Definition 10 (α-eccentricity)**.** *The* α *eccentricity of some shape S is defined as*

$$E_S^\alpha(x) \stackrel{\text{def.}}{=} \left(\int_S d_S(x,y)^\alpha \mathrm{d}y \right)^{1/\alpha}.$$

This eccentricity allows to generalize the notion of gravity center to the geodesic setting.

Definition 11 (Euclidean gravity center)**.** *The Euclidean gravity center is*

$$\operatorname*{argmin}_x \int_S \|x-y\|^2 \mathrm{d}y.$$

The α*-eccentric center is*

$$\operatorname*{argmin}_x E_S^\alpha(x).$$

Remark 1. *For* $\alpha = 2$*, the eccentric center is called geodesic gravity center (and equivalent to the Euclidean center in the case of an uniform metric).*

Having defined a function such as E_S inside a shape S, one can collect information about the shape using the histogram of that function.

Definition 12 (Descriptors)**.** *The eccentricity histogram descriptor* $h_S \in \mathbb{R}^m$ *of a shape is*

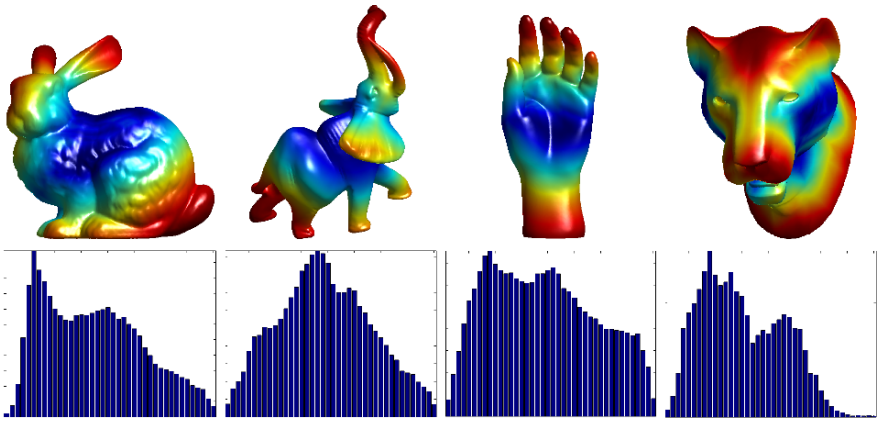


Fig. 2.14 Example of eccentricity and corresponding histograms for 3D surfaces

$$\forall i = 1,\ldots,m, \quad h_S(i) = \frac{1}{|S|}\#\left\{x \in S \setminus \frac{i-1}{m} \leq \frac{E_S(x)-\min(E_S)}{\max(E_S)-\min(E_S)} < \frac{i}{m}\right\}.$$

In particular, one can compare shapes by measuring the distance between the histograms

$$\delta(h,\tilde{h})^2 \stackrel{\text{def.}}{=} \sum_{i=1}^{m} (h(i)-\tilde{h}(i))^2.$$

These histograms are invariant if one modifies a shape isometrically. In the plane, geodesic isometry of shapes are not interesting since they are rotations and translations. One can however consider approximate isometries such as articulations, that are useful to model deformations of planar shapes, as defined in [18].

Definition 13 (ε-articulated object)**.** *An articulated object S can be split as*

$$S = \bigcup_{i=1}^{m} S_i \bigcup_{i\neq j} J_{ij},$$

(a disjoint union) with $diam(J_{ij}) \leq \varepsilon$.

Definition 14 (Articulation)**.** *An articulation is a mapping between two articulated shapes S, S' such that*

$$f : S \to S' = \bigcup_{i=1}^{m} S'_i \bigcup_{i\neq j} J'_{ij}$$

is rigid on $S_i \mapsto S'_i$.

The eccentricity is approximately invariant for shapes that are modified by articulation.

Theorem 4 (Articulation and isometry)**.** *If f is an articulation, then*

$$|d_S(x,y) - d_{S'}(x,y)| \leq m\varepsilon \quad \textit{and} \quad |E_S(x) - E_{S'}(x)| \leq m\varepsilon.$$

Starting from a shape library $\{S_1,\ldots,S_p\}$, one can use the shape signature h_S to do shape retrieval using for instance a nearest neighbor classifier, as shown in Table 2.2. Figure 2.15 shows examples of typical shape retrievals. More complex signatures can be constructed out of geodesic distances and un-supervised recognition can also be considered. We refer to [17] for a detailed study of the performance of shape recognition with eccentricity histograms. In a similar way, the eccentricity can be used to perform 3D surface retrieval, using the histograms displayed in Fig. 2.14.

Table 2.2 Shape retrieval process

1. *Dataset:* shapes $\{S_1,\ldots,S_p\}$ (binary images).
2. *Preprocessing:* compute eccentricity descriptors h_{S_i}.
3. *Input:* shape S.
4. *Retrival:* return $i^\star = \operatorname{argmin}_i \delta(h_S, h_{S_i})$.

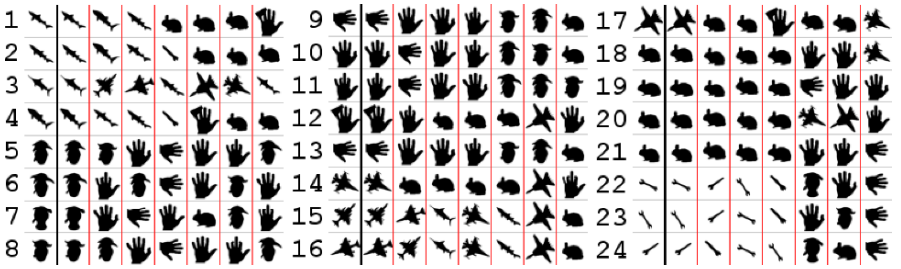


Fig. 2.15 Examples of shape recognitions. The shape on the left is the input S and the second shape in each raw is $S_{i^\star}$

2.3.2 Heuristically Driven Propagation

The various implementations of the front propagation algorithm, pseudo-code 2.1, use a simple priority $\mathcal{P}(x) = U_{x_0}(x)$, where $U(x) \approx d_{\mathcal{M}}(x_0, x)$ is the current value of the distance to the starting point. This strategy leads to an isotropic grow of the front which enforces the exploration of a large area of the computational grid. The advantage of using this priority is that it does not favor any points and thus produces provable valid approximations of geodesic distance (both on a graph with Dijkstra and on a square/triangular grid with Fast Marching).

In order to reduce the computational burden, one could think about using more aggressive ordering of the front that favors some specific direction in the front. The hope is that the front would advance faster in the direction of the goal x_1 one wishes to reach. Ultimately, one would like the front to explore only points along the geodesic $\gamma \in \mathcal{P}(x_0, x_1)$ joining the starting point to the ending point.

If one has an oracle: $V(x) \approx d(x_1, x)$ that estimates the remaining geodesic distance from the current point x to the end x_1, one can use as priority map

$$\mathcal{P}(x) = U(x) + V(x).$$

The map V is called a heuristic since the exact distance $d(x_1, x)$ is not available in practice. The value of a good heuristic close to the real distance is revealed by the following theorem.

Theorem 5 (Geodesic segment)**.** *The function $\psi(x) = d(x_0, x) + d(x_1, x)$ is minimal and constant $\psi(x) = d(x_0, x_1)$ along the geodesic path joining x_0 and x_1.*

In the setting of graph theory, the Dijkstra algorithm can be replaced by the A* (A-star), [19], which uses a heuristic to speed up computations. The following theorem proves the validity of this approach.

Theorem 6 (A* validity)**.** *If the heuristic satisfies $V(x) \leq d(x_1, x)$, then the curve $\gamma \in \mathcal{P}(x_0, x_1)$ extracted from the front propagation, algorithm 2.1, is a geodesic between x_0 and x_1.*

Over a continuous domain, one can invoke a similar (but weaker) theorem.

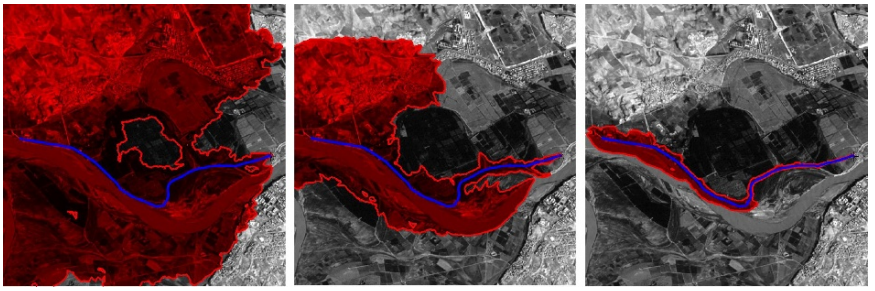


Fig. 2.16 Example of propagations with a priority $\mathcal{P}(x) = U(x) + \lambda V(x)$ for $\lambda = 0, 0.5, 0.9$

Theorem 7 (Explored area)**.** *If the heuristic satisfies $V(x) \leq d(x_1, x)$, then the geodesic $\gamma \in \mathcal{P}_1(x_0, x_1)$ between x_0 and x_1 satisfies*

$$\{\gamma(t) \setminus t \in [0,1]\} \subset \{x \setminus \mathcal{P}(x) = U(x) + V(x) \leq \mathcal{P}(x_1)\}.$$

This theorem shows why it is important to estimate the geodesic distance by below, since otherwise the region explored by the algorithm might not contain the true geodesic.

Figure 2.16 shows examples of heuristics that approximate the true remaining distance by bellow. One can see how the explored area of the propagation progressively shrinks while containing the true geodesic. Such a heuristic is however impossible to use in practice since one does not have direct access to the remaining distance during the propagation.

Many strategies can be used to estimate a heuristic. For instance, on a Riemannian metric $(\mathcal{M}, H(x))$, one could use

$$V(x) = \rho \|x - x_1\| \qquad \text{where} \qquad \rho = \min_{x \neq 0, v \neq 0} \|v\|_{H(x)}.$$

In this case, ρ is the minimum eigenvalue of all the tensors $H(x)$. This heuristic estimates the geodesic distance with a Euclidean distance and satisfies $V(x) \leq d(x_1, x)$.

For a propagation on a graph (A* algorithm) that is embedded in Euclidean space according to $i \in V \mapsto x_i \in \mathbb{R}^k$, one could also define

$$\forall\, i \in V, \quad V(i) = \|x_{i_1} - x_i\|,$$

where i_1 is the index of the ending point. This heuristic also satisfies $V(i) \leq d(i_1, i)$.

These Euclidean heuristics performs poorly on spaces that are not relatively flat. In order to compute more accurate heuristic, we use an expression of the geodesic distance as a minimization.

Theorem 8 (Reversed triangular inequality)**.** *For all $(x, y) \in \mathcal{M}$, one has*

$$d(x,y) = \sup_z \Big(|d(x,z) - d(z,y)| \Big).$$

If one restricts the minimum to a small subset of landmark points $\{z_1,\ldots,z_n\} \subset \mathcal{M}$, one can define the following approximate distance

$$\tilde{d}_{z_1\ldots z_n}(x,y) = \sup_{k=1\ldots n} \Big(|d_k(x) - d_k(y)| \Big),$$

This kind of approximation has been used first in graph theory [20] and it is defined in a continuous setting in [21]. This leads to a heuristic $V(x) = \tilde{d}(x,x_1)$ that has the following properties.

Theorem 9 (Convergence of heuristic)**.** *One has* $\tilde{d} \leq d$ *and* $\tilde{d} \overset{n\to+\infty}{\longrightarrow} d$.

In a numerical application that requires the extraction of many geodesics in real time over a large domain, one can pre-compute (off-line) the set of distance maps to the landmarks $\{d(x,z_i)\}_{i=1}^m$. At run time, this set of distances is used to compute the heuristic and speed up the propagation. Figure 2.17 shows how the quality of the heuristic increases with the number of landmarks. Figure 2.18 shows an application to geodesic extraction on 3D meshes.

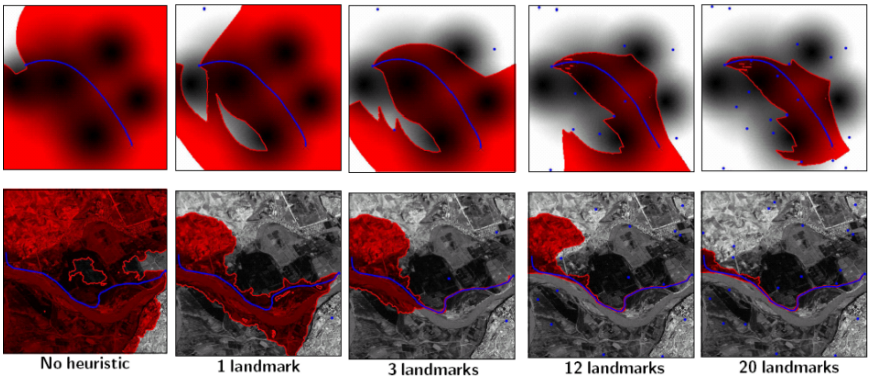


Fig. 2.17 Heuristically driven propagation in 2D with an increasing number of landmark points

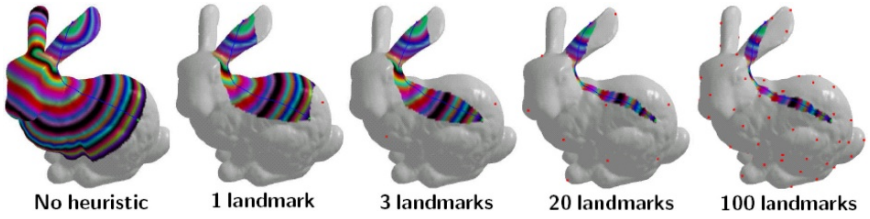


Fig. 2.18 Heuristically driven propagation on a 3D mesh with landmark points

2.4 Surface Sampling

In order to acquire discrete samples from a continuous surface, or to reduce the number of samples of an already acquired mesh, it is important to be able to seed evenly a set of points on a surface. This is relevant in numerical analysis to have a good accuracy in computational simulations, or in computer graphics to display 3D models with a low number of polygons. In practice, one typically wants to enforce that the samples are approximately at the same distance from each other. The numerical computation of geodesic distances is thus a central tool, that we are going to use both to produce the sampling and to estimate the connectivity of a triangular mesh.

2.4.1 Farthest Point Sampling

A sampling of a Riemannian surface $\mathcal{M}$ is a set of points $\{x_1,\dots,x_n\} \subset \mathcal{M}$. If the surface is parameterized by $\phi : [0,1]^2 \mapsto \mathcal{M}$, the easiest way to compute a sampling is to seed points regularly over the parametric domain

$$\forall (i,j) \in \{1,\dots,\sqrt{n}\}^2, \quad x_{i,j} = \phi(i/\sqrt{n}, j/\sqrt{n}).$$

This strategy performs poorly if the mapping ϕ introduces heavy geodesic distortion and the sampling might not be regular any more for the geodesic metric on the surface. In order to ensure the quality of a sampling, one can use the notion of a well separated covering.

Definition 15 (ε-covering)**.** *A sampling $\{x_1,\dots,x_n\} \subset \mathcal{M}$ is an ε-covering if*

$$\bigcup_i B_\varepsilon(x_i) = \mathcal{M} \qquad \textit{where} \qquad B_\varepsilon(x) \stackrel{\text{def.}}{=} \{y \setminus d_{\mathcal{M}}(x,y) \leq \varepsilon\}.$$

Definition 16 (ε-separated)**.** *A sampling $\{x_1,\dots,x_n\} \subset \mathcal{M}$ is ε-separated if*

$$\max(d_{\mathcal{M}}(x_i,x_j)) \leq \varepsilon.$$

The farthest point sampling algorithm is a simple greedy strategy able to produce quickly a good sampling. This algorithm has been introduced in image processing to perform image approximation [22]. It is used in [23] together with geodesic Delaunay triangulation (to be defined in the next section) to do surface remeshing. The detection of saddle points (local maxima of the geodesic distance) is used in [24] to perform perceptual grouping.

Table 2.3 gives the details of this iterative algorithm. In particular, note that the update of the distance $d(x)$ to the set of already seeded points goes faster at each iteration since the domain of update is smaller when the number of points increases.

The output sampling of the algorithm enjoys the property of being a well separated covering of the manifold.

Table 2.3 Farthest point sampling algorithm.

1. *Initialization:* $x_1 \leftarrow$random, $d(x) \leftarrow d_{\mathcal{M}}(x_1,x)$, set $i=1$.
2. *Select point:* $x_{i+1} = \operatorname{argmax}_x d(x)$, $\varepsilon = d(x_{i+1})$.
3. *Local update of the distance:* $d(x) \leftarrow \min(d(x), d_{\mathcal{M}}(x_{i+1},x))$.
   This update is restricted to the set of points $\{x \setminus d_{\mathcal{M}}(x_{i+1},x) < d(x)\}$.
4. *Stop:* If $i < n$ or $\varepsilon > \varepsilon_0$, set $i \leftarrow i+1$ and go back to 2.

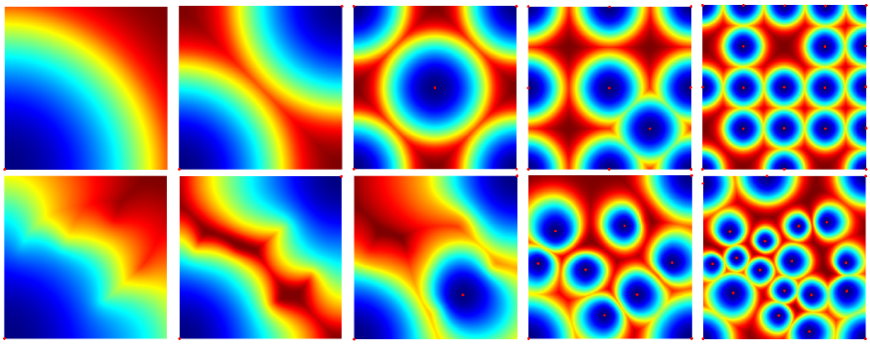


Fig. 2.19 Examples of farthest point sampling (the colormap indicates the distance function to the seeds)

Theorem 10 (Farthest seeding properties)**.** *The farthest point sampling* $\{x_1,\ldots,x_n\}$ *is an* ε*-covering that is* ε*-separated for*

$$\varepsilon = \max_{i=1,\ldots,n} \min_{j=1,\ldots,n} d_{\mathcal{M}}(x_i,x_j).$$

Note however that there is no simple control on the actual number of samples n required to achive a given accuracy ε. We refer to [25] for an in-depth study of the approximation power of this greedy sampling scheme.

Figure 2.19 shows examples of farthest point sampling with a uniform (top row) and a spatially varying isotropic metric $W(x)$ (bottom row). One can see that this scheme seeds more points in areas where the metric W is large. One can thus control the sampling density by modifying the metric W.

2.4.2 Triangulations

Having computed, for instance with farthest points, a sampling $\{x_i\}_{i\in V} \subset \mathcal{M}$, the next step is to compute some connectivity between the samples in order to build a graph, or even better, a triangulation. The problem of surface remeshing has been studied extensively in computer graphics, see the survey [26]. This section explains a solution based on the geodesic Delaunay triangulation.

The following definition generalizes the notion of an Euclidean Voronoi diagram, to an arbitrary surface.

Definition 17 (Voronoi segmentation)**.** *The Voronoi segmentation of a sampling* $\{x_i\}_{i\in V} \subset \mathcal{M}$ *is*

$$\mathcal{M} = \bigcup_i V_i \qquad \text{with} \qquad (V_i)_{i=1}^m \overset{\text{def.}}{=} \text{Voronoi}_{\mathcal{M}}(\{x_i\}_i)$$

where

$$V_i \overset{\text{def.}}{=} \{x \setminus \forall\, j \neq i,\ d_{\mathcal{M}}(x,x_i) \leq d_{\mathcal{M}}(x,x_j).\}$$

Each Voronoi cell V_i is thus composed of points that are closer to x_i than to any other sampling point. The boundary between two adjacent cells V_i and V_j is thus a piece of curve at equal distance between x_i and x_j. One can then compute the graph dual to a given partition, which joins together pair of adjacent cells. This leads to the notion of Delaunay graph.

Definition 18 (Geodesic Delaunay graph)**.** *The Delaunay graph* (V,E) *of a sampling* $\{x_i\}_{i\in V} \subset \mathcal{M}$ *is defined for* $V = \{1,\ldots,n\}$ *as*

$$E = \{(i,j) \in V \setminus \partial V_i \cap \partial V_j \neq 0\}.$$

The main interest of this Delaunay graph is that, if the number of points is large enough to capture the topology of the surface (for instance at least four points are needed on a sphere), then one gets a valid triangulation.

Theorem 11. *For a large enough number of points, the Delaunay graph is a valid triangulation.*

This theorem means that one can find a set of faces F such that (V,E,F) is a triangulated mesh. One can see [27] for a theoretical study of geodesic Delaunay triangulations.

2.4.3 Examples of Meshing and Remeshing

This Delaunay triangulation can thus be used to perform a geodesic meshing or re-meshing of any Riemannian surface, as explained in [23].

Figure 2.20 shows examples of Voronoi segmentations on the plane for various isotropic Riemannian metrics $W(x)$. The Delaunay graph allows to define a planar mesh of points evenly sampled according to the metric. Figure 2.21 shows examples of Voronoi cells on a surface embedded in $\mathbb{R}^3$.

Instead of using a constant or an isotropic metric $W(x)$, one can use a fully anisotropic metric $H(x) \in \mathbb{R}^{2\times 2}$. The local dominant eigenvector $e_1(x)$ given in the decomposition (2.1) of the tensor gives the local preferred direction of the triangles and the anisotropy $\lambda_1(x)/\lambda_2(x)$ describe how much the triangles should be

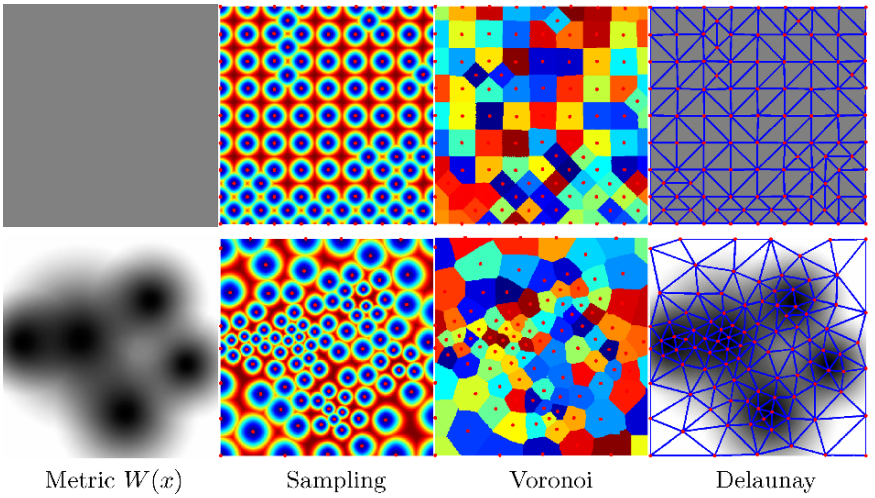


Fig. 2.20 Examples of sampling and triangulations with an isotropic metric $H(x) = W(x)^2 \mathrm{Id}_2$. The sampling is denser in the regions where the metric is small (dark)

Fig. 2.21 Example of Voronoi segmentations for an increasing number of seeding points

stretched in this direction. Figure 2.22 shows an example of meshing with a metric of decreasing anisotropy. Figure 2.23 shows an anisotropic farthest point meshing with an increasing number of sampling points.

In order to mesh the interior of a planar shape $S \subset \mathbb{R}^2$, one can use the Euclidean metric inside the shape and compute a geodesic Delaunay triangulation. Some care should be made during the algorithm so that the boundary of the domain is included in the delaunay triangulation. This requires splitting boundary edges if they disap-

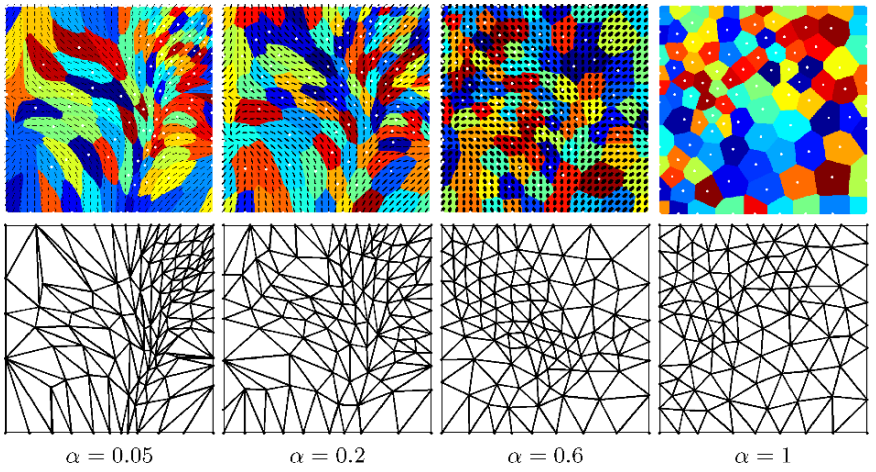


Fig. 2.22 Meshing of a square with a metric of decreasing anisotropy of a same synthetic tensor field. Top: Voronoi diagrams, tensor fields and points added by the algorithm (last image is the Euclidean case). Bottom: resulting meshes

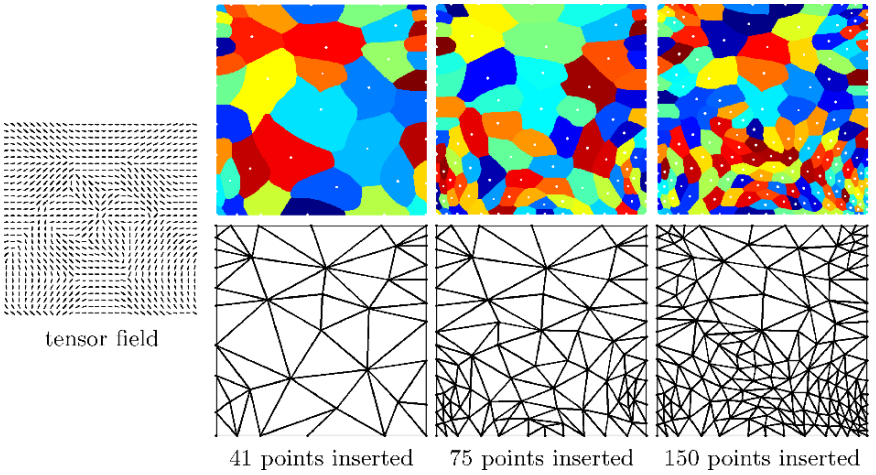


Fig. 2.23 Anisotropic meshing of a square with an increasing number of points

pear from the Delaunay graph during the algorithm. Figure 2.24 shows some examples of shape meshing with this uniform metric. This triangulation is however very close to the usual definition of a planar Euclidean Delaunay triangulation. In contrast, one can use a non-uniform metric $W(x)$ and compute a sampling inside the shape that conforms itself to this density. Figure 2.24 shows a sampling and meshing that uses a metric $W(x) = (\varepsilon + d(x, \partial S))^{-1}$ that tends to seed more points on the boundary of the shape S.

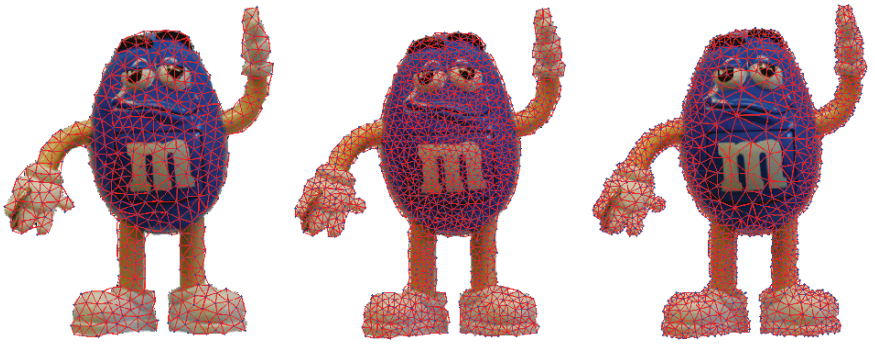


Fig. 2.24 Shape meshing with an increasing number of points. Left and center: uniform meshing, right: adaptive meshing

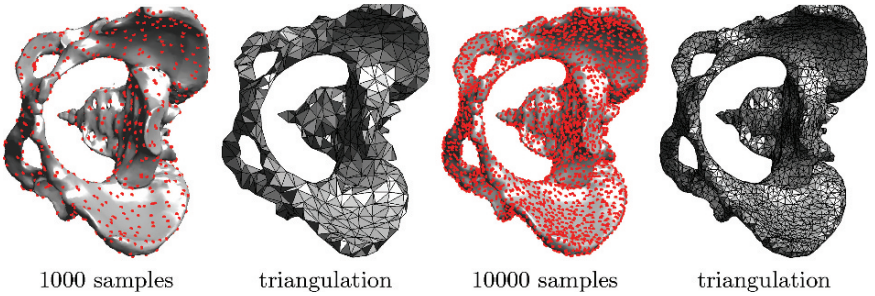


Fig. 2.25 Geodesic remeshing with an increasing number of points

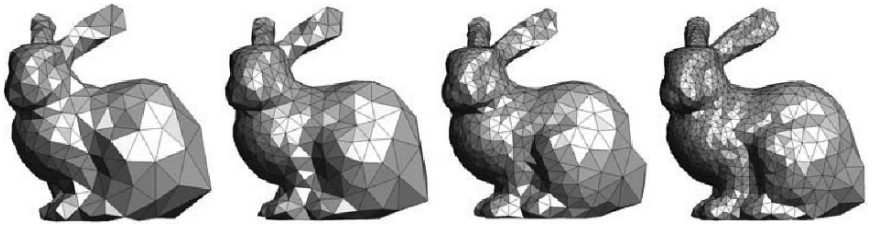


Fig. 2.26 Adaptive remeshing with a linearly increasing density

Figure 2.25 shows an example of uniform remeshing of a 3D surface acquired from medical imaging with an increasing number of points. Figure 2.26 shows how one can adapt the density by defining a non-constant isotropic metric on the surface.

An option to compute this metric is to use a texture mapped on the surface. Starting from some parametric surface: $\phi : \mathcal{D} \subset [0,1]^2 \rightarrow \mathcal{M}$, a texture T is a mapping $T : [0,1]^2 \rightarrow \mathbb{R}$. It allows to define an isotropic metric using for instance an edge adaptive function

$$\forall x \in \mathcal{D},\ H(x) = \psi_T(x)\mathrm{Id}_2.$$

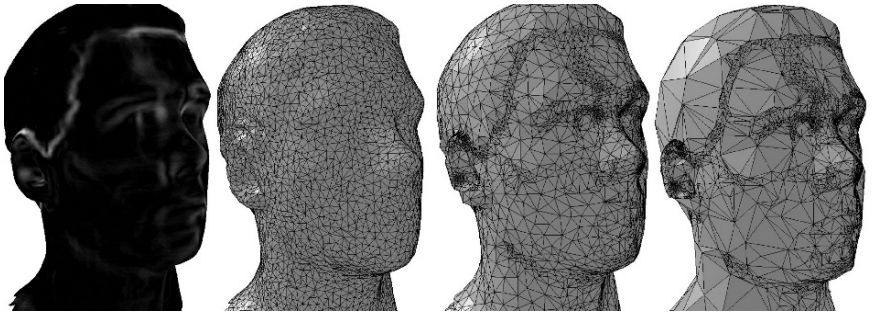


Fig. 2.27 Adaptive remeshing with a density given by a texture

where the edge-based stopping function is $\psi_T(x) = (\|\nabla_x T\| + \varepsilon)^{-1}$. Figure 2.27 shows examples of remeshing with a texture-adapted metric with a decreasing value of ε (increasing adaptivity).

2.5 Conclusion

This chapter has reviewed several applications of Riemannian metrics in computer vision and graphics. In particular, the use of geodesic distances and shortest paths is useful in many areas of these fields. The design of adapted isotropic or anisotropic metrics allows to solve efficiently segmentation, sampling, meshing and recognition problems with fast algorithms.

References

1. Cohen, L.D.: Minimal paths and fast marching methods for image analysis. In: Handbook of Mathematical Methods in Computer Vision, N. Paragios and Y. Chen and O. Faugeras Editors, Springer, Berlin (2005)
2. Prados, E., Lenglet, C., Pons, J.P., Wotawa, N., Deriche, R., Faugeras, O.D., Soatto, S.: Control theory and fast marching methods for brain connectivity mapping. In: Proc. CVPR 2006. (2006)
3. Crandall, M.G., Ishii, H., Lions, P.L.: User's guide to viscosity solutions of second order partial differential equations. Bull. Amer. Math. Soc **27**(1992) 1
4. Cohen, L.D., Kimmel, R.: Global minimum for active contour models: a minimal path approach. International Journal of Computer Vision **24**(1) (Aug. 1997) 57–78
5. Sethian, J.: Level Sets Methods and Fast Marching Methods. 2nd edn. Cambridge University Press, Cambridge (1999)
6. Osher, S.J., Fedkiw, R.: Level Set Methods and Dynamic Implicit Surfaces. Springer, New York (2002)
7. Kimmel, R.: Numerical Geometry of Images: Theory, Algorithms, and Applications. Springer, Berlin New York (2004)
8. Bronstein, A., Bronstein, M., Kimmel, R.: Numerical Geometry of Non-Rigid Shapes. Springer-Verlag New York (2007)

9. Osher, S.J., Paragios, N.: Geometric Level Set Methods in Imaging, Vision, and Graphics. Springer New York (July 2003)
10. Paragios, N., Chen, Y., Faugeras, O.D.: Handbook of Mathematical Models in Computer Vision. Springer-Verlag New York (2005)
11. Kimmel, R., Sethian, J.: Computing geodesic paths on manifolds. Proc. Natl. Acad. Sci. **95**(15) (1998) 8431–8435
12. Spira, A., Kimmel, R.: An efficient solution to the eikonal equation. Interfaces Free Boundaries **6**(3) (2004) 315–327
13. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Weighted distance maps computation on parametric three-dimensional manifolds. J. Comput. Phys., accepted (2007)
14. Kimmel, R., Amir, A., Bruckstein, A.M.: Finding shortest paths on surfaces using level sets propagation. IEEE Trans. PAMI **17**(6) (1995) 635–640
15. Deschamps, T., Cohen, L.: Fast Extraction of Minimal Paths in 3D Images and Applications to Virtual Endoscopy. Med. Image Anal. **5**(4) (December 2001)
16. Kothe, U.: Edge and junction detection with an improved structure tensor. In: Proc. DAGM03. (2003) 25–32
17. Ion, A., Peyré, G., Haxhimusa, Y., Peltier, S., Kropatsch, W.G., Cohen, L.: Shape matching using the geodesic eccentricity transform. In: Proc. OAGM'07. (2007)
18. Ling, H., Jacobs, D.W.: Using the inner-distance for classification of articulated shapes. In: CVPR 2005, 20–26 June 2005, San Diego, CA. (2005) 719–726
19. Nilsson, N.: Problem-solving Methods in Artificial Intelligence. McGraw-Hill, New York (1971)
20. Goldberg, A.V., Harrelson, C.: Computing the shortest path: A* search meets graph theory. Technical Report MSR-TR-2004-24 (2004)
21. Peyré, G., Cohen, L.: Heuristically driven front propagation for fast geodesic extraction. Int. J. Comput. Vision Biomech. **1**(1) (2007)
22. Eldar, Y., Lindenbaum, M., Porat, M., Zeevi, Y.Y.: The farthest point strategy for progressive image sampling. IEEE Trans. Image Process. **6**(9) (September 1997) 1305–1315
23. Peyré, G., Cohen, L.D.: Geodesic remeshing using front propagation. Int. J. Comput. Vision **69**(1) (2006) 145–156
24. Cohen, L.: Multiple Contour Finding and Perceptual Grouping Using Minimal Paths. J. Math. Imaging Vis. **14**(3) (May 2001) 225–236
25. Clarkson, K.L.: Building triangulations using epsilon-nets. In Kleinberg, J.M., ed.: STOC, ACM (2006) 326–335
26. Alliez, P., Ucelli, G., Gotsman, C., Attene, M.: Recent advances in remeshing of surfaces. In: AIM@SHAPE repport. (2005)
27. Leibon, G., Letscher, D.: Delaunay triangulations and voronoi diagrams for riemannian manifolds. In: SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry, New York, NY, ACM (2000) 341–349

Chapter 4
Digital Geometry and Its Applications to Medical Imaging

Reneta P. Barneva and Valentin E. Brimkov

Abstract Digital geometry is a modern discipline dealing with geometric properties of *digital objects* (also called *digital pictures*). These are usually modeled as sets of points with integer coordinates representing the pixels/voxels of the considered objects. Digital geometry is developed with the expectation that it would provide an adequate mathematical background for new advanced approaches and algorithms for various problems arising in image analysis and processing, computer graphics, medical imaging, and other areas of visual computing. In this chapter we first provide a brief discussion on the motivation, basic directions, and achievements of digital geometry. Then we consider typical examples of research problems and their solutions. We focus our attention on problems related to digital manifolds. The latter play an important role in computer graphics, 3D image analysis, volume modeling, process visualization, and so forth — in short, in all areas where discrete multidimensional data need to be represented, visualized, processed, or analyzed. The objects in these areas often represent surfaces and volumes of real objects. We discuss some applications of digital curves and surfaces to medical imaging, implied by theoretical results on digital manifolds.

4.1 Introduction

In this introductory section we briefly discuss about the motivation, basic directions, and goals of digital geometry.

R.P. Barneva
SUNY Fredonia, Fredonia, NY 14063
e-mail: reneta.barneva@fredonia.edu

V.E. Brimkov
SUNY Buffalo State College, Buffalo, NY 14222
e-mail: brimkove@buffalostate.edu

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

4.1.1 What is Digital Geometry?

Digital geometry deals with geometric properties of *digital objects* (also called *digital pictures*). These are usually modeled as sets of points with integer coordinates representing the pixels/voxels of the considered digital objects. Digital geometry has established itself as an independent discipline comparatively recently, in the second half of the 20th century, with the initiation of research in visual computing. The latter includes various applied areas such as image analysis and processing, computer vision, computer graphics, medical imaging and, more recently, multimedia technologies. The nature of the used research approaches and the obtained results put digital geometry on the border of applied mathematics and theoretical computer science, as the framework of the performed research is determined by practical applications in mind.

Digital geometry is developed with the expectation to provide an adequate theoretical (mathematical) background for new advanced approaches to and algorithms for solving various problems arising in visual computing.

4.1.2 Why Digital Geometry?

In general, the development of digital geometry follows the one of classical geometry. The latter has appeared in the remote past as a collection of practical computation rules helpful for resolving certain everyday problems. Only much later, since Euclid, it starts turning to a rigorous mathematical subject. Over the centuries, new practical tasks motivate the rise of new geometries, such as analytical geometry, differential geometry, and, more recently, computational geometry, to mention just a few. Most of these belong to the continuous domain of mathematics, with a few exceptions led by specific applications (e.g., combinatorial geometry and certain finite geometries).

In recent decades, the development of various branches of visual computing poses new challenges to the researchers. The objects operated in computer graphics, image analysis and processing are discrete sets of points. However, as a rule, continuous mathematics (in particular, classical geometry) is used for modeling and problem solving (for instance, in most works available in the SIGGRAPH volumes). Comparatively more rarely, ad-hoc algorithms are used for direct processing of discrete data.

As the computer images are discrete, it is quite natural the geometry involved to be discrete as well. Despite the presence of a lot of results, one should admit that a theory that could perfectly serve as a discrete analog of Euclidean geometry, is not completely developed yet. There is a simple reason for that: development of such a theory is hard, due to the discrete nature of the objects involved. In particular, this may cause ambiguity when one looks for the most reasonable definitions of even very basic discrete primitives, such as straight lines, circles, planes, etc. For instance, several definitions of a digital straight line are available in the literature

(see, e.g., [48]), each of which has advantages and disadvantages to the others. Certain paradoxes that do not exist in continuous spaces are also possible. See, e.g., examples in Section 1.1.4 of [28].

It should also be mentioned that, as far as a theory of digital geometry exists, it is not very well-known and is rarely used by software developers. This is conditioned by a number of reasons. Pretty often, the insufficient mathematical background of the programmers does not allow them understand and apply advanced approaches involving more sophisticated technical machinery. It is surprising that the above can sometimes be observed even in computing laboratories of some very reputable organizations. Moreover, the industry usually requires to produce software in very short periods of time without serious theoretical research. Unfortunately, in many cases this leads to lowering the quality in terms of time and memory efficiency of the developed algorithms and of the accuracy and reliability of the obtained solutions. Sometimes this may indeed be an important shortcoming (e.g., in medical applications).

In view of the preceding discussion, digital geometry is aimed at becoming a rigorous theory that serves as a universal tool for modeling and resolving various problems. The theory should be easily applicable. It is also clear that digital geometry would be of little use if it remains unknown to those for whom it is created. Thus an important task is to make it more popular by means of systematic education.

4.1.3 Mathematics of Digital Geometry

As already mentioned, digital geometry is a modern discipline that sets up itself as such in relation to its contemporary applications. However, it has its roots in a number of classical mathematical disciplines, such as number theory (since C.F. Gauss), geometry of numbers (since H. Minkowski), graph theory (since L. Euler), and combinatorial topology (since the middle of the 19th century). At present, research in digital geometry resorts to the above and some other mathematical disciplines. A more complete (although not exhaustive) list is given next.

- Number theory, geometry of numbers
- Classical Euclidean geometry, analytical geometry, affine geometry, projective geometry
- Algebraic geometry
- Vector spaces, metric spaces
- Combinatorial geometry, discrete geometry, tilings and patterns
- Computational geometry
- General topology, combinatorial topology
- Graph theory
- Linear programming, integer programming, Diophantine equations, polyhedral combinatorics, lattice polytopes
- Mathematical morphology

- Discrete dynamical systems, fractal theory
- Combinatorics on words
- Approximation theory, Diophantine approximations, continued fractions
- Probability theory and mathematical statistics
- Design and analysis of algorithms, complexity theory

Knowledge and approaches from the above-listed subjects are used to obtain theoretical results and design algorithms for solving various specific problems. Occasionally, results of digital geometry turn out to be known in different terms in the framework of earlier studies. Overall, however, digital geometry has provided a lot of new results, some of which are not only useful regarding practical applications, but also technically sound and deep from mathematical point of view.

4.1.4 Main Directions of Digital Geometry

Digital geometry is germane with discrete geometry that deals with similar and some other related matters from a bit more general perspective (see the topics of Mathematical Subject Classification number 52Cxx). In particular, discrete geometry includes a number of subjects (e.g., the ones related to matroid theory) that are not directly related to computer imagery, and tackles them from a more abstract point of view. Instead, digital geometry is closely focused on problems arising from image analysis and processing, computer graphics, and related disciplines. Below we list some basic subjects of digital geometry, among others.

Digital topology
Digital topologies (classification), topology of digital objects (basic topological invariants of curves and surfaces, topology of digital curves and surfaces, topology of linear digital objects)

Geometry of digital manifolds
Geometry of digital curves and surfaces, digital straightness in 2D and 3D, digital planarity, length and curvature of digital arcs, area and curvature of digital surfaces, digital convexity

Transformations
Axiomatic digital geometry, transformation groups and symmetries, neighborhood-preserving transformations, magnification and demagnification

Discrete tomography

Morphologic operations
Dilation, erosion, simplification, segmentation, decomposition

Deformations
Topology-preserving deformations, shrinking, thinning, deformations of curves, 3D pictures, and multivalued pictures

Picture properties
Moments, operations on pictures, invariant properties, spatial relations

For detailed presentation of these and other areas of digital geometry the reader is referred to the recent monograph [28]. In the rest of this chapter we review some actual or possible applications in medical imaging implied by theoretical results on digital manifolds. These applications include visualization of a digitized real object, identification of its topological or geometric properties (such as its tunnels, gaps, skeleton, or boundary), as well as certain metric properties. In Section 4.2 we refer to works providing a theoretical basis for the above-mentioned applications that are discussed in Section 4.3. We conclude with some remarks in Section 4.4. An extensive bibliography is provided to facilitate interested readers.

4.2 Digital Manifolds

Digital manifolds play an important role in various facets of the modern information society. By becoming a "digital society," the complexity of synthetic digital worlds is increasing. They often represent surfaces and volumes of real objects. This is for example the case in such fields as medicine (e.g., organ and tumor measurements in CT images, beating heart, or lung simulations), bioinformatics (e.g., protein binding simulations), robotics (e.g., motion planning), engineering (e.g., finite elements stress simulations), and security (biometrics). With the rapidly growing variety of synthetic surfaces and volumes, it is becoming critical to develop a relevant theory of digital manifolds and based on it methods for resolving a wide range of problems.

Theory of digital manifolds is a vivid topic of research. In this section we first briefly list and comment some literature sources containing recent developments on the subject. Then we introduce several notions playing an important role in research and related to applications presented in the subsequent sections.

4.2.1 Research on Digital Curves and Surfaces

Before providing a brief overview of results on digital curves and surfaces, we recall a few basic notions. Two 3-cells (voxels) c_1 and c_2 are called α-*adjacent* iff their intersection $c_1 \cap c_2$ contains an α-cell, where $\alpha \in \{0,1,2\}$. Alternatively, two grid points $p_1, p_2 \in \mathbb{Z}^3$ are called 6-*adjacent* iff $0 < d_e(p_1,p_2) \leq 1$, 18-*adjacent* iff $0 < d_e(p_1,p_2) \leq \sqrt{2}$, and 26-*adjacent* iff $0 < d_e(p_1,p_2) \leq \sqrt{3}$, where d_e is the Euclidean distance.

Digital surfaces have been studied frequently over the years. For example, [30] defines digital surfaces in $\mathbb{Z}^3$ based on adjacencies of 3-cells. A mathematical framework (based on a notion of "moves") for defining and processing digital manifolds is proposed in [12]. For obtaining α-surfaces by digitization of surfaces in $\mathbb{R}^3$, see [15]. It is proved in [40] that there is no local characterization of a 26-connected subset S of $\mathbb{Z}^3$ such that its complement $\overline{S}$ consists of two 6-components and every voxel of S is adjacent to both of these components. Reference [40] defines a class of

18-connected surfaces in $\mathbb{Z}^3$, proves a surface separation theorem for those surfaces, and studies their relationship to the surfaces defined in [41]. Reference [3] introduces a class of "strong" surfaces and proves that both the 26-connected surfaces of [41] and the 18-connected surfaces of [40] are strong. For further studies on 6-surfaces, see [11]. Digital surfaces in the context of arithmetic geometry are studied in [5]. For various other topics related to digital manifolds we also refer to [9,10].

A recent paper [7] provided the first definition of digital manifolds involving the notion of dimension in discrete spaces [42]. Accordingly, a digital curve is one dimensional while a digital surface is $(n-1)$-dimensional set of voxels, where n is the dimension of the considered discrete space. The definition allows classification of all digital manifolds with respect to the type of their "gaps." The concepts of tunnels and gaps and their relevance to certain practical problems is discussed next.

4.2.2 Tunnels, Gaps, and Skeletons

A *gap* is an important notion in discrete geometry and topology. Usually, gaps are defined through separability as follows: Let a digital object M be m-separating but not $(m-1)$-separating in a digital object D. Then M is said to have *k-gaps* for any $k < m$. A digital object without m-gaps is called *m-gapfree*. See Fig. 4.1.

Homology groups in topology define *tunnels*, and 2-gaps are sometimes also discussed as being tunnels. Information about the number of gaps or tunnels has been a subject of interest in various disciplines, such as digital topology [19, 39, 43, 51], image analysis [32, 37, 49], graph theory [52], and computational modeling of 3D forms [17]. Gaps or tunnels are related to important topological concepts such as Euler characteristic and Betti numbers. See [28] for more details.

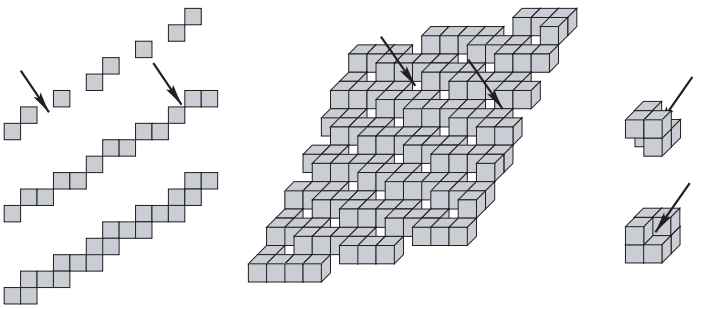


Fig. 4.1 *Left:* From top to bottom: portions of digital lines defined by $0 \le 3x - 5y < 3$, $0 \le 3x - 5y < 5$, and $0 \le 3x - 5y < 8$. The first one has 1-gaps (and, therefore, also 0-gaps; a 1-gap is pointed out by an arrow), the second one has 0-gaps (one of them pointed out by an arrow) but no 1-gaps, and the third one is gap-free. *Middle:* Portion of a digital plane defined by $0 \le 2x + 5y + 9z < 7$. It has 2-gaps (and, therefore, also 1- and 0-gaps). A 2-gap and a 1-gap are pointed out by arrows. *Right:* Configuration of voxels (in two different orientations) that features a 0-gap (pointed out by an arrow)

For various applications it is useful to obtain the *skeleton* of a digital set. Skeletons represent the basic topological features of the considered object while being easier to study. They are obtained by thinning algorithms. For more details refer, e.g., to [25–27, 31, 44–46].

4.3 Applications to Visualization, Processing, and Structural Analysis of Digitized Objects

In this section we briefly discuss possible applications of digital manifolds, mainly in the area of medical imaging.

4.3.1 Finding and Counting Gaps

Knowledge about gaps is important for ray tracing or understanding of the topology of digitized 3D sets (see [21–24]). Assume, for example, that an unknown closed continuous surface Γ has been digitized, e.g., by a tomography scanner. Let M be the resulting digital set of voxels. Let the border $\partial(M)$ of M be determined in a way to constitute a digital surface. The requirement for gap-freeness of $\partial(M)$ is important when a discrete model of a surface is traced through digital rays (e.g., for visualization or illumination purposes), since the penetration of a ray through the surface causes a false hole in it. Knowledge about the type of gaps of $\partial(M)$ may predetermine the usage of an appropriate type of digital rays for tracing the border in order to avoid wrong conclusions about the topology of the original continuous 3D set having the frontier Γ. Then, for the purposes of surface reconstruction, one will be able to faithfully model the geometry of the original 3D set. This is of importance for 3D imaging, e.g., in medicine.

Information about gaps is also important for ensuring correctness of representation for simulation purposes. For example, a small hole in a heart surface created by imperfections of the synthetic representation, while possibly insignificant (or simply unnoticeable) for visualization, renders the synthetic surface useless for blood flow simulation. Further, finite element simulations may yield incorrect results if surfaces have singularities. Therefore, it is of primary importance to have sound mathematical methods that can assure correctness of key topological, geometric, and metric properties of synthetic surfaces and volumes.

In [8] Brimkov et al. generalize the notion of gap to higher dimensions. The following formula for the number of $(n-2)$-dimensional gaps in a digital object S has been obtained. Let S_k be the set of k-cells of S and $s_i = |S_i|$, $0 \leq k \leq n$. Then

$$g_{n-2} = -2n(n-1)s_n + 2(n-1)s_{n-1} - s_{n-2} + b,$$

where b is the number of $2^2 1^{n-2}$-blocks of S (see [8] for denotations, definitions, and other details). In particular, the above formula counts the number of 0-gaps and

1-gaps in digital 2D/3D digital objects. A computer program (based on simple linear time algorithm) has been designed to compute the number of 0- and 1-gaps as well as other object parameters. The program also allows to visualize the digital picture S and interactively rotate it along the Ox-, Oy-, and Oz- axes so that the object can be seen from different viewpoints.

4.3.2 Number of Tunnels

Several works address the more difficult and equally important problem of computing the number of tunnels in a digital object. An algorithm from [49] computes the number of tunnels in a $3 \times 3 \times 3$ neighborhood of a point but not for the whole region. Several other works [1, 2, 13, 18, 20, 47] provide algorithms for the problem, however, with no estimation of the computational complexity.

Using a graph-theoretical approach, in [36] the authors present a computationally efficient algorithm with a guaranteed polynomial worst case running time. There is an evidence that the same approach could provide an algorithm to compute homology for digitized sets in arbitrary dimension.

4.3.3 Visualization, Skeletonization, and Measurements

Some theoretical developments related to digital manifolds are particularly relevant to the analysis of curve-like structures in biomedical images. An ongoing research project [25] at the University of Auckland aims at analyzing confocal microscope images of human brain tissue (which contain cells called astrocytes, see Fig. 4.2). These images have been taken layer by layer and constitute a volume defined on a 3D regular orthogonal grid. The curve-like structures have been obtained by applying a thinning algorithm (see Fig. 4.3). Reference [25] proposes a classification of voxels in 3D skeletons of binarized volumes for subsequent structural analysis and length measurements of digital arcs. For the former, a specific graph is associated with the skeleton (Fig. 4.4). The nodes of the graph, called *junctions*, exhibit certain interesting properties. However, within the proposed model they are considered as singletons that constitute the set of graph vertices. For the purposes of length measurements, the digital curves are segmented into subsequent maximum-length digital straight-line segments, and the total length of those is used to evaluate the length of the curves. For more details we refer to [25].

Note that the arcs of the skeleton form one-dimensional digital curves and as a whole the skeleton is a digital curve satisfying definitions recently proposed by Brimkov and Klette [7]. These properties support the segmentation process through a number of available efficient algorithms and, in turn, the curve length measurements. Note that curve-like structures appear also in other biomedical images, for example in 3D scans of blood vessels or in 3D ultrasound images.

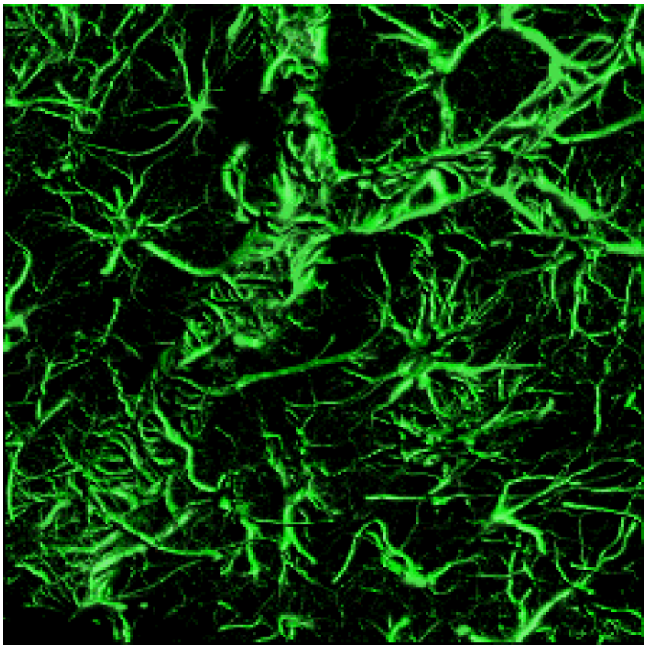


Fig. 4.2 Example of an input data set composed of 42 slices of 256×256 density images generated by confocal microscopy from a sample of human brain tissue (Courtesy of Gisela Klette)

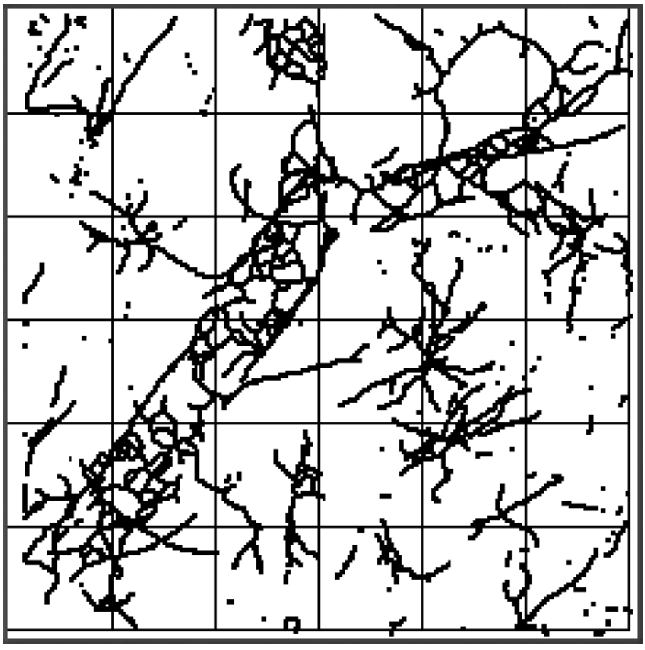


Fig. 4.3 A skeleton of the binarized volume shown in Fig. 4.2 (Courtesy of Gisela Klette)

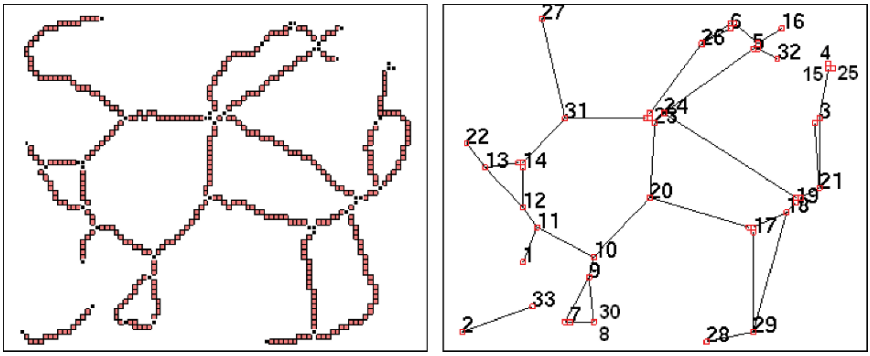


Fig. 4.4 *Left:* Portion of the skeleton from Fig. 4.2 (junctions are shown as small black squares). *Right:* a graph associated with the skeleton from the left. Nodes are labeled by positive integers (Courtesy of Gisela Klette)

4.3.4 Determination of Object Boundary

Another possible application of the theory of digital manifolds is seen in designing algorithms for determining the border of a digital object. Because of its importance, this problem has attracted considerable attention (see, e.g. [16, 33, 35] and the bibliographies therein).

Our hypothesis is that one would benefit from an algorithm that constructs the border as a digital surface as defined in [7]. As already discussed earlier, the reason for this is the knowledge about the gaps in the surface.

If a digital object has been obtained by digitizing a set with a "regular" shape (e.g. featuring convexity), then, in practice, the border voxels indeed constitute a digital surface satisfying those definitions. Moreover, for data compression purposes the obtained digital surface can be "linearized" by partitioning it into polygonal portions of digital planes. The fact that any digital plane is a digital surface explains why in practice the requirement for two-dimensionality supports the minimization of the number of digital plane patches. This problem is considered in more detail in the next section.

In some cases however it is possible that the border voxels of a digital set do not constitute a digital surface. This usually happens when the digital object has a very complex and irregular structure. An illustration of such a complexity is provided in Figs. 4.5 and 4.6. They present digitized images of a human brain tissues, studied within the previously mentioned astrocyte project. In such cases, one possibility is to algorithmically "repair" the set of border voxels in order to make it two-dimensional. Some theoretical results from [7] suggest that such a digitization always exists. Repairing digital objects in order to achieve desired properties has been already used by some researchers (e.g. [35, 50]).

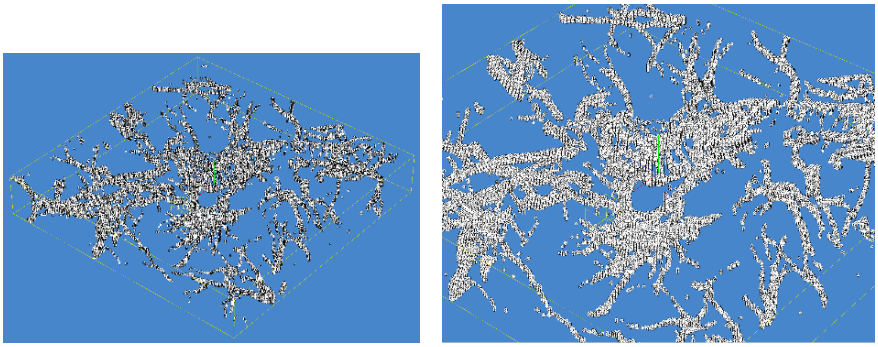


Fig. 4.5 *Left:* Large view of a sample of human brain tissue, studied within the astrocyte project. The data have been obtained by confocal microscopy and visualized in voxel view mode. *Right:* Enlarged view of a detail of the volume on the left

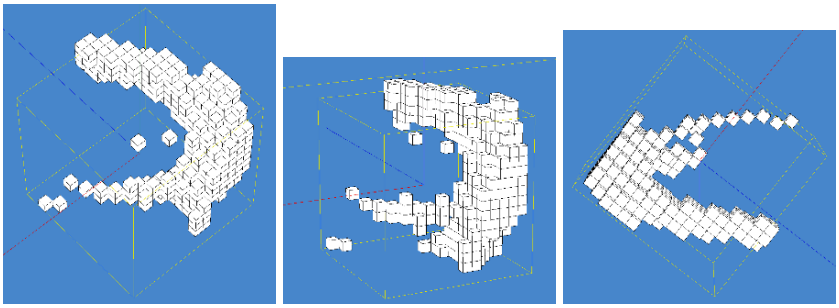


Fig. 4.6 Further enlargements of subvolumes of the digital image of Fig. 4.2

4.3.5 Digital Surface Segmentation

Digital surface segmentation is a fundamental problem in image analysis. The main motivation for it comes from medical imaging and other visualization problems where discrete volumes of voxels result from scanning and MRI techniques. Since digital medical images involve a huge number of points, it is quite problematic to apply traditional rendering or texture algorithms to obtain satisfactory visualization. Moreover, one can face difficulties in storing or transmitting data of that size. There are multiple sources of data being transmitted for many diverse uses, such as telemedicine, mine detection, tele-maintenance, ATR, visual display, cueing, and others. In all these applications the coding compression methodology used is paramount. For this, one can try to transform a discrete data set to a polyhedron, such that the number of its 2-facets is as small as possible. Such polyhedrizations are also searched for the purposes of geometric approximation of surfaces as well as for surface area and volume estimation. Note that the optimization version of this last problem is NP-hard [4].

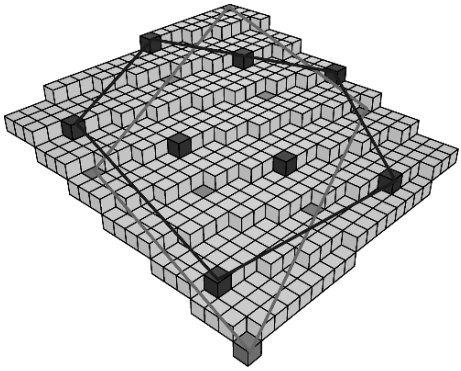


Fig. 4.7 Illustration of a subset of a digital plane $D_{7,17,57,0,57}$ with its lower and upper convex hulls on the supporting planes

A detailed accounting of works devoted to digital surface segmentation is available in a recent survey by Brimkov et al. [6]. Here we briefly comment the incremental algorithm from [29] that appears to be a very efficient one. In order to let ourselves be more specific, we recall some well-known definitions.

A set $D_{a,b,c,\mu,\omega} = \{(i,j,k) \in \mathbb{Z}^3 : \ \mu \leq ai + bj + ck < \mu + \omega\}$ is called a *digital plane* with *normal* $\mathbf{n} = (a,b,c)$, *intercept* μ, and *thickness* ω. If $\omega = \max\{|a|,|b|,|c|\}$, then $D_{a,b,c,\mu,\omega}$ is called a *naive plane*, that is the thinnest hole-free digital plane. A digital plane with $\omega = |a| + |b| + |c|$ is called *standard*. A *digital plane segment* (DPS) is a connected portion of a digital plane. One can define lower (resp. upper) supporting points that determine the lower (resp. upper) supporting continuous planes defining a digital plane (see Fig. 4.7). The preimage of a DPS, S, is the set of planes whose digitizations contain S. It appears to be the solution of a system of linear inequalities with unknowns a, b, and c. Thus, it is a convex polyhedron (possibly empty).

The algorithm from [29] takes advantage of certain geometric properties of digital planes and repeatedly updates a list of supporting planes. The set of points is accepted as a DPS iff the final list of planes is non-empty. The updating step is time-efficient.

One can perform a breadth-first search of the face graph to agglomerate the faces into DPSs. Figure 4.8 illustrates results of the agglomeration process for a digitized sphere and for an ellipsoid with semi-axes 20, 16, and 12. Faces that have the same gray level belong to the same DPS. The respective numbers of faces of the digital surfaces of the sphere and ellipsoid are 7,584 and 4,744. The numbers of DPSs are 285 and 197; the average sizes of these DPSs are 27 and 24 faces.

To complete the polyhedrization process, one sets all the face vertices that are incident to at least three of the DPSs to be vertices of the polyhedron. Fig. 4.9 (left) shows the final polyhedra for the sphere and ellipsoid. Note that these polyhedra are not simple; their surfaces are not hole-free.

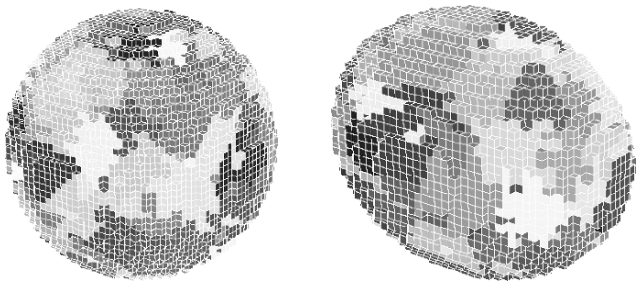


Fig. 4.8 Agglomeration into DPSs of the faces of a sphere and an ellipsoid (grid resolution $h = 40$)

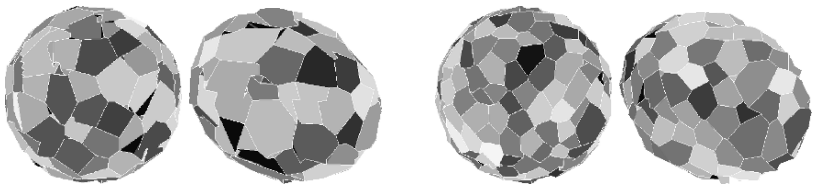


Fig. 4.9 *Two on the left:* polyhedrized sphere and ellipsoid. *Two on the right:* the polyhedrized sphere and ellipsoid when the breadth-first search depth is restricted to 7

Restricting the depth of the breadth-first search changes the polyhedrization from global to local and results in "more uniform" polyhedra. Figure 4.9 (right) shows results when the depth is restricted to 7. The number of small DPSs is reduced and the sizes of the DPSs are more evenly distributed. The respective numbers of DPSs are 282 and 180 and their average sizes are 27 and 26; note that these are nearly the same as in the unrestricted case.

The output of Klette-Sun's algorithm is not, in general, a valid polyhedron but like a *patchwork* of planar segments. It is also desirable to obtain a polyhedron with the following reversibility property: the polyhedron digitization coincides with the originally given set of grid points. An algorithm from [14] addresses the problem of such a reversible polyhedrization.

The main idea is to simplify the polyhedron obtained by a Marching-Cubes (MC) algorithm [38], using information about the digital surface segmentation. The MC algorithm is a widely used isosurface generation algorithm in 3D volume data. This method considers local grid point configurations to replace them by small triangles composing the global isosurface. With a reference to [34], the triangulated surface obtained by the MC algorithm is a combinatorial manifold. In other words, the surface is closed, hole-free and without self-crossing. Furthermore, the object boundary quantization of this polyhedron is exactly the input binary object. See Fig. 4.10.

The output of the algorithm is a digital polyhedron such that a large facet is associated to each recognized DPS. The facets of the polyhedron are stitched together by strips of triangles. These triangles are called *non-homogeneous* in [14] because

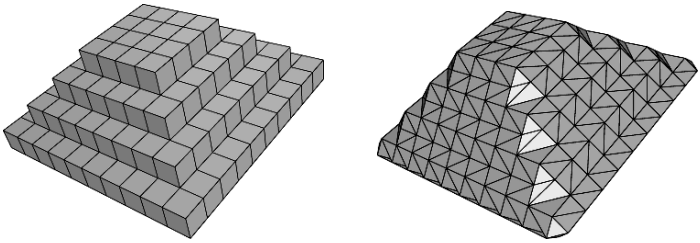


Fig. 4.10 A $\{0,1\}$-binary object and a Marching-Cubes surface obtained with an iso-level in $]0,1[$

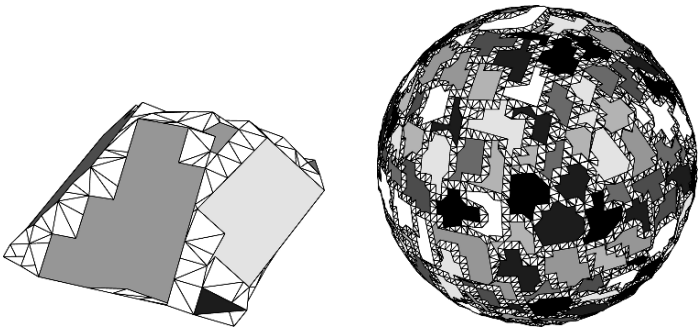


Fig. 4.11 *Left:* Final result on the object of Fig. 4.10. *Right:* Result on a sphere of radius 25

their three vertices do not belong to the same digital plane. The obtained polyhedron is a combinatorial manifold and possesses the reversibility property. See Fig. 4.11.

For more details on the presented problems and algorithms we refer to the recent survey on digital planarity [6].

4.4 Concluding Remarks

One of the purposes of this chapter was to introduce the reader to the subject of digital geometry. In the last two decades the interest to the latter in the scientific community is constantly increasing. A number of related conferences provide researchers in the field with opportunities for regular meetings. Industrial interest to the subject is increasing, as well. The authors believe that digital geometry is setting itself as a valuable theoretical foundation for research and software development in all areas of visual computing, in particular in medical imaging.

Another our goal was to present some directions of ongoing research on digital manifolds. This appears to be an important direction of digital geometry with various applications in medical imaging. Mathematically sound foundations may guarantee high quality rendering of objects, faultless simulations (e.g., of organ functions), and computational efficiency of the image analysis and processing. In

order to achieve optimal effect, theoretical research should go in parallel with applied work. Close collaboration between specialist with diverse expertise will become increasingly important.

References

1. Basu S (2006) Journal of Symbolic Computation 41(10):1125–1154
2. Basu S, Pollack R, Roy M (2005) Computing the first Betti number and describing the connected components of semi-algebraic sets. In: Proc. STOC'05
3. Bertrand G, Malgouyres R (1999) Journal of Mathematical Imaging Vision 11:207–221
4. Brimkov VE (2006) Discrete volume polyhedrization: complexity and bounds on performance. In: Tavares JM et al. (eds) CompIMAGE – Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications. Taylor & Francis, London, Leiden, New York, Philadelphia, Singapore
5. Brimkov VE, Andres E, Barneva RP (2002) Pattern Recognition Letters 23:623–636
6. Brimkov VE, Coeurjolly D, Klette R (2007) Discrete Applied Mathematics 155:468-495
7. Brimkov VE, Klette R (2004) Curves, hypersurfaces, and good pairs of adjacency relations. In: Proc. Int. Workshop Combinatorial Image Analysis, LNCS 3322. Springer, Berlin, Heidelberg, New York
8. Brimkov VE, Moroni D, Barneva R (2006) Combinatorial relations for digital pictures. In: Kuba, A et al. (eds), Discrete Geometry for Computer Imagery, LNCS 4245. Springer, Berlin, Heidelberg, New York
9. Chen L (2004) Discrete Surfaces and Manifolds: A Theory of Digital-Discrete Geometry and Topology. Scientific & Practical Computing, Rockville
10. Chen L (2005) Gradually varied surfaces and gradually varied functions. CITR-TR 156. The University of Auckland, Auckland
11. Chen L, Cooley DH, Zhang J (1999) Information Sciences 115:201–220
12. Chen L, Zhang J (1993) Digital manifolds: an intuitive definition and some properties. In: Proc. Symp. Solid Modeling Applications, ACM/SIGGRAPH
13. CHomP (Atlanta) & CAPD (Kraków). Homology algorithms and software. www.math.gatech.edu/chomp/homology/
14. Coeurjolly D, Guillaume A, Sivignon I (2004) Reversible discrete volume polyhedrization using Marching Cubes simplification. In: Proc. Vision Geometry XII, SPIE 5300
15. Cohen-Or D, Kaufman A, Kong TY (1996) On the soundness of surface voxelizations. In: Kong TY, Rosenfeld A (eds), Topological Algorithms for Digital Image Processing, Elsevier, Amsterdam
16. Daragon X, Couprie M, Bertrand G (2005) Journal of Mathematical Imaging and Vision, 23(3):379–399
17. Desbrun M, Kanso E, Kong Y (2005) Discrete differential forms for computational modeling. In: ACM SIGGRAPH 2005 Course Notes on Discrete Differential Geometry, Chapter 7
18. De Silva, Plex V - A Mathlab library for studying simplicial homology. math.stanford.edu/comptop/programs/plex/plexintro.pdf
19. Fourey S, Malgouyres R (2002) Discrete Applied Mathematics 125:59–80
20. Kaczynski T, Mischaikow K, Mrozek M (2004) Applied Mathematical Sciences, Vol. 157. Springer, Berlin
21. Kaufman A (1987) An algorithm for 3D scan-conversion of polygons. In: Proc. Eurographics
22. Kaufman A (1993) Volume Graphics 26(7):51–64
23. Kaufman A (1987) Computer Graphics 21(4):171–179
24. Kaufman A, Shimony E (1986) 3D scan-conversion algorithms for voxel-based graphics. In: Proc. Workshop on Interactive 3D Graphics: ACM, New York

25. Klette G (2006) Branch voxels and junctions in 3D skeletons. In: Reulke R et al. (eds), Combinatorial Image Analysis, LNCS 4040, Springer, Berlin, Heidelberg, New York
26. Klette G, Pan M (2004) 3D topological thinning by identifying non-simple voxels. In: Proc. Int. Workshop Combinatorial Image Analysis, LNCS 3322
27. Klette G, Pan M (2005) Characterization of curve-like structures in 3D medical images. In: Proc. Image Vision Computing New Zealand
28. Klette R, Rosenfeld A (2004) Digital Geometry – Geometric Methods for Digital Picture Analysis. Morgan Kaufmann, San Francisco, CA
29. Klette R, Sun H-J (2001) Digital planar segment based polyhedrization for surface area estimation. In: Arcelli C, Cordella LP, Sanniti di Baja G (eds), Visual Form, Springer, Berlin
30. Kim CE (1983) IEEE Transactions on Pattern Analysis Machine Intelligence 5:231–234
31. Kong TY (2004) International Journal of Pattern Recognition Artificial Intelligence 9:813–844
32. Kong TY, Rosenfeld A (1989) Computer Vision Graphics Image Processing 48:357–393
33. Kovalevsky VA (1989) Computer Vision, Graphics, and Image Processing 46(2):141–161
34. Lachaud J-O, Montanvert A (2000) Graphical Models and Image Processing 62:129–164
35. Latecki LJ (1998) Discrete Representations of Spatial Objects in Computer Vision. Kluwer, Dordrecht
36. Li F, Klette R (2006) Calculation of the number of tunnels. IMA Preprint Series 2113
37. Lohmann G (1988) Volumetric Image Analysis. Wiley & Teubner, Chichester
38. Lorensen WE, Cline HE (1987) Computer Graphics 21:163–170
39. Ma C-M, Wan S-Y (2000) Computer Vision Image Understanding 80:364–378
40. Malgouyres R (1997) Theoretical Computer Science 186:1–41
41. Morgenthaler DG, Rosenfeld A (1981) Information Control 51:227–247
42. Mylopoulos JP, Pavlidis T (1971) Journal of the ACM 18:239–246
43. Nakamura A, Morita K, Imai K (2006) B-problem. CITR-TR-180. The University of Auckland, Auckland
44. Palagyi K, Kuba A (2003) Directional 3D thinning using 8 subiterations. In: Proc. Discrete Geometry Computational Imaging, LNCS 1568
45. Palagyi K, Kuba A (1998) Pattern Recognition Letters 19:613–627
46. Palagyi K, Sorantin E, Balogh E, Kuba A, Halmai C, Erdohelyi B, Hausegger K (2001) A sequential 3D thinning algorithm and its medical applications. In: Proc. Information Processing Medical Imaging, LNCS 2082
47. Peltier S, Alayrangues S, Fuchs L, Lachaud J (2005) Computation of homology groups and generators. In: Proc. DGCI, LNCS 3429
48. Rosenfeld A, Klette R (2001) Digital straightness. In: Electronic Notes in Theoretical Computer Science, Vol. 46
49. Saha PK, Chaudhuri BB (1996) Computer Vision Image Understanding 63:418–429
50. Siguera M, Latecki LJ, Gallier J (2005) Making 3D binary digital images well-composed. In: Proc. Vision Geometry, SPIE 5675
51. Srihari SN (1981) ACM Computing Surveys 13:399–424
52. White AT (1972) Pacific Journal of Mathematics 41:275–279

Chapter 5
Multimodality in Brain Imaging: Methodologic Aspects and Applications

Sónia I. Gonçalves

5.1 Introduction

The human brain is probably the most sophisticated result of evolution and its existence has allowed the human species to shape its environment in a definitive way. Though the structure and function of the brain are very complex, together they make the brain attain a remarkable degree of effectiveness.

The human brain controls the central nervous system (CNS), the peripheral nervous system (PNS) and it regulates virtually all human activity [1, 24]. Different types of activity are controlled by different elements of the central-peripheral nervous systems. Involuntary functions such as heart rate control, respiration or digestion are unconsciously controlled through a part of the peripheral nervous system which is the autonomic nervous system [1, 24] whereas complex mental activities such as thought, reason or abstraction are consciously controlled.

From an anatomical point of view, the brain can be divided into three parts: the forebrain, the midbrain and the hindbrain. The forebrain includes the cerebral cortex which is organized in lobes and it controls higher functions. The human cerebral cortex is 2–4 mm thick and it plays a central role in many complex brain functions including memory, attention, perceptual awareness, language and consciousness. From an evolutionary point of view, what distinguishes humans from other less developed species is the fact that the mass of the cerebral cortex, specifically that of the neocortex, which is involved in language and consciousness, increased much more in the course of time. This evolutionary difference gave humans unique mental capacities despite having a neuroarchitecture which is very similar to that of more primitive species.

The working of the human brain has since long been a source of many questions and hypothesis. For example, Aristotle thought that mental activity was located in

S.I. Gonçalves
VU University Medical Centre (Dpt. PMT), De Boelelaan 1117, 1081 HV, Amsterdam, The Netherlands
e-mail: s.goncalves-verheij@vumc.nl

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

the heart whereas ancient Greek scholars assumed correctly that the brain had a role in cooling the body but incorrectly presumed the brain to have radiating functions. The Alexandrian biologists Herophilos and Erasistratus were among the first to conclude that intelligence was "located" in the brain.

In our days, the study of the brain and its functions is the central role of modern Neuroscience. It combines disciplines such as Neurophysiology, which aims at studying the normal healthy brain as well as neurology or psychiatry which focus on pathology. In parallel, technical developments have allowed the use of new methodologies to quantitatively investigate the brain, both from a structural and functional point of view.

In particular, Neuroimaging has allowed the function of the living brain to be studied in detail without damaging it.

One of the first techniques to be used in Neuroimaging was Electroencephalography (EEG) [20] which consists of measuring the electric activity of the brain from electrodes placed on the scalp. Two of the main advantages of the EEG, when compared to other techniques is that it measures the electric activity of the brain directly with a high time resolution (on the level of a single millisecond). It has however several limitations: the high electric resistivity of the skull smears the EEG signal, degrading the spatial resolution, and deep sources of activity contribute negligibly to the EEG. A closely related technique to EEG is Magnetoencephalography (MEG) [21] which measures the magnetic activity produced by the brain with sensors that are placed above the scalp. Similar to EEG, MEG has a high temporal resolution. However, contrary to the EEG, magnetic fields are much less distorted by the skull and as a consequence, the spatial resolution of the MEG is better than that of the EEG. However, MEG is almost insensitive to activity which is radial to surface of the scalp whereas EEG can detect activity that is both tangential and radial to the surface of the scalp.

In order to localize the sources of brain activity, several advanced signal processing techniques have been applied to both EEG and MEG data. In formal terms, the problem is often formulated as finding the sources of the electric/magnetic activity which is measured on the surface of the scalp, i.e. finding a solution to the *Inverse Problem* [15, 27]. The primary technical difficulty is that the Inverse Problem does not a have a unique solution and therefore strategies to constrain the solution have to be found. One of the ways to constrain the solution is to specify a model to describe the source. Source models can be either overdetermined or underdetermined. An overdetermined model may consist of a few point-like dipolar sources whose locations and orientations are then estimated from the data. The underdetermined models may be used in cases where many different distributed areas are activated. In this situation, because several distributions are possible, the most likely is chosen under a given criteria [16, 22, 26].

Magnetic resonance imaging (MRI) is a technique which allows the study of brain structure. It uses the variation in signals produced by water protons in the body when the head is placed in a strong magnetic field [23] to produce highly detailed images of the brain anatomy. In the beginning of the 1990s, the understanding that blood flow changes induced by brain activity could be imaged using MRI

culminated in the development of functional MRI (fMRI). FMRI-BOLD (blood oxygenation level-dependent) is able to measure small changes in blood oxygenation level associated with brain activity (e.g. due to the activation of a certain brain involved in some task performed by the subject), which cause slight variations in the MR signal. By comparing the signal in MR images acquired during the period when the subject is performing some task (task period) to that acquired during the period when the subject is at rest (rest period), it is possible to determine which brain areas were involved in processing a given task. The high spatial resolution of fMRI, of the order of millimeters makes it the preferred technique to localize a given brain process. Contrary to EEG or MEG however, the temporal resolution of fMRI is much lower (of the order of the second) and it measures electric brain activity indirectly through the variations in blood oxygenation and blood volume.

Multimodal imaging is a concept which consists of combining existing brain imaging techniques in order to allow a better interpretation of the data and a better understanding of brain functions. The simultaneous measurement of MEG and EEG is an example of such multimodal approach where advantage is taken of the different sensitivities of EEG and MEG to radial and tangential sources as well as of the higher spatial resolution of MEG when compared to EEG alone. Another example multimodal imaging is the combination of (f)MRI with EEG or MEG. This combination of techniques took initially the form of fusion through constraint [14] where information derived from (f)MRI was used to constrain the solutions of the EEG/MEG inverse problem. Recently, the possibility of recording EEG and fMRI simultaneously [6, 19] has originated a new approach where the EEG is used to predict the variations in MR signal, i.e. fusion through prediction [14]. This approach relates the information derived directly from the electric brain activity to the hemodynamic phenomena which can be localized with a high spatial resolution. It has therefore the possibility to give new perspectives into neuronal sources.

In this paper, the concept of multimodality is illustrated with two applications. The first one is a method that combines EEG and MEG to estimate in-vivo the electric resistivities of brain and skull as way of optimizing the solution to the EEG/MEG inverse problem. The second application makes use of the co-registration of EEG and fMRI to study the spontaneous variations of the alpha rhythm [4, 7, 11, 17, 18, 25] during the resting state.

5.2 Materials and Methods

5.2.1 In-vivo Measurement of Brain and Skull Resistivities

5.2.1.1 Data Acquisition

Data from six normal young subjects were acquired using the Omega MEG/EEG system (CTF Systems Inc.) with 151 MEG channels and 64 EEG channels. Details can be found in [9, 10]. For the EIT, a sinusoidal current of 60 Hz and 10 μV rms.

Data were acquired at a rate of 1,250 Hz and epochs of 103 s were recorded for each injecting pair, each epoch consisting of 32 trials of 3.28 s which were recorded in sequence. For the SEF/SEP data, the median nerve was stimulated with an electric current having a frequency of 2 Hz and 0.2 ms duration. The current intensity depended on the subject's sensitivity and it varied between 2.5 and 12 mA. After stimulus onset, MEG and EEG were simultaneously recorded at a rate of 1,250 Hz. A total of 500 to 700 epochs of 0.44 s were recorded in order to improve the SNR of both EEG and MEG.

5.2.1.2 EIT Method

The application of the principles of EIT (Electric Impedance Tomography) in combination with EEG to estimate in-vivo the resistivities of brain and skull has been demonstrated in [8–10]. The principles are illustrated in Fig. 5.1.

A known current I is injected into the scalp and the resulting potential distribution V is measured. The head geometry is assumed to be known and it can be described using a spherical model (Fig. 5.1) consisting of three concentric spheres representing brain, skull and scalp respectively. Alternatively, a realistic model for the head can be used and in this case, the contours corresponding to the three head compartments are derived from the individual MR scans. In both cases, the models are considered to be homogeneous and isotropic. Since injected current, potential distribution and head model are known, the only unknowns that remain are the resistivities of brain (ρ_{brain}) and skull (ρ_{skull}) where it is assumed that $\rho_{scalp} = \rho_{brain}$.

In the case of the spherical head model, an analytic solution is available and it can be found in [8]. For the realistic head model only a numerical solution is available and it is found using the BEM (Boundary element method) according to what is described in [9] (Fig. 5.2). Briefly, each surface is discretized into a set of triangles (Fig. 5.2b) and a solution to a system of equations in matrix form is found.

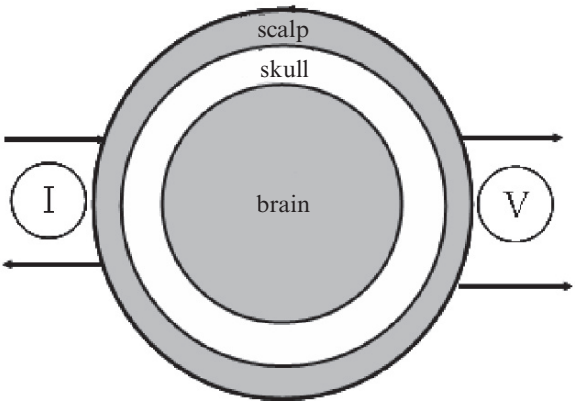


Fig. 5.1 Schematic representation of the principles of EIT when applied to estimate the resistivities of brain and skull

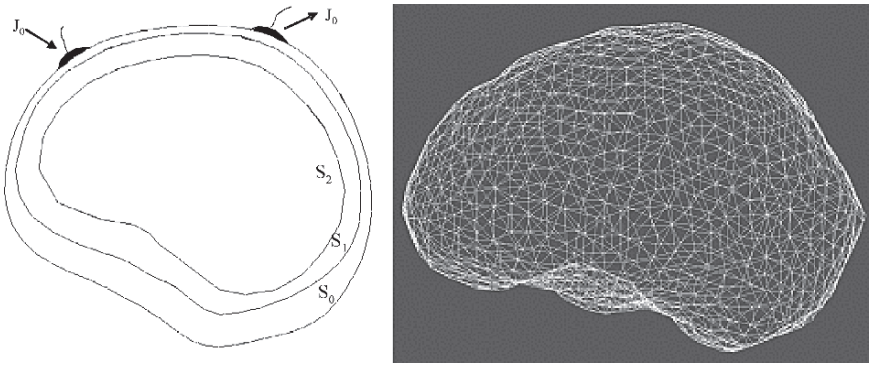


Fig. 5.2 Realistic model for the head. (**a**) Schematic representation of the head compartments with boundaries S0 (scalp), S1 (skull) and S2 (brain) and corresponding electric resistivities The "+" and "−" subscripts refers to inner and outer resistivities respectively. (**b**) Example of discretized surface to be used with BEM

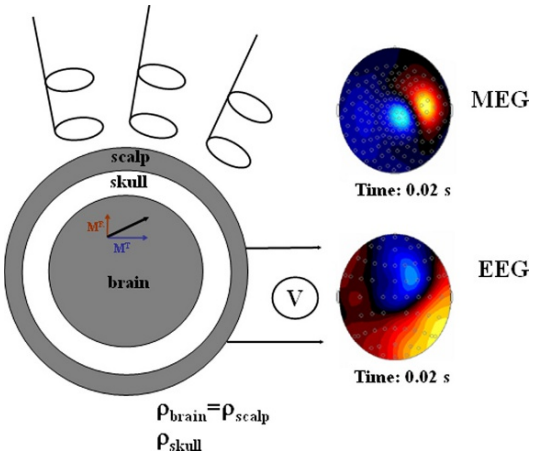


Fig. 5.3 Schematic representation of the combined analysis of SEF/SEP data to estimate the electric resistivities of brain and skull

5.2.1.3 Combined Analysis of Somatosensory Evoked Fields (SEF) and Somatosensory Evoked Potentials (SEP)

In this approach, the combined analysis of SEF/SEP data, for which it is well known that both for MEG and EEG a single dipole model is adequate, is used in order to estimate the electric resistivities of brain and skull. The formal description of the method can be found in [10] and it is schematically represented in Fig. 5.3. Briefly, the method can be resumed to the following steps:

(a) The sources that best explain the MEG data at time instants around the N20 response are determined. The positions and the tangential components (M^T) of the sources are then known.
(b) The EEG data is used to fit the remaining unknowns: the radial components (M^R) of the dipoles, ρ_{brain}, and $\rho_{skull}/\rho_{brain}$.

5.2.2 The Study of the Spontaneous Variations of the Alpha Rhythm Using Co-registered EEG-fMRI

5.2.2.1 Data Acquisition

EEG and fMRI data was acquired from eight healthy subjects while the subjects lied still inside the scanner. The subjects were instructed to keep their eyes closed, remain rested but without falling asleep. The goal was to optimize the conditions to record the spontaneous variations of the alpha rhythm. The EEG was acquired using an MR-compatible EEG amplifier (SD-MRI, Micromed, Treviso, Italy) at an acquisition rate of 1,024 Hz. Functional images were acquired on a 1.5 T MR scanner (Magnetom Sonata, Siemens, Erlangen, Germany) using a T2*-weighed EPI sequence, with a TR equal to 3 s and 24 transversal slices covering the complete occipital lobe and most of the parietal and frontal lobes. A total of 400 volumes, i.e. 20 min of data, was recorded for each subject. In addition, a high resolution MPRAGE sequence was also acquired in order to provide the anatomical reference to the functional scans. Further details regarding data acquisition can be found in [11].

5.2.2.2 Data Processing

The EEG data was first corrected for gradient and pulse artefacts caused by the MR sequence according to the method described in [12]. Subsequently, the data was processed in order to derive the average temporal power variation of the alpha rhythm corresponding to several bipolar derivations of interest located in the central and occipital regions. The functional MR data was motion corrected and spatially smoothed with a spatial filter of 4 mm radius.

The BOLD signal in each voxel was correlated to the average alpha power time series using the General Linear Model (GLM) (Cox et al. 1995; Kherif et al. 2002; Worsley et al. 2002). Furthermore, the average alpha power time series was convolved with a canonical hemodynamic response function (HRF) [5]. The correction for multiple comparisons was made by means of controlling the false-discovery rate (FDR) (Benjamin and Hochberg, 1995). Further details can be found in [11].

5.3 Results

5.3.1 In-Vivo Measurement of Brain and Skull Resistivities

The results obtained for the spherical model using both the EIT method and the combined analysis of SEF/SEP data are presented in Table 5.1 and Fig. 5.4. Results show that there is an agreement between both methods, especially regarding the values of $\rho_{skull}/\rho_{brain}$. Only for subjects one and six the agreement is not so good. But for this subjects the absolute values of ρ_{skull} and ρ_{brain} differ also more amongst the two methods. It is also observed that most of the inter-subject variability in the values of $\rho_{skull}/\rho_{brain}$ results from the variability associated with ρ_{skull}.

The results obtained for the realistic head model are presented in Table 5.2 for the EIT method.

Table 5.1 Results obtained for the spherical model using both the EIT method and the combined analysis of SEF/SEP data. The entry identified with SD refers to the standard deviation associated with the computed average

Subject	ρ_{brain} ($\Omega\ cm$)	ρ_{skull} ($\Omega\ cm$)	$\rho_{skull}/\rho_{brain}$	ρ_{brain} ($\Omega\ cm$)	ρ_{skull} ($\Omega\ cm$)	$\rho_{skull}/\rho_{brain}$
1	440	13,300	30	175	7,441	43
2	245	30,800	127	–	–	–
3	280	26,900	96	280	24,000	86
4	295	20,000	68	250	16,300	65
5	245	16,100	66	250	18,600	74
6	330	15,000	45	215	14,600	68
Average	305	20,355	–	234	16,185	–
SD (%)	24	35	–	17	37	–

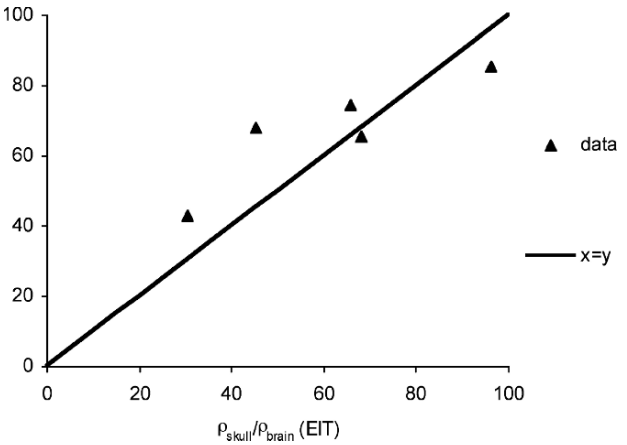


Fig. 5.4 Plot of $\rho_{skull}/\rho_{brain}$ computed using the EIT method against the same parameter obtained with the combined analysis of SEF/SEP data. The identity line is also plotted for comparison

Table 5.2 Results obtained for the spherical and realistic models for the EIT method

Subject	Spherical			Realistic		
	ρ_{brain} ($\Omega\ cm$)	ρ_{skull} ($\Omega\ cm$)	$\rho_{skull}/\rho_{brain}$	ρ_{brain} ($\Omega\ cm$)	ρ_{skull} ($\Omega\ cm$)	$\rho_{skull}/\rho_{brain}$
1	435	13,272	30	333	11,928	36
2	243	30,799	127	292	12,344	42
3	279	26,898	96	292	14,217	49
4	295	20,057	68	311	13,598	44
5	245	16,072	66	234	13,174	56
6	332	15,031	45	346	8,119	23

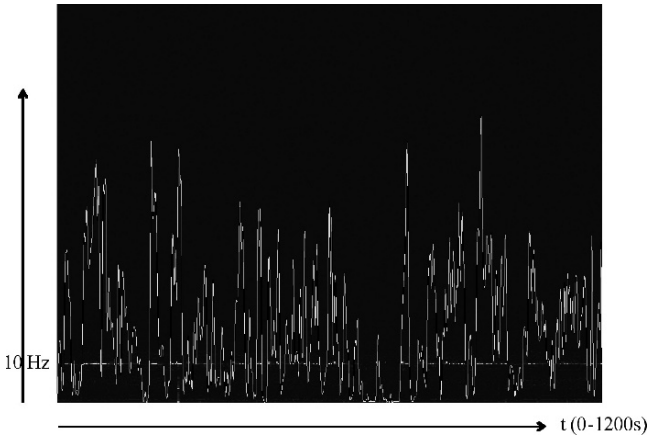


Fig. 5.5 Typical example of a spectrogram obtained during rest. The yellow profile shows the temporal variations of the alpha rhythm

From Table 5.2 it can be seen that the inter-subject variability of $\rho_{skull}/\rho_{brain}$ decreases considerably when using realistic models, being that this decrease is mainly do the lower inter-subject variability associated with ρ_{skull}.

5.3.2 The Study of the Spontaneous Variations of the Alpha Rhythm Using Co-registered EEG-fMRI

A typical example of a spectrogram obtained during rest is shown in Fig. 5.5. The superimposed yellow profile shows the temporal variations of the alpha rhythm power. In the figure, brighter colors correspond to larger power values whereas darker colors correspond to lower power values. In Fig. 5.6, samples of EEG signals recorded with eyes open and eyes closed are shown. These signals were corrected for gradient artefacts and the absence/presence of the alpha rhythm is well visible. Also visible is the pulse artefact which is identified with a circle. As mentioned

Fig. 5.6 Samples of EEG signals obtained in eyes open and eyes closed conditions. The circle shows the pulse artefact

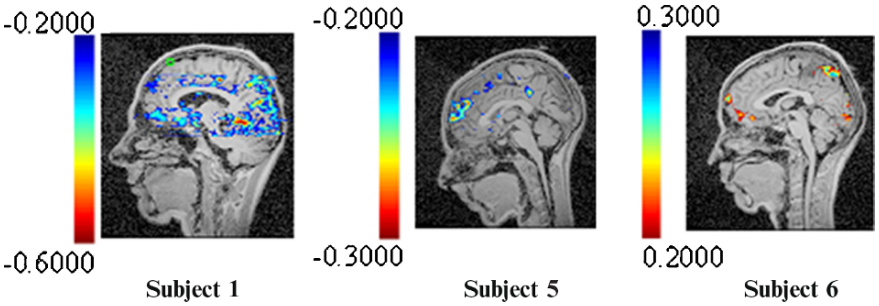


Fig. 5.7 Individual correlation maps corresponding to three representative subjects

before, this type of artefact was corrected using the method presented in [12]. In Fig. 5.7, typical examples of the correlation distribution between the spontaneous variations of the alpha rhythm is shown for three different subjects. It can be seen that in general the alpha rhythm is negatively correlated to BOLD is cortical regions, although for one of the subjects the correlations are positive. Furthermore, as shown in more detail in [11] positive correlations between alpha and BOLD are also found in the thalamus, thus confirming previous studies [7,17,18,25]. Contrary to previous studies however, remarkable inter-subject variability was observed.

5.4 Discussion and Conclusions

The results obtained for the in vivo computation of brain and skull resistivities show that the variability of the resistivity among subjects, using spherical models, was confirmed by two independent methods. Furthermore, it was shown that the realistic models corrects for the true skull thickness for each subject. This eliminates the compensation effect resulting from using the same skull thickness for all subjects, when using the spherical model. This corrects then for an over- or underestimation of the resistivities as explained in detail in [8]. However, even with realistic models, the average of $\rho_{skull}/\rho_{brain}$ amongst subjects is approximately half the value that is commonly accepted, i.e. 80. Furthermore, even with realistic models, some inter-subject variability can still be observed which might eventually explained by natural causes.

The results regarding the spontaneous variations of the alpha rhythm suggest that the resting state is not comparable amongst subjects and sometimes, not even within one subject. In addition, a recent study [4] about the HRF of the alpha rhythm shows that although the its shape appears to be very constant over subjects, the delay in peak response tends to vary systematically over different brain regions.

As the resting state plays an important role in fMRI analysis where the paradigms are of the type "rest-task", the abovementioned inter-subject variability should be considered when questioning how comparable are fMRI results from different subjects.

The advantage of using multimodality to study the brain function has been demonstrated using two examples of significant importance in the field of brain imaging. Multimodality implies a deep insight into each of the techniques being used in order to overcome problems of technical nature. This usually requires a multidisciplinary collaboration. Furthermore, multimodality also poses new methodological questions that require a deep and fundamental understanding of each technique. Nevertheless, the future research of the brain lies in the use of multimodality as this is the best way to gather complementary information about the brain processes.

References

1. Afifi AK, Bergman RA (2005) Functional neuroanatomy: text and atlas. McGraw-Hill, New York.
2. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. JR Statist Soc B 57(1): 289–300.
3. Cox RW, Jesmanowicz A, Hyde JS (1995) Real-Time functional magnetic resonance imaging, MRM 33: 230–236.
4. De Munck JC, Gonçalves SI, Huijboom L, Kuijer JPA, Pouwels PJW, Heethaar RM, Lopes da Silva FH (2007) The hemodynamic response of the alpha rhythm: An EEG/fMRI study. NeuroImage 35: 1142–1151.
5. Glover GH (1999) Deconvolution of impulse response in event-related BOLD fMRI. Neuro Image 9: 416–429.

6. Goldman RI, Stern JM, Engel Jr. J, Cohen M (2000) Acquiring simultaneous EEG and functional MRI. Clin Neurophysiol 111: 1974–1980.
7. Goldman RI, Stern JM, Engel Jr. J, Cohen M (2002) Simultaneous EEG and fMRI of the alpha rhythm. NeuroReport 13(18): 2487–2492.
8. Gonçalves S, De Munck JC, Heethaar RM, Lopes da Silva FH, van Dijk BW (2000) The application of electrical impedance tomography to reduce systematic errors in the EEG inverse problem – a simulation study. Physiol Meas 21: 379–393.
9. Gonçalves SI, De Munck JC, Verbunt JPA, Bijma F, Heethaar RM, Lopes da Silva FH (2003a) In Vivo measurement of the brain and skull resistivity using an EIT-Based Method and realistic models for the head. IEEE Trans Biomed Eng 50(6): 754–767.
10. Gonçalves SI, De Munck JC, Verbunt JPA, Heethaar RM, Lopes da Silva FH (2003b) In vivo measurement of the brain and skull resistivities using an EIT-based method and the combined analysis of SEF/SEP data. IEEE Trans Biomed Eng 50(9): 1124–1128.
11. Gonçalves SI, De Munck JC, Pouwels PJW, Schoonhoven R, Kuijer JPA, Maurits N, Hoogduin JM, Van Someren EJW, Heethaar RM, Lopes da Silva FH (2006) Correlating the alpha rhythm to BOLD using simultaneous EEG/fMRI: inter-subject variability. NeuroImage 30: 203–213.
12. Gonçalves SI, De Munck JC, Pouwels PJW, Kuijer JPA, Heethaar RM (2007) Artifact removal in co-registered EEG-fMRI by selective average subtraction. Clin Neurophysiol 118: 2437–2450.
13. Kherif F, Poline JB, Flandin G, Benali H, Simon O, Dehaene S, Worsley KJ (2002) Multivariate model specification for fMRI data. NeuroImage 16: 1068–1083.
14. Kilner JM, Mattout J, Henson R, Friston FJ (2005) Hemodynamic correlates of EEG: A heuristic. NeuroImage 28: 280–286.
15. Koles ZJ (1998) Trends in EEG source localization. Electroenceph Clin Neurophysiol 106: 127–137.
16. Lantz G, Michel CM, Pascual-Marqui RD, Spinnelli L, Seeck M, Seri S, Landis T, Rosen I (1997) Extracranial localization of intracranial interictal epileptiform activity using LORETA (low resolution electromagnetic tomography). Electroencephalogr Clin Neurophysiol 102(5): 414–422.
17. Laufs H, Krakow K, Sterzer P, Eger E, Beyerle A, Salek-Haddadi A, Kleinschmidt A (2003a) Electroencephalographic signatures of attentional and cognitive default modes in spontaneous brain fluctuations at rest. PNAS 100(19): 11053–11058.
18. Laufs H, Kleinschmidt A, Beyerle A, Eger E, Salek-Haddadi A, Preibisch C, Krakow K (2003b) EEG-correlated fMRI of human alpha activity. NeuroImage 19: 1463–1476.
19. Lemieux L, Allen PJ, Franconi F, Symms MR, Fish DR (1997) Recording of EEG during fMRI experiments: patient safety. MRM 38: 943–952.
20. Lopes da Silva F, Niedermeyer E (1999) Electroencephalography: basic principles, clinical applications, and related fields. Williams & Wilkins, Baltimore, MD.
21. Lounasmaa OV, Hämäläinen M, Hari R, Salmelin R (1996) Information processing in the human brain: magnetoencaphalographic approach. Proc Natl Acad Sci USA 93(17): 8809–8815.
22. Mattout J, Philips C, Penny WD, Rugg MD, Friston KJ (2006) MEG source localization under multiple constraints: an extended Bayesian framework. NeuroImage 30(3): 753–767.
23. Mark Haacke E, Brown RW, Thompson MR, Venkatesan R (1999) Magnetic resonance imaging: physical principles and sequence design. Wiley, New York.
24. Mendonza J, Foundas AL (2008) Clinical neuroanatomy: a neurobehavioral approach. Springer Science + Business Media, New York.
25. Moosmann M, Ritter P, Krastel I, Brink A, Thees S, Blankenburg F, Taskin B, Obrig H, Villringer A (2003) Correlates of alpha rhythm in functional magnetic resonance imaging and near infrared spectroscopy. NeuroImage 20: 145–158.
26. Trujillo-Barreto NJ, Aubert-Vázquez E, Penny WD (2008) Bayesian M/EEG source reconstruction with spatio-temporal priors. NeuroImage 39(1): 318–335.
27. Sarvas J (1987) Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem. Phys Med Biol 32: 11–22.

Chapter 6
Research Steps Towards Human Sequence Evaluation

Jordi Gonzàlez, F. Xavier Roca, and Juan J. Villanueva

Abstract *Human Sequence Evaluation* (HSE) concentrates on how to extract descriptions of human behaviour from videos in a restricted discourse domain, such as (i) pedestrians crossing inner-city roads where pedestrians appear approaching or waiting at stops of busses or trams, and (ii) humans in indoor worlds like an airport hall, a train station, or a lobby. These discourse domains allow exploring a coherent evaluation of human movements and facial expressions across a wide variation of scale. This general approach lends itself to various cognitive surveillance scenarios at varying degrees of resolution: from wide-field-of-view multiple-agent scenes, through to more specific inferences of emotional state that could be elicited from high resolution imagery of faces. The true challenge of HSE will consist of the development of a system facility which starts with basic knowledge about pedestrian behaviour in the chosen discourse domain, but could cluster evaluation results into semantically meaningful subsets of behaviours. The envisaged system will comprise an internal logic-based representation which enables it to comment each individual subset, giving natural language explanations of why the system has created the subset in question.

6.1 Introduction

Hermeneutics, according to Wilhelm Dilthey, is the art of interpretation of hidden meanings. The name comes from HERMES, the God known as the messenger of the intentions of the Gods to the human beings. In particular, interpretation in cultural sciences requires to *know* its object, a human being, from the inside. That means,

J. Gonzàlez
Institut de Robòtica i Informàtica Industrial (UPC-CSIC), Edifici U, Parc Tecnològic de Barcelona, Barcelona 08028, Catalonia, Spain

F.X. Roca and J.J. Villanueva
Computer Vision Center, Edifici O, Campus UAB, Bellaterra 08193, Catalonia, Spain

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

we can infer the intentions of a person because we also are persons. Towards this end, we will address basic methods for the extraction, description and animation of human motion in the same scenario (indoor or outdoor), and new methods for the interpretation of dynamic scenes.

The design and implementation of such a cognitive system still constitutes a challenge, even if the discourse domain will be drastically constrained within which it is expected to operate. An algorithmic system with analogous capabilities can be considered an instantiation of a 'cognitive system'. In particular, the term Human Sequence Evaluation (HSE) denotes the design, implementation and evaluation of such a cognitive system [11]. In general terms, we proposed to develop towards weakly embodied cognition within a system for understanding an environment containing autonomous agents. By understanding, we mean that the system must move beyond merely describing the scene: in addition it must be able to reason about the scene and give suitable explanations for various events and behaviours.

Thus, the generation of semantic descriptions conveys the meaning of motion, i.e. *where, when, what, how* and also *why* the motion is being detected. As a result, this high-level understanding provide a richer, broader and even more challenging domains of research, which will encompass not only research in Computer Vision, but also in Pattern Recognition, Artificial Intelligence and Computer Animation, to cite few.

At present, few video surveillance systems exploits all these aspects of cognition: we restrict cognition to assure HSE, that means, on the one hand, to develop transformation processes to perform human motion understanding and, on the other hand, to convey inferred interpretations to human operators by means of natural language texts or synthesized agents in virtual environments.

This paper presents how HSE considers the interpretation of human motion as a transformation process between raw video signals and high-level, qualitative descriptions. At least, this process will involve (i) the extraction of relevant visual information from a video sequence, (ii) the representation of that information in a suitable form, and (iii) the interpretation of visual information for the purpose of recognition and learning about human behaviour.

6.2 State of the Art

During the past three decades, important efforts in Computer Vision research have been focused on developing theories, methods and systems applied to video sequences [22]. Broadly speaking, research is focused on describing *where and when* motion is being detected by camera sensors. For this purpose, the goal is set to describe motion using quantitative values, such as the spatial position of a given agent over time, for example.

Suitable discourse domains are, e.g., well-frequented streets, pedestrian-crossings, bus-stops, reception desks of public buildings, railway platforms. This demand in surveillance systems is due to the huge amount of video which should be selected, watched, and analyzed by a small number of operators in real time. Current

textual descriptions generated automatically from surveillance sequences helps to detect abnormal and dangerous situations on-line. As a long-term result of HSE, surveillance systems will not only recognize and describe, but also *predict* abnormal or dangerous behaviours on-line, instead of merely record video sequences.

The basis of current research in any of the aforementioned domains is the detection of agents within the scene. Two different approaches are found in the literature, namely, *background modeling/substraction* and *motion detection*. The former necessitates implementing a suitable background model of the scene to determine foreground regions. Most referred publications use a background modeling-based approach [12, 18, 35]. On the other hand, motion detection computes motion information from consecutive frames. Consequently, an action can be described in terms of a proper motion characterization [19, 21, 30].

Additionally, *tracking* procedures are usually incorporated in order to reduce segmentation errors [32]. In recent years, new tracking techniques are defined based on a hypothesis/validation principle [5]. Thus, the tracking process is modeled using a probabilistic scheme, which is based on the Bayes' rule [3, 14, 28, 34].

Tracking techniques should embed knowledge about the human agent, such as its observed motion, appearance, or shape. This knowledge can be based on *image features* or *predefined body models*. On the one hand, the spatial information of the agent state in video surveillance systems is often represented using simple image features, such as points, lines, or regions.

Most popular representations are blobs [17] or blob attributes, such as the centroid, median or bounding box. On the other hand, model-driven approaches incorporate known physical constraints of limbs and extremities of the body to help both localisation and tracking. By providing a synthetic body model, anatomical information and kinematic constraints are incorporated into the action model, thus allowing tracking of limbs, synthesis of motion, and performance analysis. Most referred models are those based on stick figures [6, 7, 15], 2-D contours [38, 40] and volumetric models [2, 10, 27].

Once the body model is properly tracked over time, it is possible to recognize predefined motion patterns and to produce high-level descriptions. In fact, the basis of motion understanding is *action recognition*. In order to deal with the inherent temporal and spatial variability of human performances, suitable analytical methods have been used in the literature for matching time-varying data. Most referred algorithms are Dynamic Time Warping (DTW), Hidden Markov Models (HMM) and Neural Networks (NN) [9, 39].

Subsequently, human motion information is then combined with the known information about the environment in order to derive complex semantic descriptions [11]. From a semantic perspective, conceptual predicates extracted from video sequences are classified according to different criteria, such as *specialization relationship* [15], *semantic nature* [29] or *temporal ordering* [13]. Likewise, suitable behaviour models explicitly represent and combine the specialization, semantic and temporal relationships of their constituent semantic predicates [24]. For this purpose, semantic primitives involved in a particular behaviour are organized into hierarchical structures, such as networks [31] or trees [16, 37] which allow motion understanding.

On the one hand, semantic interpretation is still mostly restricted to express the *relationships* of an agent with respect to its environment. However, the *internal state* of the agent has traditionally received little (or none) attention in human motion understanding. But human agents have inner states (based on emotions, personality, feelings, goals and beliefs) which may determine and modify the execution of their movement. These inner states are hard to be derived from a single picture. Instead, we need image sequences to evaluate emotions, like *sad, happy* or *angry*, in a temporal context.

Emotion descriptions will require high-detailed images which will be obtained by means of active cameras. In fact, camera's zoom are controlled to supply imagery at the appropriate resolution for motion analysis of the human face, thus facilitating emotion analysis [4, 41]. Current state-of-the-art is mainly concerned with posed facial expression recognition. In the proposed scenario, we would encounter spontaneous expressions that are considerably more difficult to handle. Only few publications can be found on spontaneous facial expression recognition and are mostly limited to very specific facial motions such as eye blinking.

On the other hand, semantic interpretation also leads to *uncertainty*, due to the vagueness of the semantic concepts utilized, and the incompleteness, errors and noise in the agent state's parameters [20]. In order to cope with the uncertainty aspects, integration and fusion methods can be learnt using a probabilistic framework: PCA, Mixtures of Gaussians (MoG) [8,23] and Belief Networks (BN) [13,29] provide examples. Alternatively, Fuzzy Metric Temporal Logic (FMTL) can also cope with the temporal and uncertainty aspects of integration in a goal-oriented manner [33].

6.3 Approach to Research

The main objective of HSE is to develop a cognitive artificial system based in a framework model which allows both recognition and description of a particular set of human behaviours arising from real-world events. Specifically, we propose to model the knowledge about the environment in order to make or suggest interpretations from motion events, and to communicate with people using natural language texts, audio or synthetic films. These events will be detected in image data-streams obtained from arrays of multiple active cameras (including zoom, pan and tilt).

HSE thus aims to design a Cognitive Vision System for human motion and behaviour understanding, followed by communication of the system results to end-users, based on two main goals. We assume that three different types of descriptions can be obtained, which depend on the resolution of the acquisition process: facial expressions, body postures and agent trajectories, where each topic demands its own specific requirements and computational models for a proper representation.

So the first goal is to determine which interpretations are feasible to be derived in each category of human motion, see Fig. 6.1. Consequently, for each category, suitable human-expressive representations of motion will be developed and tested.

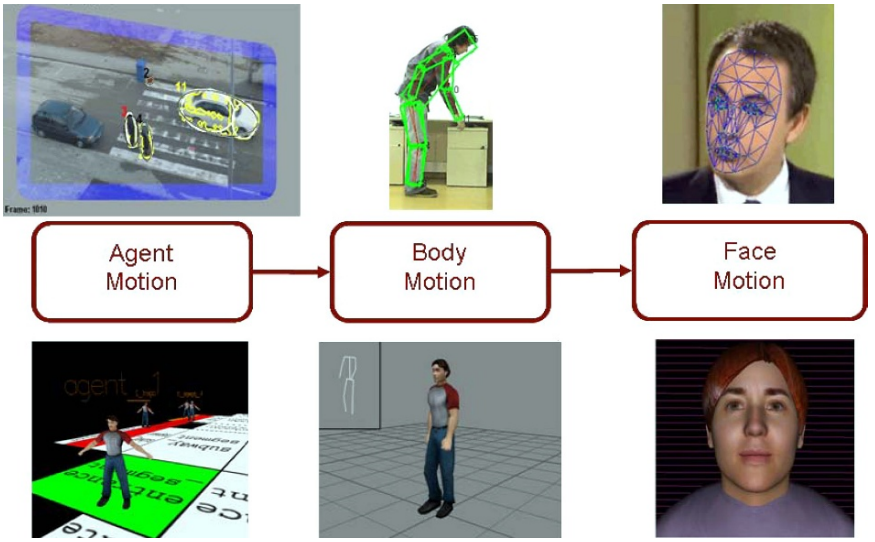


Fig. 6.1 Human-expressive representations of motion

In particular, HSE will interpret and combine the knowledge inferred from three different categories of human motion, namely the motion of agent, body and face, in the same discourse domain.

The distinction between these motion categories is due to the fact that knowledge of different nature is required to interpret agent trajectories, body poses and facial expressions, since these types of interpretations strongly depend on the details of motion which can be inferred from active video cameras. The strategy is to obtain the available information at a particular level (i.e., agent), thereby providing this incomplete knowledge to higher levels (i.e., body and face) which can update their representations as more information becomes available, and which can feedback the new information to the lower level.

The second objective of HSE is set to establish how these three types of interpretations can be linked together in order to coherently evaluate the human motion as a whole in image sequences. Such evaluation will require, at the very least, to *acquire* human motion from video cameras, to *represent* the recorded human motion using computational models, to *understand* the developments observed within a scene using high-level descriptions, and to *communicate* the inferred interpretations to a human operator by means of natural language texts or synthesized virtual agents as a visual language.

Thus, the main procedure of HSE will be the combination of:

- Detection and tracking of agents while they are still some distance away from a particular location (for example a bus station, a pedestrian crossing, or a passenger in an airport, or a guest in a lobby).

- When these agents come closer to the camera, or when the active camera zooms in on these agents, their body posture will be evaluated to check for compatibility with behaviour hypotheses generated so far.
- If they are even closer and their face can be resolved sufficiently well, facial emotions will be checked in order to see whether these again are compatible with what one expects from their movements and posture in the observational and locational context which has been accumulated so far by the system.

Naturally, the interest is greater to integrate the three different components of human motion for someone approaching than someone leaving the camera. In addition, the most complex task (emotion evaluation) will come last, when the most is known already about the person in question from the preceding observations. Moreover, emotion recognition will become more specific because it can be embedded into the context of the preceding observations and it can exploit the rigid and non-rigid motion of the face.

A suitable discourse domain comprises two types of scenarios: (i) open worlds (such as well-frequented streets, pedestrian-crossings and bus-stops), and (ii) indoor worlds (airports, train stations or lobbies, for example). Multiple active cameras will record people to infer what the humans intend to do. The main objective is the characterisation of humans to study the behaviour of people in these domains. It will be interesting whether the abilities to detect, track and characterize pedestrians would already be sufficiently advanced to reliably detect regional differences within the EU.

By implementing the aforementioned tasks, HSE will fulfill two main objectives, see Fig. 6.2: on the one hand, the goal will be *description*, or the generation of conceptual descriptions based on acquired and analysed motion patterns. On the

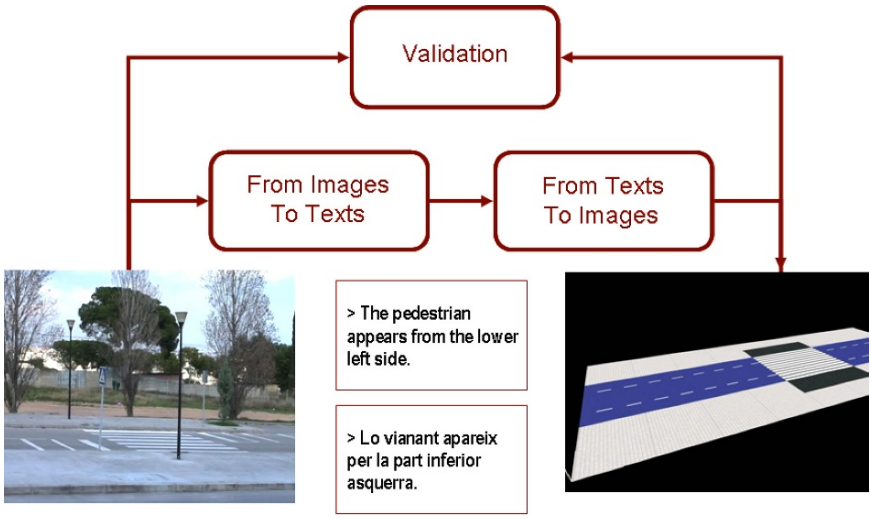


Fig. 6.2 Evaluation of human behaviours in image sequences

other hand, the aim will be communication using *visualization*, or the generation of synthetic motion patterns based on textual descriptions.

Firstly, natural language text generation will be accommodated within HSE based on the following considerations:

- Semantic descriptions will enable researchers to check details of the conceptual knowledge base.
- Semantic descriptions will allow communication with end-users of HSE in a most natural manner.
- Semantic descriptions will support conceptual abstraction, thereby facilitating the communication of short messages or essential details, possibly in response to inquiries communicated by a microphone near the recording camera or by an UMTS mobile phone, for example for blind people.

Descriptive texts will be applied to outdoor or indoor scenes from different parts of the EU. The inclusion of videos from different parts of Europe will also constitute a means to prevent over adaptation of HSE to a small set of learning videos. In addition, once a system-internal conceptual representation has been built, it will be possible to enlarge this for natural language text generation in the languages of all groups cooperating within HSE. Also, we will test whether the same video recordings are interpreted in different manners in different parts of Europe (or similar situations just happen in a different manner, for example people nicely queuing up at a bus station in one country and habitually cluster around the bus doors in another). Thus, on the one hand, HSE will achieve automatic translation of visual information and, on the other hand, it will be able to investigate how and why human motion may produce different descriptions, due to the cultural characteristics of the areas where a given language is spoken.

Secondly, animation will be accommodated within HSE based on the following considerations:

- Analysis-by-synthesis at the three stages of human behaviour, i.e. motion of people, their posture analysis, and their face characterisation.
- Animated computer graphics as a visual language to quickly communicate essential aspects to involved people like bus-drivers, policemen for helping people at pedestrian crossings, waiters in a lobby, etc.
- Animated computer graphics, again at three motion categories, for checking the conceptual knowledge base underlying the entire approach. Since this knowledge base is expected to grow or need adaptation throughout the project, animated computer graphics will provide the means to quickly check larger parts.

Using both natural language text generation and animation, quantitative measures and qualitative descriptions will be developed to analyze the robustness and the efficiency of the proposed cognitive system. In fact, the performance of the system will be studied by considering the following strategy [1,25]: let the system generate a synthetic image sequence using the textual descriptions obtained from a previously recorded image sequence. Both synthetic and original sequences can be compared to evaluate the suitability and correctness of the knowledge being considered so far.

Additionally, it will be possible to assess the results of the system by controlling the inference processes which are applied. The objectives will be:

- To trace the computational process which generates the result
- To determine the internal information requested by the system and
- To assess the selection of a particular interpretation

As a result, it will become easier to debug the system. Therefore, the designer can decide to incorporate extra knowledge (by means of models, restrictions, and default options) for improving the performance of the system in terms of reliability. Also, there will be an increase of confidence of end-users in the results reported by the system: evaluation, in the sense of explanation, will ease the understanding of the results by non-expert users.

6.4 Innovation Brought by HSE Research

As an innovation, HSE proposes to develop an unified framework for human motion analysis which will be applied to confront both animation and description. Our basis is that procedures for synthetic video generation should rely on knowledge very similar to the knowledge required for textual description.

Image-sequence evaluation will be driven to incorporate assessment strategies to guide and validate the system results by:

- Presenting the results of cognition using natural language texts or virtual animations and
- Arguing about inferred interpretations in order to assist and validate the system processes

Using this know-how, we will be able to look for characterizing the behaviour of pedestrians approaching to a traffic-light-controlled pedestrian-crossing of a well-frequented inner-city street, for example. In this particular domain, a pedestrian-crossing could switch to green without grossly interfering with vehicle traffic by preparing the transition phase (green-yellow-red for vehicles) while vehicles are still some distance away. Also, switching back to green earlier, even saving gas, thus helping the environment, compared with stopping a cavalcade of vehicles in full drive after having had the pedestrians waiting for several minutes. A similar idea could survey the environment around bus-stops with an associated gain in efficiency and comfort for all involved agents. Furthermore, provided one can extend this characterization of pedestrians reliably enough, it might become possible to design special help for handicapped people.

The basic procedure of HSE will be the combination of detection and tracking of agents while they are still some distance away from a particular location. Detecting and tracking people in crowded scenes is a challenging problem as people differ in their appearance caused by various types and styles of clothing and occluding accessories, undergo a large range of movements and moreover occlude each other. Previous approaches have either used appearance-based models or local features to detect people while a majority of trackers is still based on interactive initialization.

In HSE, cooperating pan-tilt-zoom sensors will also enhance this process of cognition via controlled responses to uncertain or ambiguous interpretations. Therefore, the challenge will be to provide sensor data for each of the modules by coupling the modules together in a sensor perception/action cycle [26, 36]. These cooperating pan-tilt-zoom sensors involved in acquisition will also serve the purpose of providing sensor data for each of the modules, but more importantly couple the other workpackages together in a sensor perception/action cycle. The use of zoom will provide an unification for interpretations of different resolution imagery, and will bestow the ability to switch the sensing process between different streams in a controlled fashion.

6.5 Conclusions

Multiple issues will be contemplated to perform HSE, such as detection and localization; tracking; classification; prediction; concept formation and visualization; communication and expression, etc. And this is reflected in the literature: a huge number of papers confront some of the levels, but rarely all of them. Summarizing, agent motion will allow HSE to infer behaviour descriptions. The term behaviour will refer to one or several actions which acquire a meaning in a particular context.

Body motion will allow HSE to describe action descriptions. We define an action as a motion pattern which represents the style of variation of a body posture during a predefined interval of time. Therefore, body motion will be used to recognize style parameters (such as age, gender, handicapped, identification, etc.).

Lastly, face motion will lead to emotion descriptions. The emotional characteristics of facial expressions will allow HSE to confront personality modeling, which would enable us to carry out multiple studies and researches on advanced human-computer interfaces.

So these issues will require, additionally, assessing how, and by which means, the knowledge of context and a plausible hypothesis about he internal state of the agent may influence and support the interpretation processes.

Acknowledgements This work is supported by EC grants IST-027110 for the HERMES project and IST-045547 for the VIDI-video project, and by the Spanish MEC under projects TIN2006-14606, DPI-2004-5414 and CONSOLIDER-INGENIO 2010 (CSD2007-00018). Jordi Gonzàlez also acknowledges the support of a Juan de la Cierva Postdoctoral fellowship from the Spanish MEC.

References

1. M. Arens, H.-H. Nagel, "*Behavioural knowledge representation for the understanding and creation of video sequences*", in: Proceedings of the 26th German Conference on Artificial Intelligence (KI-2003), Hamburg, Germany, LNAI 2821, Springer (2003) pp. 149–163.

2. J. Ben-Aire, Z. Wang, P. Pandit, S. Rajaram, "*Human activity recognition using multidimensional indexing*", IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002) 1091–1104.
3. D. Bullock, J. Zelek, "*Real-time tracking for visual interface applications in cluttered and occluding situations*", Image and Vision Computing 22 (12) (2004) 1083–1091.
4. I. Cohen, N. Sebe, A. Garg, L. Chen, T.S. Huang, "*Facial expression recognition from video sequences: temporal and static modeling*", Computer Vision and Image Understanding 91 (1–2) (2003) 160–187.
5. D. Comaniciu, V. Ramesh, P. Meer, "*Kernel-based object tracking*", IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–577.
6. J. Deutscher, I. Reid, "*Articulated body motion capture by stochastic search*", International Journal of Computer Vision 61 (2) (2005) 185–205.
7. S. Dockstader, M. Berg, A. Tekalp, "*Stochastic kinematic modeling and feature extraction for gait analysis*", IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (8) (2003) 962–976.
8. A. Fod, M. Mataric, O. Jenkins, "*Automated derivation of primitives for movement classification*", Autonomous Robots 12 (1) (2002) 39–54.
9. A. Galata, N. Johnson, D. Hogg, "*Learning variable-length markov models of behaviour*", Computer Vision and Image Understanding 81 (3) (2001) 398–413.
10. D. Gavrila, L. Davis, "*3D model-based tracking of humans in action: a multiview approach*", in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'96) (1996) pp. 73–80.
11. J. Gonzàlez, "*Human Sequence Evaluation: The Key-Frame Approach*", PhD Thesis, Universitat Autònoma de Barcelona, October 2004.
12. I. Haritaoglu, D. Harwood, L.S. Davis, "W^4: *real-time surveillance of people and their activities*", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (8) (2000) 809–830.
13. S. Intille, A. Bobick, "*Recognized planned, multiperson action*", International Journal of Computer Vision 81 (3) (2001) 414–445.
14. M. Isard, A. Blake, "*Condensation: conditional density propagation for visual tracking*", International Journal of Computer Vision 29 (1) (1998) 5–28.
15. I. Karaulova, P. Hall, A. Marshall, "*Tracking people in three dimensions using a hierarchical model of dynamics*", Image and Vision Computing 20 (2002) 691–700.
16. A. Kojima, T. Tamura, K. Fukunaga, "*Natural language description of human activities from video images based on concept hierarchy of actions*", International Journal of Computer Vision 50 (2) (2002) 171–184.
17. Y. Li, S. Ma, H. Lu, "*A multiscale morphological method for human posture recognition*", in: Proceedings of Third International Conference on Automatic Face and Gesture Recognition, Nara, Japan (1998) pp. 56–61.
18. L. Li, W. Huang, I. Gu, Q. Tian, "*Statistical modeling of complex backgrounds for foreground object detection*", IEEE Transactions on Image Processing 11 (13) (2004) 1459–1472.
19. A. Lipton, H. Fujiyoshi, R. Patil, "*Moving target classification and tracking from real-video*", in: IEEE Workshop on Applications of Computer Vision (WACV'98), Princeton, NJ (1998) pp. 8–14.
20. M. Ma, P. McKevitt, "*Interval relations in lexical semantics of verbs*", Artificial Intelligence Review 21 (3–4) (2004) 293–316.
21. O. Masoud, N. Papanikolopoulos, "*A method for human action recognition*", Image and Vision Computing 21 (8) (2003) 729–743.
22. T. Moeslund, E. Granum, "*A survey of computer vision based human motion capture*", Computer Vision and Image Understanding 81 (3) (2001) 231–268.
23. R. Morris, D. Hogg, "*Statistical models of object interaction*", International Journal of Computer Vision 37 (2) (2000) 209–215.
24. H.-H. Nagel, "*From image sequences towards conceptual descriptions*", Image and Vision Computing 6 (2) (1988) 59–74.

25. H.-H. Nagel, "*Steps toward a cognitive vision system*", AI Magazine, Cognitive Vision 25 (2) (2004) 31–50.
26. A. Nakazawa, H. Kato, S. Hiura, S. Inokuchi, "*Tracking multiple people using distributed vision systems*", IEEE International Conference on Robotics and Automation (2002) pp. 2974–2981.
27. H. Ning, T. Tan, L. Wang, W. Hu, "*People tracking based on motion model and motion constraints with automatic initialization*", Pattern Recognition 37 (2004) 1423–1440.
28. K. Nummiaro and E. Koller-Meier, L.J. Van Gool, "*An adaptive color-based particle filter*", Image Vision Computing 21 (1) (2003) 99–110.
29. P. Remagnino, T. Tan, K. Baker, "*Agent oriented annotation in model based visual surveillance*", in: Proceedings of International Conference on Computer Vision (ICCV'98), Mumbai, India (1998) pp. 857–862.
30. Y. Ricquebourg, P. Bouthemy, "*Real-time tracking of moving persons by exploiting spatio-temporal image slices*", IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 797–808.
31. G. Sagerer, H. Niemann, "*Semantic networks for understanding scenes*", in: M. Levine (Ed.), Advances in Computer Vision and Machine Intelligence, Plenum, New York (1997).
32. A. Sanfeliu, J.J. Villanueva, "*An approach of visual motion analysis*", Pattern Recognition Letters 26 (3) (2005) 355–368.
33. K. Schäfer, "*Fuzzy spatio-temporal logic programming*", in: C. Brzoska (Ed.), Proceedings of 7th Workshop in Temporal and Non-Classical Logics – IJCAI'97, Nagoya, Japan (1997) pp. 23–28.
34. H. Sidenbladh, M. Black, L. Sigal, "*Implicit probabilistic models of human motion for synthesis and tracking*", in: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), Proceedings European Conference on Computer Vision (ECCV), Vol. 1, LNCS 2353, Springer, Denmark (2002) pp. 784–800.
35. C. Stauffer, W. Eric, L. Grimson, "*Learning patterns of activity using real-time tracking*", IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 747–757.
36. N. Ukita, T. Matsuyama, "*Real-time cooperative multiple-target tracking by communicating active vision agents*", Computer Vision and Iage Understanding 97 (2) (2005) 137–179.
37. S. Wachter, H.-H. Nagel, "*Tracking persons in monocular image sequences*", Computer Vision and Image Understanding 74 (3) (1999) 174–192.
38. D. Wagg, M. Nixon, "*Automated markerless extraction of walking people using deformable contour models*", Computer Animation and Virtual Worlds 15 (3–4) (2004) 399–406.
39. L. Wang, W. Hu, T. Tan, *"Recent developments in human motion analysis"*, Pattern Recognition 36 (3) (2003) 585–601.
40. M. Yamada, K. Ebihara, J. Ohya, "*A new robust real-time method for extracting human silhouettes from color images*", in: Proceedings of Third International Conference on Automatic Face and Gesture Recognition, Nara, Japan (1998) pp. 528–533.
41. Y. Zhang, E. Sung, E.C. Prakash, "*3D modeling of dynamic facial expressions for face image analysis and synthesis*", International Conference on Vision Interface, Canada (2001).

Chapter 7
3D Object Reconstruction from Uncalibrated Images Using an Off-the-Shelf Camera

Teresa C.S. Azevedo, João Manuel R.S. Tavares, and Mário A.P. Vaz

Abstract Three-dimensional (3D) objects reconstruction using just bi-dimensional (2D) images has been a major research topic in Computer Vision. However, it is still a hard problem to address, when automation, speed and precision are required and/or the objects have complex shapes or image properties. In this paper, we compare two Active Computer Vision methods frequently used for the 3D reconstruction of objects from image sequences, acquired with a single off-the-shelf CCD camera: Structure From Motion (SFM) and Generalized Voxel Coloring (GVC). SFM recovers the 3D shape of an object based on the relative motion involved, while VC is a volumetric method that uses photo-consistency measures to build the required 3D model. Both methods considered do not impose any kind of restrictions on the relative motion involved.

7.1 Introduction

Three-dimensional (3D) models built by computational systems are an intensive and long-lasting research problem for the Graphic and Computer Vision research communities. Since Computer Vision is concerned with the development of computational theories and methods for the automatic extraction of useful information

Teresa C.S. Azevedo
INEGI Instituto de Engenharia Mecânica e Gestão Industrial
LOME Laboratório de óptica e Mecânica Experimental
FEUP Faculdade de Engenharia da Universidade do Porto – Portugal
e-mail: teresa.azevedo@fe.up.pt

João Manuel R.S. Tavares and Mário A.P. Vaz
INEGI, LOME, FEUP
DEMEGI – Departamento de Engenharia Mecânica e Gestão Industrial
e-mail: tavares@fe.up.pt; gmavaz@fe.up.pt

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

from images, it offers the opportunity to build 3D models directly from real-world scenes with high accuracy and visual realism.

The main goal of this work was to compare two Computer Vision image-based methods commonly used for 3D objects reconstruction: *Structure From Motion* (*SFM*) and *Generalized Voxel Coloring* (*GVC*). This paper starts with an introduction to the state-of-art in 3D reconstruction, describing some commonly used reconstruction methods. After, some emphasis on *SFM* and *GVC* reconstruction methods is given. Then, the followed methodologies are described. After, some of the obtained experimental results are presented. Finally, some conclusions and guidelines for future work are given.

7.2 3D Reconstruction

Since most 3D reconstruction methods require considerable computational and imaging resources, there is always a trade-off between used hardware and software, computational complexity and results accuracy, realism and processing speed. In the last decades, the explosive growth in computers processing power and memory storage and their continuous reducing price, has enabled the common use of 3D reconstruction solutions in a variety of application fields, such as:

- Industry, for instance, in clothing industry (e.g. [1,2]), on-line measurements and production line control (e.g. [3,4])
- Navigation systems, for example, in autonomous vehicle guidance (e.g. [5, 6]) and pose estimation (e.g. [7, 8])
- Virtual reality, such as to build virtual actors, objects or environments (e.g. [9, 10]) and augmented/mixed reality (e.g. [11, 12])
- Biomedicine, in anthropometric studies (e.g. [13, 14]), detection of tumors or other deformations (e.g. [15, 16]), manufacturing of prosthetic devices (e.g. [17, 18]) and surgery planning (e.g. [19, 20]), for example
- Architecture/archaeology, for instance, in 3D architectural site reconstruction (e.g. [21, 22]) or archeological documentation (e.g. [23, 24])
- Security systems, like in visual surveillance (e.g. [25, 26]) and biometric or morphologic information retrieval (e.g. [27, 28])

7.3 Methods for 3D Reconstruction

The usually available methods for 3D reconstruction of objects are typically classified into *contact* or *non-contact*, Fig. 7.1.

Contact-based methods can achieve high accuracy levels and are suitable for a wide range of applications. However, these methods involve mechanical movement of a probe device from one measurement point to the next. Consequently, the data acquisition can be very time consuming. Moreover, since the probes collect only a

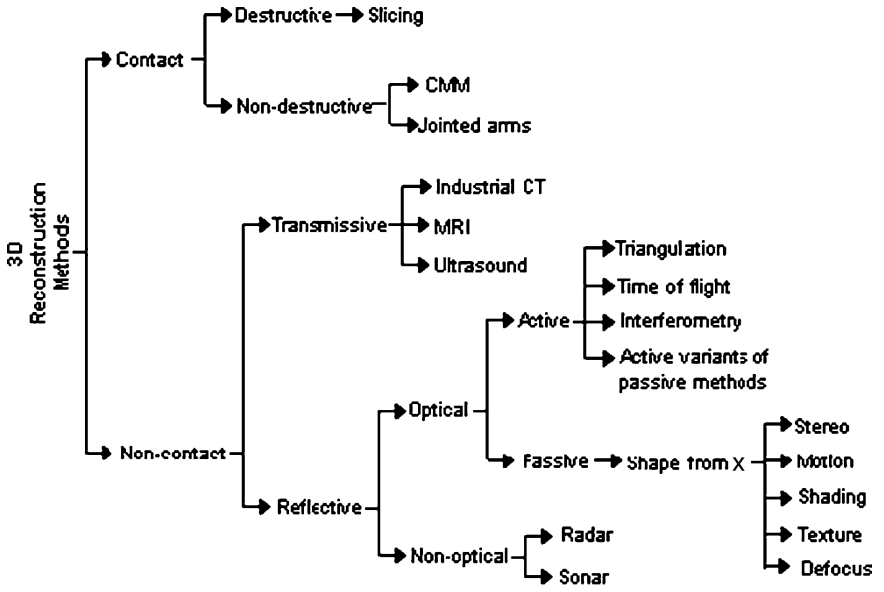


Fig. 7.1 Common division of usual 3D reconstruction methods

sparse data set from the object to be reconstructed, some of its critical areas might stay unmeasured. Also, the act of scanning the object by touching it can modify or even damage it, in particular if the object involved is very soft. *CMM*s (*Coordinate Measuring Machine*) capable of measuring objects of large dimensions are very large in size and so somewhat cumbersome to be used in usual production environments. Furthermore, frequently they need to be placed in controlled-environment rooms, for their protection against temperature variation and vibrations [29].

Nowadays, the generation of a 3D model is mainly achieved by using non-contact image-based methods. These are usually divided into two main groups [30]:

1. *Active*: methods that require some sort of energy projection (such as, lasers or structured light) or use the relative motion between camera(s) and objects, to obtain 3D information on the objects shape
2. *Passive*: methods that do not require energy projection or relative motion, and work under ambient illumination

Most common non-contact methods use image data, range sensors, or a combination of both. Image-based methods are widely used, in particular for industrial applications (e.g. [4, 31]), or for precise terrain and city modeling (e.g. [21, 32]).

Range sensors acquire distance measurements from a well known 3D reference coordinate system to the surface points on the object to be reconstructed. They are very common when highly detailed models are required and are already used in industry (e.g. [33,34]), for documentation of buildings and landscapes (e.g. [35,36]) or for the recording of objects in archaeology and cultural heritage (e.g. [37, 38]). However, they are costly (at least for now), spatially limited, most of the systems

available do not provide color information about the reconstructed object and the quality of the obtained 3D models can be affected by the reflective characteristics of the objects surfaces [39].

The main difference between image- and range-based methods is that, when using image data, it is necessary to have a mathematical model to derive the objects 3D coordinates, which can be sometimes a complex problem to solve [40]. Building 3D models using range methods is simpler, because the range data acquired already contains the 3D coordinates necessary for the 3D reconstruction.

The next two subsections will focus on two commonly used image-based reconstruction methods: *Structure From Motion* (*SFM*), that belongs to the standard stereo-based methods, and *Generalized Voxel Coloring* (*GVC*), that belongs to the more recent volumetric reconstruction methods.

7.3.1 Structure from Motion

Proposed in [41], *SFM* is a stereo-based method, Fig. 7.2. It uses the relative motion between the camera(s) used and the objects to be reconstructed, to make assumptions about the 3D objects shape. Thus, by knowing the trajectories of objects feature points in the image plane, this method determines the 3D shape and motion that better describes most of the trajectories of the referred points.

This method has received several contributions and diverse approaches: for example, in [42] the use of an extended *Kalman* filter was investigated for estimating the motion and structure from a sequence of monocular images; in [43] an algorithm was developed for shape and motion estimation under orthographic projection using the factorization projection; in [44] a method was proposed that computes the final reconstruction from intermediate reconstructions by analyzing the uncertainties in them, rather than from image data directly; in [45] the problem

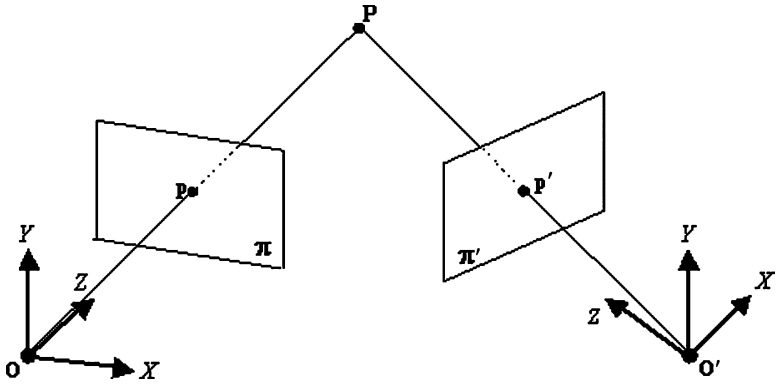


Fig. 7.2 Stereo vision principle: 3D coordinates of point **P** are determined through the intersection of the two lines defined by the optical centers **O** and **O′** and the matched 2D image points **p** and **p′**

of solving the *SFM* issue without prior knowledge of point correspondence was addressed; more recently, in [46], an holistic approach was used to compute *SFM* in stages by gradually computing 3D scene information of increasing complexity through processes which operate on increasingly large spatial image areas; among many others. However, *SFM* may suffer from difficulties on finding interest points and/or matching them along the input image sequence [47]. First, if the object to reconstruct has a smooth surface and low texture, the extraction of interest features may be difficult or even incorrect since the local appearance is uniform within the neighborhood of each candidate feature. Secondly, matching correspondence cannot be established by just comparing local image measurements, unless the object has a *lambertian* surface; that is, its appearance does not change with the viewpoint. Finally, occlusions in the scene make the correspondence between images difficult or even impossible to obtain.

7.3.2 Generalized Voxel Coloring

As referred earlier, stereo-based methods, like *SFM*, fail to capture objects with complex shapes, smooth surfaces with lack of texture or when occlusion phenomena occur.

For smooth objects, 3D reconstruction using volumetric methods have been quite popular for some time [48]. These methods are silhouette-based reconstruction methods: intersecting the visual cones generated by the silhouettes and the projection centers of each image, a 3D model can be determined, Fig. 7.3. This 3D model is denominated as *visual hull* [49], a locally convex over-approximation of the volume occupied by an object.

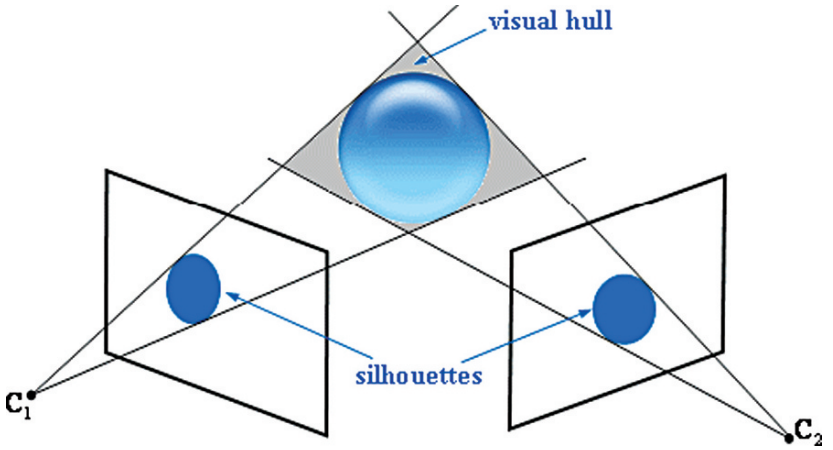


Fig. 7.3 *Visual hull* obtained from two different viewpoints (C_1, C_2)

Volumetric methods represent the 3D space model by using *voxels* (regular volumetric structures also known as 3D pixels). The space of interest is divided into discrete voxels which are then classified into two categories: inside and outside. The union of all the inside voxels is an approximation of the visual hull. The accuracy of the reconstruction obtained depends on the number of images used, the positions of each viewpoint considered, the precision of the camera calibration and the complexity of the objects shape.

Generalized Voxel Coloring (*GVC*) is a volumetric method that uses photo-consistency criterion, Fig. 7.4, to determine if a certain voxel belongs or not to the object being reconstructed. With this method, the resulting 3D model is the *photo hull*, Fig. 7.5, defined as the largest volume of voxels that are photo-consistent with all viewpoints considered in the reconstruction. Photo-consistency is checked statistically: a voxel is considered consistent if the mean deviation of the pixels color, which results from the voxel image projection, is under a predefined threshold. Thus, *GVC* simultaneously builds and colors the obtained 3D model.

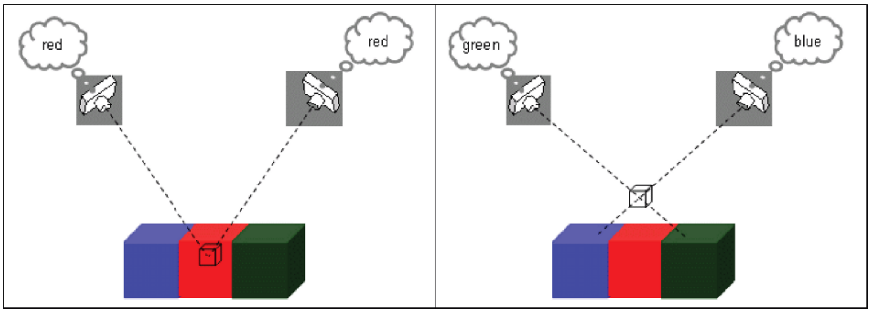


Fig. 7.4 Color consistency: if the voxel is inside the object surface it will reproject the same color onto all viewpoints where it is visible (left); otherwise, if the voxel is outside the object surface it will most likely reproject distinct colors

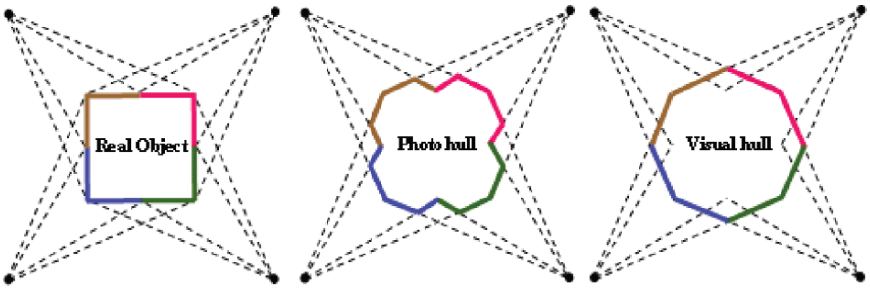


Fig. 7.5 Relation between *photo* and *visual hull*: the real object is contained inside the *photo hull* which is in turn inside the *visual hull*

7.4 Methodologies Followed

In this work, *SFM* and *GVC* methods were tested on two objects with different shape properties: a simple parallelepiped and a human hand model.

The parallelepiped has a straightforward topology, with flat orthogonal surfaces, whose vertices are easily detected in each image and simply matched along the acquired image sequence. On the contrary, the hand model has a smooth surface and a more complicated shape.

7.4.1 SFM Methodology

To test the *SFM* method, we follow the methodology proposed in [50], and resumed in Fig. 7.6:

1. The first step is to acquire two uncalibrated images, of the object to be reconstructed, using a single off-the-shelf digital camera.
2. Then, image feature points of the considered object are extracted. Feature or interesting points are those who reflect the relevant discrepancies between their intensity values and those of their neighbors. Usually, these points represent vertices, and their correct detection allows posterior matching along the image sequences acquired. Many algorithms for interest points detection are available, but the point features detectors based on the *Harris*'s principles [51], are the most commonly used.

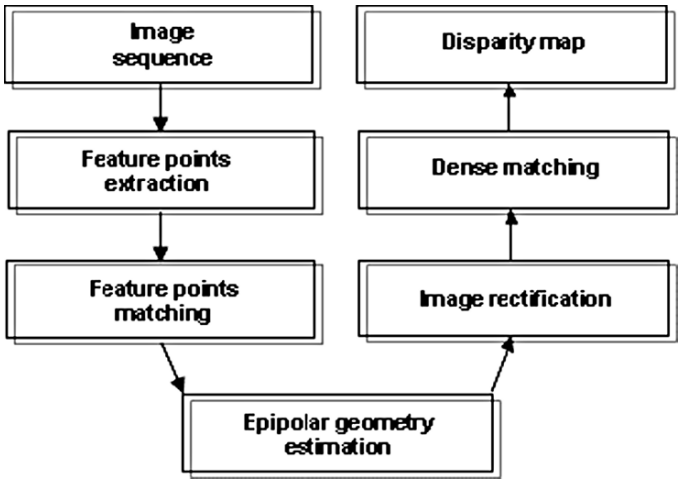


Fig. 7.6 *SFM* methodology followed for the 3D reconstruction of objects

3. After being extracted, feature points must be matched. The matching process is a 2D points association between sequential images that are the projection of the same 3D object point. Automatic detection of matching points between images can be achieved using several cross-correlation processes. They all use small image windows from a first image as templates for matching in the subsequent images [52]. The most common matching methods include *Normalized Cross-Correlation* [53, 54], and *Sum-of-Squared-Differences* [50, 55].
4. Then the epipolar geometry is estimated. Epipolar geometry determines a pairwise relative orientation and allows for rejection of previous false matches (or *outliers*). When the interior orientation parameters of both images are the same, it mathematically expresses itself by the *fundamental matrix*, a projective singular correlation between two images [56]. At least seven matches are required to compute the fundamental matrix, but to cope with possible outliers, robust methods of estimation are required. In general, the *RANSAC – RANdom Sampling Consensus* – algorithm [57], achieves a robust estimation of the epipolar geometry.
5. Next step is image rectification. It is the act of projecting two stereo images onto a common plane, such that pairs of conjugate epipolar lines (derived from the fundamental matrix) become collinear and parallel to one of the image axes. Performing this step simplifies the posterior process of dense matching, because the search problem is reduced to 1D.
6. Finally, dense matching is performed, where a disparity map is obtained. A disparity map codifies the distance between the object and the camera(s): closer points will have maximal disparity and farther points will get zero disparity. For short, a disparity map gives some perception of discontinuity in terms of depth (2.5D reconstruction).

If the camera were calibrated, the obtained 2.5D reconstruction could be upgraded to 3D, using the triangulation concept, described in Fig. 7.2.

7.4.2 GVC Methodology

To test the *GVC* method we follow the methodology proposed in [58], and represented in Fig. 7.7.

In this methodology, it is necessary to acquire two image sequences:

- A first one, acquired moving a planar chessboard calibration pattern freely in 3D space.
- For the second sequence, the object to reconstruct is placed on a simple turntable device, with the same chessboard pattern beneath it; keeping the camera untouched, the second sequence of images is acquired, spinning the turntable device until a full rotation is performed.

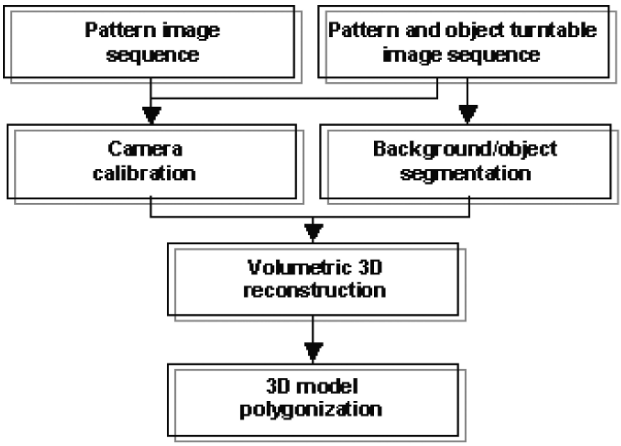


Fig. 7.7 *GVC* methodology followed for the 3D reconstruction of objects

No restrictions are made on the number of images acquired, nor the rotation angle between two consecutive images of the second image sequence needs to be known.

Then, the used camera is calibrated, in order to find the transformation that maps the 3D world in the associated 2D image space. The calibration procedure is based on *Zhang*'s algorithm [59]. Intrinsic parameters (focal length and principal point) and distortion parameters (radial and tangential) are obtained from the first image sequence; using the second image sequence, the extrinsic parameters (rotation and translation) associated with each viewpoint considered in the reconstruction process are determined.

Then, to obtain the object silhouettes from the input images, image segmentation is performed. This step is required, because, even when the scene background has low color variation, the photo-consistency criterion may not be sufficient for accurate 3D reconstructions [60]. Also, since the used calibration pattern will rotate along with the object to be reconstructed, it will not be considered has background and, consequently, will be reconstructed as if it was part of the object of interest. Images are here segmented by first removing the red and green channels from the original RGB images and, finally, by image binarization using a user-defined threshold value.

Combining the original image sequence and associated silhouette images, and considering the previously obtained camera calibration parameters, the 3D models are built using the *GVC* volumetric method implemented in [61].

Finally, the volumetric model obtained is polygonized and smoothed using the *Marching Cubes* algorithm [62]. Basically, this algorithm extracts a polygonal surface from the volumetrical data. Thus, it proceeds through the voxelized model, and, for each voxel, it determines the polygon(s) needed to represent the patch of the isosurface that passes through the referred voxel.

7.5 Experimental Results

In this section, some of the obtained experimental results for both followed methodologies and both considered objects will be presented and analyzed.

7.5.1 SFM Method

Figure 7.8 shows the acquired stereo image pairs of both objects used in this work.

For both objects, 200 image features were extracted using the *Harris*'s corner detector [51], imposing a minimum distance between each detected feature. Robust matching of features between the stereo images was made using the *RANSAC* algorithm [57]. The results obtained can be observed in Fig. 7.9. Since the hand model presents a smooth surface, obviously many wrong matches were detected and, consequently, the determined epipolar geometry will be incorrectly estimated.

After, both stereo pairs were rectified using the algorithm presented in [63]. As observed in Figs. 7.10 and 7.11, the results were much less accurate for the hand

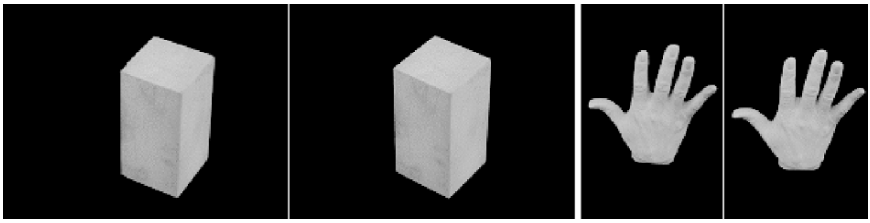


Fig. 7.8 Stereo image pairs of the objects used to test the *SFM* reconstruction method

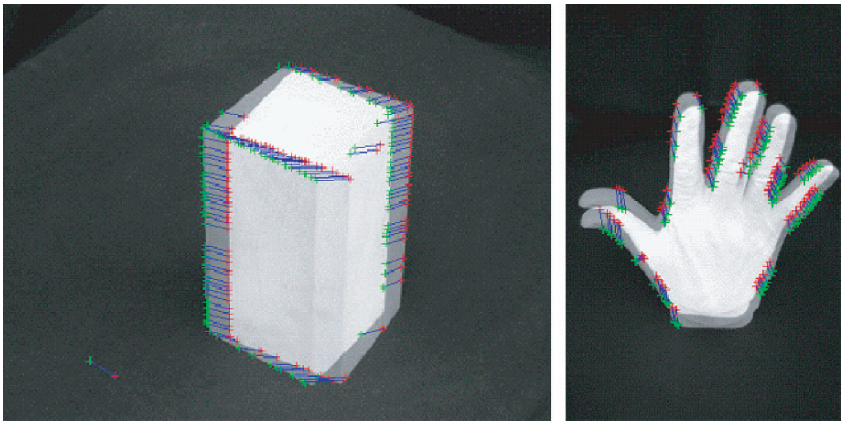


Fig. 7.9 Results of the (robust) feature points matching for both objects considered: green crosses represent the matched feature points of the first image and the red crosses represent the correspondent matched feature points of the second image

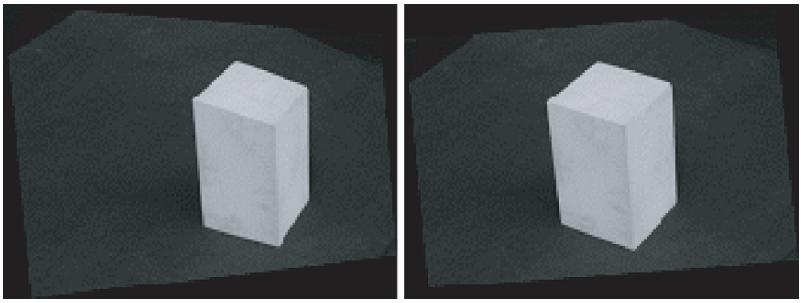


Fig. 7.10 Rectification results for the stereo images of the parallelepiped object

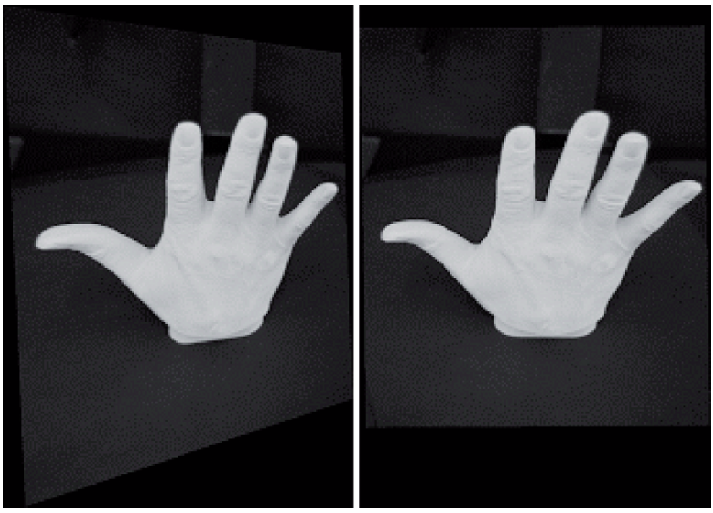


Fig. 7.11 Rectification results for the stereo images of the hand model object

model, due to the wrong matches from the previous step. This caused a strong image distortion during the rectification step for this object.

Then, dense matching was performed using *Stan Birchfield*'s algorithm [64]. The results obtained for both objects considered in this work can be observed in Figs. 7.12 and 7.13. Again, from the incorrect results obtained in the previous steps, the dense matching for the hand model was, consequently, of low quality. For the parallelepiped object case, the generated disparity map matches reality better.

7.5.2 GVC Method

Figure 7.14 shows some examples of the second image sequence acquired for the 3D reconstruction of both objects using the *GVC* method.

Fig. 7.12 Disparity map obtained for the parallelepiped object

Fig. 7.13 Disparity map obtained for the hand model object

For both objects considered, the results of the extrinsic calibration procedure are represented in Fig. 7.15. The 3D graphics shown represent the viewpoints considered in the second image acquisition process, considering the world coordinate system fixed on the lower-left corner of the chessboard pattern and the camera rotating around the object.

Another way to verify the accuracy of the calibration results obtained is to reproject the 3D points from the chessboard pattern in all images of the second sequence

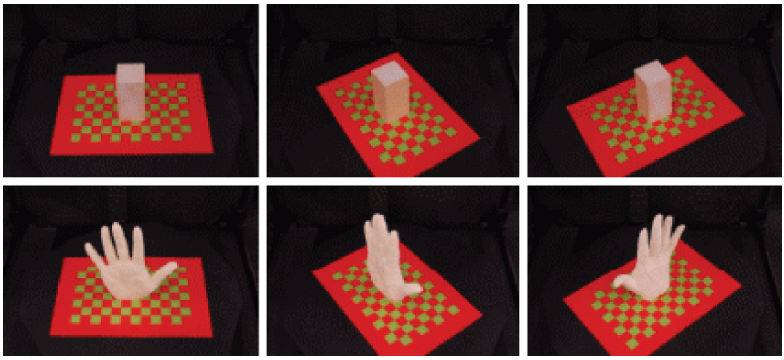


Fig. 7.14 Three images used for the 3D reconstruction of the parallelepiped (top) and the hand model (bottom)

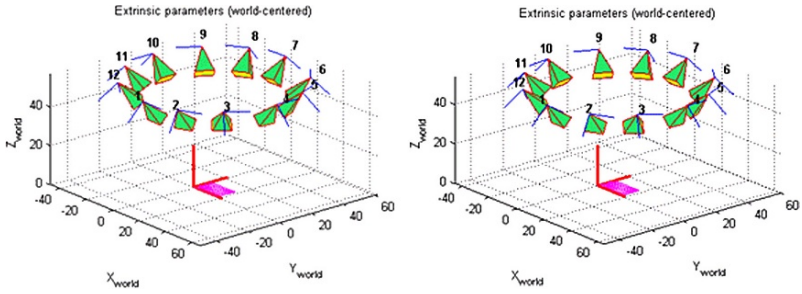


Fig. 7.15 Three-dimensional graphical representation of the extrinsic parameters obtained from the camera calibration process for the parallelepiped object case, on the left, and for the hand model case, on the right

Table 7.1 Error of the reprojection of the pattern points into all images of the second image sequence

Object	Reprojection error (in pixels)			
	Average		Standard deviation	
	X	*Y*	*X*	*y*
Parallelepiped	−1.24e-04	−2.67e-05	0.545	0.594
Hand model	−7.31e-05	−2.61e-05	0.673	0.840

considered. The standard deviations of the reprojection errors (in pixels) for the hand and torso models cases are indicated in Table 7.1. The results obtained from the camera calibration were very accurate for both cases.

The efficacy of our segmentation method enabled us to obtain good silhouette images for both considered objects, Fig. 7.16.

Figures 7.17 and 7.18 show the results of the 3D reconstruction obtained for both objects using the *GVC* method. Both reconstructed models are very similar to the real 3D object, even in the case of the hand model. Comparing these results with

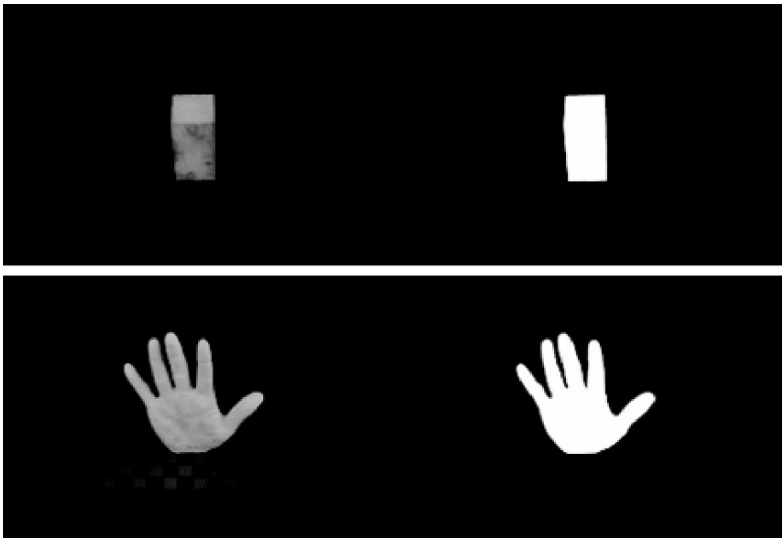


Fig. 7.16 One example of image segmentation for the parallelepiped (top) and the hand model (bottom): on the left, the blue channel of the original image; on the right, the binary image obtained

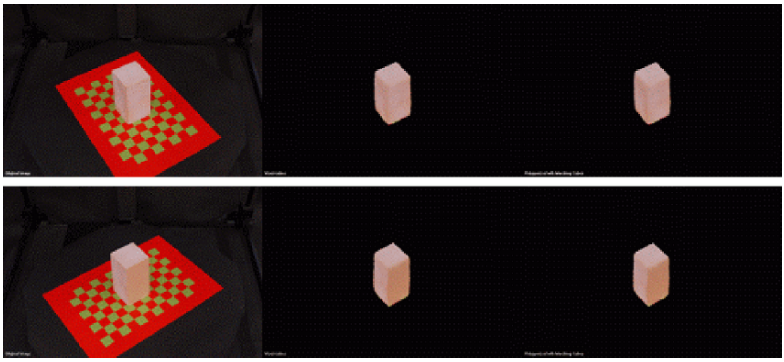


Fig. 7.17 Two different viewpoints (by row) of the 3D model obtained for the parallelepiped case: on the left, original image; in the centre, voxelized 3D model; on the right, polygonized and smoothed 3D model

the previous obtained by the *SFM* methodology, *GVC* has no problem to reconstruct objects with smooth and complex shapes. On the other hand, the accuracy of the 3D models built by this last methodology is highly dependent on the calibration and segmentation steps. Thus, *GVC* puts some restrictions, such as a background with low color variation and suitable calibration apparatus, making it less appropriated for unconstrained real-world reconstructions.

GVC methodology was also tested on another object to verify its accuracy: a torso model. Comparing with the previous used objects, the torso has considerably higher

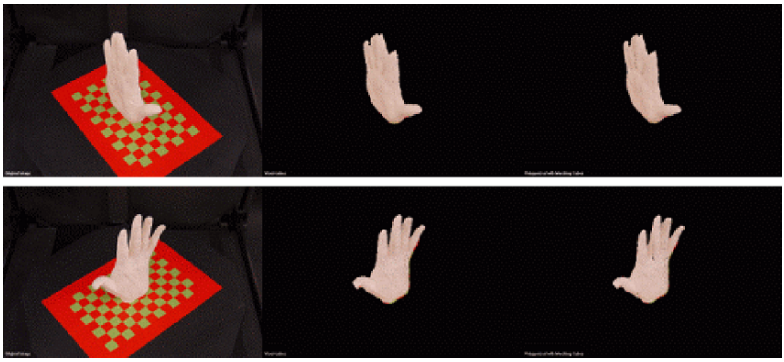


Fig. 7.18 Two different viewpoints (by row) of the 3D model obtained for the hand model case: on the left, original image; in the centre, voxelized 3D model; on the right, polygonized and smoothed 3D model

dimensions. Thus, a different calibration pattern was required. Some of the results obtained after the reconstruction process can be seen on Fig. 7.19, where it can be noticed that the torso reflects the calibration pattern on its surface. As consequence, the inferior zone of the reconstructed 3D model is not very accurate, both in terms of shape and color.

From the voxelized 3D model obtained, some geometrical measures can be determined, such as height, length and width. Figure 7.20 compares these values with the real ones, obtained using an usual ruler, for all reconstructed objects. This comparison confirms the approximated reconstruction results of the considered objects, using the *GVC* methodology.

7.6 Conclusions

The main goal of this paper was to compare experimentally two commonly used image-based methods for 3D object reconstruction: *Structure From Motion* (*SFM*) and *Generalized Voxel Coloring* (*GVC*).

To test and compare both methods, two objects with different shape properties were used: a parallelepiped and a hand model.

Our adopted *SFM* methodology produced fine results when the objects present strong feature points, and so, are easy to detect and match along the input images. However, we can conclude that even small errors in the matching or in the epipolar geometry estimation can seriously compromise the success of the remaining steps.

The models built using the *GVC* methodology were quite similar to the real objects, be it in terms of shape or in color. Nevertheless, the reconstruction accuracy was highly dependent on the quality of the results from camera calibration and image segmentation steps. These can be two major drawbacks in real-world scenes, because they can limit the application of the *GVC* method. Moreover, the reflectance

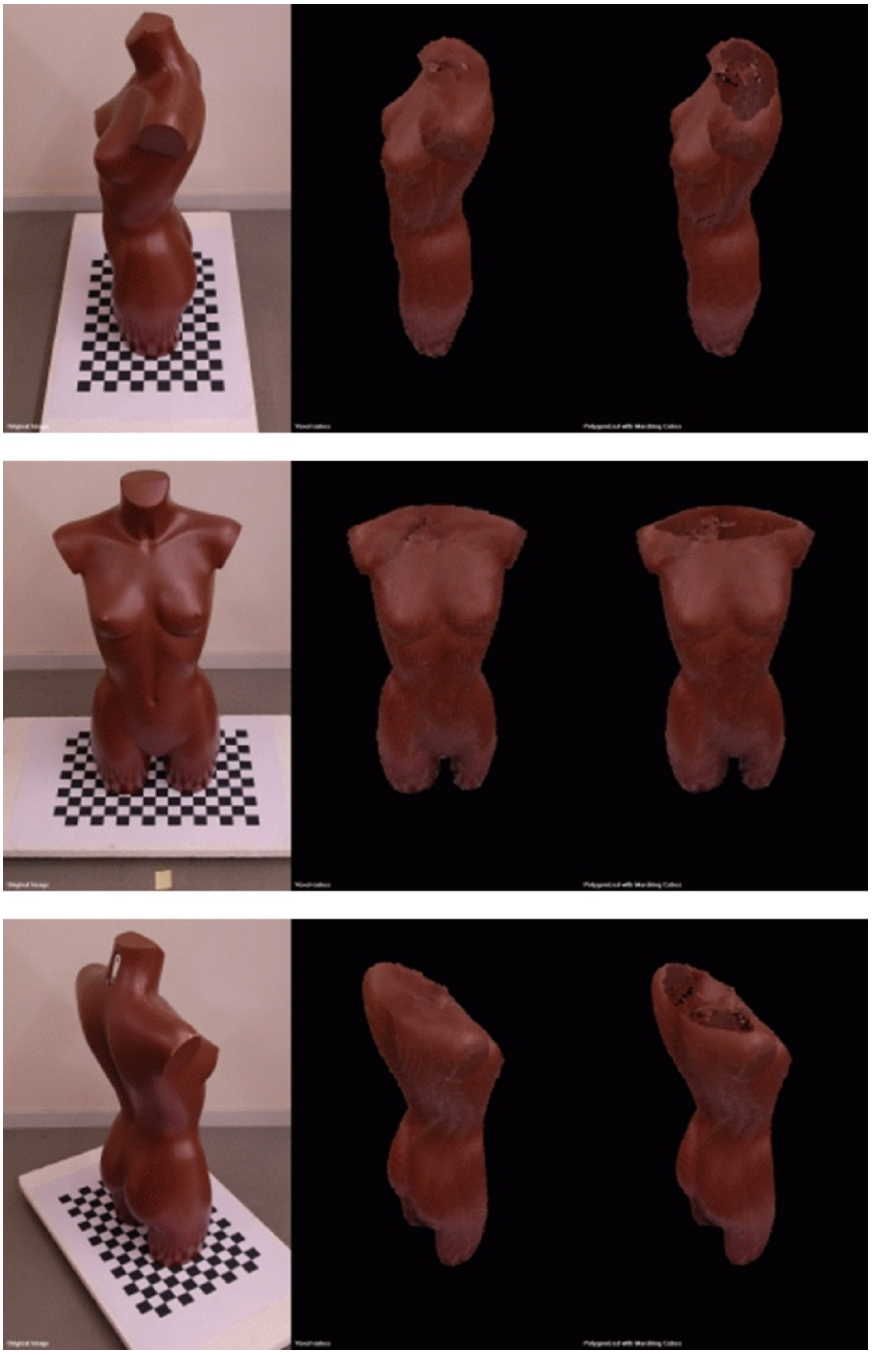


Fig. 7.19 Three different viewpoints (by row) of the 3D model obtained for the torso model case: on the left, original image; in the centre, voxelized 3D model; on the right, polygonized and smoothed 3D model

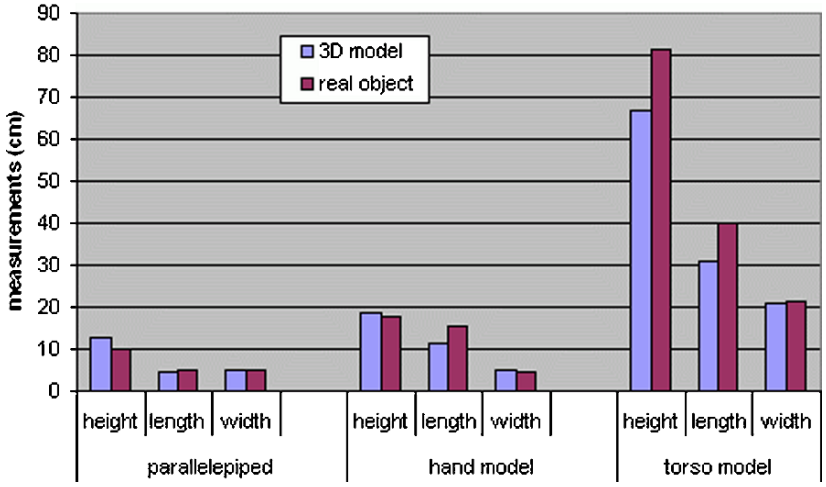


Fig. 7.20 Comparison of the obtained measurements from the reconstructed 3D models with the real objects measures

of their surfaces is an aspect that must be considered for more accurate 3D reconstructions. In resume, we can conclude that in controlled environments the *GVC* methodology is capable to obtain adequate 3D static reconstructions of objects from images. In addition, its major contribution may be the fact that it is fully automatic and suitable for many real applications.

Thus, when comparing the two methods, we can conclude that, on one hand, *GVC* performs better in 3D reconstruction of objects with complex shapes and, on the other hand, *SFM* is better for unconstrained real-world objects reconstruction.

Acknowledgements This work was partially done in the scope of project "Segmentation, Tracking and Motion Analysis of Deformable (2D/3D) Objects using Physical Principles", with reference POSC/EEA-SRI/55386/2004, financially supported by *FCT – Fundação para a Ciência e a Tecnologia* from Portugal.

The first author would like to thank the support of the PhD grant SFRH/BD/27716/2006, also from *FCT*.

References

1. T. I. Vassilev (2000). Dressing Virtual People, Systemics, Cybernetics and Informatics, Orlando, FL.
2. P. Volino, F. Cordier and N. Magnenat-Thalmann (2005). From early virtual garment simulation to interactive fashion design, Computer-Aided Design, 37(6): 593–608.
3. S. M. Bhandarkar, T. D. Faust and M. Tang (1999). CATALOG: a system for detection and rendering of internal log defects using computer tomography, Machine Vision and Applications, 11: 171–190.

4. L. Song, X. Qu and S. Ye (2007). Improved SFS 3D measurement based on BP neural network, Image and Vision Computing, 25(5): 614–622.
5. D. Cobzas, H. Zhang and M. Jagersand (2003). Image-based localization with depth-enhanced image map, IEEE International Conference on Robotics and Automation, Taipeh, Taiwan, 2: 1570–1575.
6. K. O. Arras, N. Tomatis, B. T. Jensen, et al. (2001). Multisensor on-the-fly localization: precision and reliability for applications, Robotics and Autonomous Systems, 34(2–3): 131–143.
7. A. J. Davison, A. G. Cid and N. Kita (2004). Real-Time 3D SLAM with Wide-Angle Vision, 5th IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal.
8. T. Lemaire, C. Berger, I. -K. Jung, et al. (2007). Vision-based SLAM: stereo and monocular approaches, International Journal of Computer Vision, 74(3): 343–364.
9. J. -C. Nebel, A. Sibiryakov and X. Ju (2003). V-Man Generation for 3-D Real Time Animation, a Symposium on Intelligent Motion and Interaction Within Virtual Environments, London.
10. A. Koutsoudis, F. Arnaoutoglou and C. Chamzas (2007). On 3D reconstruction of the old city of Xanthi. A minimum budget approach to virtual touring based on photogrammetry, Journal of Cultural Heritage, 8(1): 26–31.
11. H. Aans and F. Kahl (2002). Estimation of Deformable Structure and Motion, Vision and Modelling of Dynamic Scenes Workshop, Copenhagen, Denmark.
12. B. Vagvolgyi, C. Reiley, G. Hager, et al. (2008). Augmented reality using registration of 3D computed tomography to stereoscopic video of laparoscopic renal surgery, The Journal of Urology, 179(4): 241–242.
13. P. Tukuisis, P. Meunier and C. E. Jubenville (2001). Human body surface area: measurement and prediction using three dimensional body scans, European Journal of Applied Physiology and Occupational Physiology, 85(3–4): 264–271.
14. L. Benton and J. -C. Nebel (2002). Study of the breathing pattern based on 4D data collected by a dynamic 3D body scanner, 7th Numérisation 3D/Scanning, Paris, France.
15. C. Beneder, F. G. Fuechsel, T. Krause, et al. (2008). The role of 3D fusion imaging in sentinel lymphadenectomy for vulvar cancer, Gynecologic Oncology, 109(1): 76–80.
16. G. D. Giammarco, M. L. Storto, R. Marano, et al. (2006). Superior vena cava syndrome: a 3D CT-scan reconstruction, European Journal of Cardio-Thoracic Surgery, 30(2): 384–385.
17. N. H. Mahmood and T. Tjahjadi (2006). 3D Reconstruction From Multiple Views for Orthotic and Prosthetic Design: An Overview, SCOReD - Student Conference on Research and Development, Shah Alam, Selangor, Malaysia, 27–28.
18. S. I. Buchaillard, S. H. Ong, Y. Payan, et al. (2007). 3D statistical models for tooth surface reconstruction, Computers in Biology and Medicine, 37(10): 1461–1471.
19. A. Pednekar and I. A. Kakadiaris (2000). Applications of Virtual Reality in Surgery, Indian Conference on Computer Vision, Graphics and Image Processing, Graphics and Applications session, Bangalore, India.
20. J. P. Helferty, A. J. Sherbondy, A. P. Kiraly, et al. (2007). Computer-based system for the virtual-endoscopic guidance of bronchoscopy, Computer Vision and Image Understanding, 108(1–2): 171–187.
21. D. P. Robertson and R. Cipolla (2002). Building Architectural Models from Many Views Using Map Constraints, 7th European Conference on Computer Vision-Part II, Copenhagen, Denmark, 155–169.
22. S. Lee, D. Feng and B. Gooch (2008). Automatic Construction of 3D Models from Architectural Line Drawings, Symposium on Interactive 3D Graphics and Games, Redwood City, California, 123–130.
23. R. Sablatnig, S. Tosovic and M. Kampel (2002). Combining Shape from Silhouette and Shape from Structured Light for Volume Estimation of Archaeological Vessels, International Conference on Pattern Recognition, Quebec City, Canada 1: 364–367.
24. M. Kampel and R. Sablatnig (2007). Rule based system for archaeological pottery classification, Pattern Recognition Letters, 28(6): 740–747.
25. A. Calbi, C. S. Regazzoni and L. Marcenaro (2006). Dynamic Scene Reconstruction for Efficient Remote Surveillance, IEEE International Conference on Video and Signal Based Surveillance, Sydney, Australia, 99.

26. S. Fleck, F. Busch, P. Biber, et al. (2006). 3D Surveillance a Distributed Network of Smart Cameras for Real-Time Tracking and its Visualization in 3D, IEEE Computer Vision and Pattern Recognition, Workshop on Embedded Computer Vision, New York, 118.
27. N. Lynnerup, M. Andersen and H. P. Lauritsen (2003). Facial image identification using Photomodeler®, Legal Medicine, 5(3): 156–160.
28. J. Lee, E.-D. Lee, H.-O. Tark, et al. (2008). Efficient height measurement method of surveillance camera image, Forensic Science International, 177(1): 17–23.
29. R. Gershon and M. Benady (2001). Noncontact 3-D measurement technology enters a new era, http://www.qualitydigest.com/sept01/html/3d.html (retrieved in June 2007).
30. S. M. Seitz (1999). An Overview of Passive Vision Techniques, SIGGRAPH 2000 Course on 3D Photography, Course Notes, New Orleans, Louisiana, LA.
31. S. M. Youssef and R. M. Salem (2007). Automated barcode recognition for smart identification and inspection automation, Expert Systems with Applications, 33(4): 968–977.
32. Y. Yu and J. Malik (1998). Recovering photometric properties of architectural scenes from photographs, Computer Graphics, 32: 207–217.
33. G. Zhang, J. He and X. Li (2005). 3D vision inspection for internal surface based on circle structured light, Sensors and Actuators A: Physical, 122(1): 68–75.
34. C. P. Witana, S. Xiong, J. Zhao, et al. (2006). Foot measurements from three-dimensional scans: a comparison and evaluation of different methods, International Journal of Industrial Ergonomics, 36(9): 789–807.
35. C. Früh and A. Zakhor (2004). An automated method for large-scale, ground-based city model acquisition, International Journal of Computer Vision, 60(1): 5–24.
36. J. Chen and B. Chen (2008). Architectural modeling from sparsely scanned range data, International Journal of Computer Vision, 78(2–3): 223–236.
37. M. Levoy, K. Pulli, B. Curless, et al. (2000). The Digital Michelangelo Project: 3D Scanning of Large Statues, ACM SIGGRAPH 2000 - International Conference on Computer Graphics and Interactive Techniques, New Orleans, Louisiana, 131–144, ACM Press, Addison Wesley, New York.
38. G. Guidi, B. Frischer, M. Russo, et al. (2006). Three-dimensional acquisition of large and detailed cultural heritage objects, Machine Vision and Applications, 17(6): 349–360.
39. F. Remondino and S. El-Hakim (2006). Image-based 3D modelling: a review, The Photogrammetric Record, 21(115): 269–291.
40. F. Remondino, A. Guarnieri and A. Vettore (2005). 3D modeling of close-range objects: photogrammetry or laser scanning?, SPIE-IS&T Electronic Imaging, 5665: 216–225.
41. S. Ullman (1979). The Interpretation of Visual Motion, MIT Press, Cambridge, MA.
42. T. Broida and R. Chellappa (1991). Estimating the kinematics and structure of a rigid object from a sequence of monocular images, Pattern Analysis and Machine Intelligence, 13: 497–513.
43. C. Tomasi and T. Kanade (1992). Shape and motion from image streams under orthography: a factorization method, International Journal of Computer Vision, 9: 137–154.
44. J. Oliensis (1999). A multi-frame structure-from-motion algorithm under perspective projection, International Journal of Computer Vision, 34: 1–30.
45. F. Dellaert, S. Seitz, C. Thorpe, et al. (2000). Structure from motion without correspondence, IEEE Computer Vision and Pattern Recognition, Hilton Head, SC, 2: 557–564.
46. J. Hui (2006). A holistic approach to structure from motion, PhD on Computer Science, University of Maryland, MD.
47. R. Hartley and A. Zisserman (2004). Multiple View Geometry in Computer Vision, 2nd ed., Cambridge University Press, Cambridge.
48. S. Seitz and C. R. Dyer (1997). Photorealistic Scene Reconstruction by Voxel Coloring, Computer Vision and Pattern Recognition Conference, San Juan, Puerto Rico, 1067–1073.
49. A. Laurentini (1994). The visual hull concept for silhouette-based image understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(2): 150–162.
50. M. Pollefeys, L. V. Gool, M. Vergauwen, et al. (2004). Visual modeling with a hand-held camera, International Journal of Computer Vision, 59(3): 207–232.

51. C. G. Harris and M. J. Stephens (1988). A Combined Corner and Edge Detector, Fourth Alvey Vision Conference, University of Manchester, UK, 15: 147–151.
52. R. C. Gonzalez and P. Wintz (1987). Digital Image Processing, 2nd ed., Addison Wesley, Reading, MA.
53. P. J. Burt, C. Yen and X. Xu (1982). Local Correlation Measures for Motion Analysis: A Comparative Study, Pattern Recognition and Image Processing, Las Vegas, NV, 269–274.
54. J. P. Lewis (1995). Fast Normalized Cross Correlation, Proceedings Vision Interface (VI '95), Quebec, 120–123.
55. A. Moore (2006). Stereo, Computer Vision Lectures, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
56. O. Faugeras, Q. -T. Luong and S. J. Maybank (1992). Camera Self-Calibration: Theory and Experiments, 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy, Lecture Notes in Computer Vision, Springer, 588: 321–334.
57. M. A. Fischler and R. Bolles (1981). RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography, Communications of the ACM, New York, 24(6): 381–395.
58. T. C. S. Azevedo, J. M. R. S. Tavares and M. A. P. Vaz (2007). 3D Volumetric Reconstruction and Characterization of Objects from Uncalibrated Images, 7th IASTED International Conference on Visualization, Imaging, and Image Processing, Palma de Mallorca, Spain, 141–146.
59. Z. Zhang (2000). A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11): 1330–1334.
60. K. Sande (2004). A Practical Setup for Voxel Coloring using off-the-shelf Components, Bachelor Project, Universiteit van Amsterdam, The Netherlands.
61. M. Loper (2002). Archimedes: Shape Reconstruction from Pictures - A Generalized Voxel Coloring Implementation, http://matt.loper.org/Archimedes/(retrieved in August 2006).
62. W. E. Lorensen and H. E. Cline (1987). Marching cubes: a high resolution 3D surface construction algorithm, ACM SIGGRAPH Computer Graphics, New York, NY, 21(4): 163–169.
63. F. Isgrò and E. Trucco (1999). Projective rectification without epipolar geometry, IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, CO, 1: 94–99.
64. S. Birchfield (1999). Depth discontinuities by pixel-to-pixel stereo, International Journal of Computer Vision, 35(3): 269–293, http://vision.stanford.edu/∼birch/p2p/(retrieved in November 2004).

Chapter 8
Edge-Images Using a Uninorm-Based Fuzzy Mathematical Morphology: Opening and Closing

Manuel González-Hidalgo, Arnau Mir Torres, Daniel Ruiz-Aguilera, and Joan Torrens Sastre

Abstract In this paper a fuzzy mathematical morphology based on fuzzy logical operators is proposed and the Generalized Idempotence (GI) property for fuzzy opening and fuzzy closing operators is studied. It is proved that GI holds in fuzzy mathematical morphology when the selected fuzzy logical operators are left-continuous uninorms (including left-continuous t-norms) and their corresponding residual implications, generalizing known results on continuous t-norms. Two classes of left-continuous uninorms are emphasized as the only ones for which duality between fuzzy opening and fuzzy closing holds. Implementation results for these two kinds of left-continuous uninorms are included. They are compared with the classical umbra approach and the fuzzy approach using t-norms, proving that they are specially adequate for edge detection.

8.1 Introduction

The identification of objects, object feature extraction and anomalies detection in automated industrial processes are closely connected with the recognition of shapes and therefore with the recognition or vision systems. In this context the mathematical morphology is an useful tool for extracting image components that are useful in the representation and description of region shapes, such as boundaries, skeletons, and convex hull.

M. González-Hidalgo
Computer Graphics, Vision and Artificial Intelligence Group,
Mathematics and Computer Science Department University of the Balearic Islands, Spain
e-mail: manuel.gonzalez@uib.es

A. Mir-Torres, D. Ruiz-Aguilera, and J. Torrens-Sastre
Fuzzy Logic and Information Fusion Group,
Mathematics and Computer Science Department University of the Balearic Islands, Spain
e-mail: arnau.mir@uib.es, daniel.ruiz@uib.es, dmijts0@uib.es

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

The basic tools of mathematical morphology are the morphological operators. A morphological operator P transforms the structure that we want to analyze A (an image) by means of a small object B, called *structuring element*, with which we want to probe the structure of A, into a new object $P(A,B)$ (a new image). The size and shape of B can be chosen by the morphologist in order to analize the structure of A. The basic morphological operators are the dilation and erosion. These operators are based on set theory and were originally developed for binary images (black and white) and afterwards successfully extended to gray-scale images ([22]).

Nevertheless, the shapes in an image are not always crisply defined, and uncertainty can arise within each level of image analysis and pattern recognition. It can occur at the low-level in the raw sensor output, and it can be extended all the way through intermediate and higher levels. A recognition or computer vision system must have sufficient flexibility for processing the uncertainty in any of these levels, so that the system could retain as much of the information content of the data as possible, at each level. As the first essential step of a recognition or vision system is the feature extraction, the method used should have a provision for representing and manipulating the uncertainties. Fuzzy set theory provides a mechanism to represent and manipulate uncertainty and ambiguity. Fuzzy operators and their properties as well as fuzzy inference rules have found considerable applications in image analysis and pattern recognition, see by example [9, 15] and [17] and references therein.

In order to do that, the fuzzy mathematical morphology is an alternative extension of binary morphology to gray-scale morphology [22] using concepts and techniques from fuzzy set theory. Several researchers have introduced alternative morphological operators (see for example [4, 7, 10] and [5]). Bloch and Maître [7] follow an approach using t-norms and the associated model implication, with an involutive negation. Fuzzy set inclusion was used by Zadeh, Sinha and Dougherty, Kitainik and Bandler and Kohout in order to define fuzzy morphological operators. The *Minkowsky addition* was initially used by De Baets et al. in [4,5]. A detailed account can be found in [9, 15] and references therein. In this work we focus our attention in the general framework for fuzzy mathematical morphology constructed by De Baets in [10] where he uses *conjunctions* and *implications* in order to define *fuzzy erosion* and *fuzzy dilation*, without forcing duality relationships between these operators, and obtaining good properties for the corresponding fuzzy closing and fuzzy opening operators.

The most usual conjunctions used in order to define the fuzzy mathematical operators are t-norms and their residual implications. Recently, conjunctive uninorms (as another particular case of conjunctions) have also been used for the same purpose, see [6] and [12]. In particular, it is studied in [12] which conjunctive uninorms need to be chosen in order to preserve the algebraic and morphological properties needed to obtain a *good* mathematical morphology, going so far than De Baets in [6] including the representable and the idempotent uninorms.

Our goal is to extend the results presented in [12], in two directions. On one hand, we want to study some properties of fuzzy opening and fuzzy closing as well as open and closed fuzzy objects, when left-conjunctive uninorms are used. In particular, we mainly deal with the generalized idempotence property. On the other hand, we

present some experimental results based on these morphological operators, using two particular classes of left-continuous conjunctive uninorms.

The chapter[1] is organized as follows. In the next section we review the basic definitions and properties of fuzzy logical operators needed in the subsequent sections. In Section 8.3, we recall briefly some algebraic and morphological properties (see [12]), satisfied by the fuzzy morphological operators based on left-continuous conjunctive uninorms, to be used in the next section. In particular, we recall two classes of left-continuous conjunctive uninorms as the only ones (among representable and idempotent uninorms) for which duality between fuzzy morphological operators is satisfied. In Section 8.4, our goal is to study algebraic and morphological properties of *fuzzy closing* and *fuzzy opening*, closed and open fuzzy objects, when we use left-continuous conjunctive uninorms. Similar properties are obtained to those described by De Baets in [2] for some particular conjunctions and, in a most general context by Bodenhofer in [8]. In particular, in Section 8.5, we prove that when we take a *left-continuous uninorm* we obtain the so-called *generalized idempotence law* for the fuzzy closing and fuzzy opening. The results presented in this section generalize our results of [13] where we gave a generalized idempotence law using conjunctive representable uninorms, and also those presented by De Baets in [2] where he use continuous t-norms. Finally, in Section 8.6 we display some comparative experimental results using several left-continuous conjunctive uninorms in the two classes for with duality holds. The chapter ends with some conclusions and future work.

8.2 Fuzzy Logical Operators

Let us recall the fuzzy logical operators that we will use throughout the paper. More details on these operators can be found for instance in [16] and [19].

Definition 1. A decreasing and involutive unary operator $\mathcal{N}$ on $[0,1]$ with $\mathcal{N}(0)=1$ and $\mathcal{N}(1)=0$ is called a *strong negation*.

Definition 2. An increasing binary operator $\mathcal{C}$ on $[0,1]$ is called a *conjunction* if it satisfies

$$\mathcal{C}(0,1)=\mathcal{C}(1,0)=0 \quad \text{and} \quad \mathcal{C}(1,1)=1.$$

Definition 3. A binary operator $\mathcal{I}$ on $[0,1]$ is called an *implication* if it is decreasing with the first partial map, increasing with the second one, and it satisfies

$$\mathcal{I}(0,0)=\mathcal{I}(1,1)=1 \quad \text{and} \quad \mathcal{I}(1,0)=0.$$

[1] This work is an extended version of [14] presented in VipIMAGE 2007 Conference, held in Porto, Portugal, during the period 17–19 October 2007.

One can construct conjunctions and implications from each other. On one hand, given an implication $\mathcal{I}$ and a strong negation $\mathcal{N}$ the binary operator defined by

$$\mathcal{C}_{\mathcal{I},\mathcal{N}}(a,b) = \mathcal{N}(\mathcal{I}(a,\mathcal{N}(b)))$$

is a conjunction. On the other hand, given a conjunction $\mathcal{C}$ and a strong negation $\mathcal{N}$ the binary operator defined by

$$\mathcal{I}_{\mathcal{C},\mathcal{N}}(a,b) = \mathcal{N}(\mathcal{C}(a,\mathcal{N}(b)))$$

is an implication. Another way to construct implications from conjunctions is by residuation. Given a conjunction $\mathcal{C}$ the binary operator

$$\mathcal{I}_{\mathcal{C}}(a,b) = \sup\{c \in [0,1] \mid \mathcal{C}(a,c) \leq b\}$$

is an implication called the *r*esidual implication of $\mathcal{C}$.

A special kind of conjunctions are the well known t-norms. In fact, fuzzy morphological operators are usually constructed from t-norms and, a special kind of them, the nilpotent ones, has been proved to be the most useful in this framework, see for instance [18]. However, a generalization of t-norms appeared in [11], the uninorms.

Definition 4. A binary operator U on $[0,1]$ is called a *uninorm* if it is associative, commutative, increasing in each place and such that there exists some $e \in [0,1]$, called the *neutral element*, such that $U(e,x) = x$ for all $x \in [0,1]$.

It is clear that function U becomes a t-norm when $e = 1$ and a t-conorm when $e = 0$. For any uninorm we have $U(0,1) \in \{0,1\}$, if $U(1,0) = 0$, U is called *conjunctive* and if $U(1,0) = 1$ it is called *disjunctive*. Moreover, a uninorm U is called *idempotent* whenever $U(x,x) = x$ for all $x \in [0,1]$.

This kind of operators is specially interesting because of their behaviour: like a t-norm in $[0,e]^2$, and like a t-conorm in $[e,1]^2$. Note that conjunctive uninorms are particular cases of conjunctions and consequently they can be used in fuzzy mathematical morphology.

There are several known classes of conjunctive uninorms (see [1] and [11]). Some of them have been already used in fuzzy morphology in [6]. Since left-continuity is essential in order to have *good* properties, we will only use here representable (see Definition 5 below) and idempotent left-continuous, conjunctive uninorms. Of course, a fuzzy morphology can be done using also other kind of uninorms, not necessarily left-continuous, but all properties stated and proved in this paper where left-continuity is required can fail, for this kind of logical operators. Let us recall here the definitions and characterizations of representable and idempotent uninorms, but more details of these classes can be found in [11] and [1] respectively.

Definition 5. Let $e \in (0,1)$ and let $h : [0,1] \to [-\infty,+\infty]$ be a strictly increasing, continuous function with $h(0) = -\infty$, $h(e) = 0$ and $h(1) = +\infty$. The binary operator U defined by

$$U(a,b) = h^{-1}(h(a)+h(b))$$

for all $(a,b) \in [0,1]^2 \setminus \{(0,1),(1,0)\}$ and $U(0,1) = U(1,0) = 0$ is a conjunctive uninorm with neutral element e. This kind of uninorms are usually called representable uninorms and h is called its aditive generator.

The set of representable uninorms will be denoted by U_{rep} and a representable uninorm U, with additive generator h, and neutral element e, will be denoted by $U \equiv \langle h,e\rangle_{\text{rep}}$.

Theorem 1. *(See [21]) A uninorm U with neutral element $e \in (0,1)$ is representable if and only if it is continuous on $[0,1]^2 \setminus \{(0,1),(1,0)\}$.*

Observed that, since any conjunctive representable uninorm satisfies $U(x,0) = 0$ for all $x \in [0,1]$, from the thorem above it must be left-continuous.

Theorem 2. *(See [11]) A conjunctive uninorm U with neutral element $e \in (0,1)$ is representable if and only if it is strictly increasing and continuous on $(0,1)^2$ and there is a strong negation $\mathcal{N}$ with $\mathcal{N}(e) = e$, such that for any $(a,b) \in [0,1]^2 \setminus \{(0,1),(1,0)\}$*

$$U(a,b) = \mathcal{N}(U(\mathcal{N}(a),\mathcal{N}(b))).$$

Theorem 3. *(See [3]) Let U be a representable uninorm with additive generator h, then its residual implication I_U is given by*

$$I_U(x,y) = \begin{cases} h^{-1}(h(y)-h(x)) & \text{if } (x,y) \in [0,1]^2 \setminus \{(0,0),(1,1)\} \\ 1 & \text{otherwise} \end{cases}$$

Following with the idea to have the *good* properties derived from left-continuity, we will only use in this paper left-continuous, conjunctive, idempotent uninorms. However, note again that any other kind of conjunctive, idempotent uninorms (see [20]) can also be used in the same way.

Theorem 4. *(See [1]) A binary operator U is a left-continuous idempotent uninorm with neutral element $e \in (0,1)$ if and only if there exists a decreasing function $g : [0,1] \to [0,1]$ with fix point e, satisfying $g^2(x) = g(g(x)) \geq x$ for all $x \leq g(0)$ and $g(x) = 0$ for all $x > g(0)$ such that, for all $x,y \in [0,1]$, U is given by*

$$U(x,y) = \begin{cases} \min(x,y) & \text{if } y \leq g(x) \text{ and } x \leq g(0) \\ \max(x,y) & \text{elsewhere} \end{cases}$$

In view of the theorem above, any left-continuous idempotent uninorm U is uniquely determined by a decreasing function g, with we will call from now on the *associated function* of U. Moreover, any left-continuous idempotent uninorm U with neutral element e and associated function g will be denoted by $U \equiv \langle g,e\rangle_{\text{ide}}$.

Note that given any strong negation $\mathcal{N}$ we obtain a left-continuous idempotent uninorm just taking $g = \mathcal{N}$, that will be denoted by $U^{\mathcal{N}} \equiv \langle \mathcal{N},e\rangle_{\text{ide}}$.

On the other hand, a left-continuous idempotent uninorm $U \equiv \langle g, e \rangle_{\text{ide}}$ is conjunctive if and only if $g(0) = 1$ and this is also a necessary and sufficient condition for deriving residual implications from it. The residual implication of an idempotent uninorm is given by

Theorem 5. *(See [20]) Let U be any idempotent uninorm with $g(0) = 1$. The residual implication I_U is given by:*

$$I_U(x,y) = \begin{cases} \min(g(x), y) & \text{if } y < x \\ \max(g(x), y) & \text{if } y \geq x \end{cases}$$

Proposition 1. *(See [3]) Let U be a conjunctive uninorm and I_U its residual implication.*

- *The second partial map of I_U is right-continuous and for all $x, y \in [0,1]$,*

$$y \leq I_U(x, U(x,y)).$$

- *If U is left-continuous then so is the first partial map of I_U, I_U satisfies the exchange principle:*

$$I_U(x, I_U(y,z)) = I_U(y, I_U(x,z)),$$

and also the following properties:

$$U(x, I_U(x,y)) \leq y, \quad I_U(U(x,y), z) = I_U(x, I_U(y,z))$$

for all $x, y, z \in [0,1]$.

8.3 Fuzzy Morphological Operators

In the following sections, $\mathcal{I}$ will denote an implication, $\mathcal{C}$ a conjunction, $\mathcal{N}$ a strong negation, U a conjunctive uninorm with neutral element e, $\mathcal{I}_U$ its residual implication, A a gray-scale image, and B a gray-scale structuring element.

From the definition of classical erosion and dilation [22] it is clear that the intersection and inclusion of sets play a major role. The idea of De Baets [10] was to fuzzify the underlying logical operations, i.e. the Boolean conjunction and the Boolean implication, to obtain a successful fuzzification. An n-dimensional gray-scale image is modeled as an $\mathbb{R}^n \to [0,1]$ function. It is required that the gray values of the image belong to the real unit interval in order to consider an image as a fuzzy object. Thus, we have the following definitions.

Definition 6. The *fuzzy dilation* $D_{\mathcal{C}}(A,B)$ and *fuzzy erosion* $E_{\mathcal{I}}(A,B)$ of A by B are the gray-scale images defined by

$$D_{\mathcal{C}}(A,B)(y) = \sup_x \mathcal{C}(B(x-y),A(x))$$
$$E_{\mathcal{I}}(A,B)(y) = \inf_x \mathcal{I}(B(x-y),A(x)).$$

As in classical morphology, the difference between the fuzzy dilation and the fuzzy erosion of a gray-scale image, $D_U(A,B) \setminus E_{\mathcal{I}_U}(A,B)$, called the *fuzzy gradient* operator, can be used in edge detection.

Definition 7. The *fuzzy closing* $C_{\mathcal{C},\mathcal{I}}(A,B)$ and *fuzzy opening* $O_{\mathcal{C},\mathcal{I}}(A,B)$ of A by B are the gray-scale images defined by

$$C_{\mathcal{C},\mathcal{I}}(A,B)(y) = E_{\mathcal{I}}(D_{\mathcal{C}}(A,B),-B)(y) = \inf_x I(B(y-x), \sup_z C(B(z-x),A(z)))$$
$$O_{\mathcal{C},\mathcal{I}}(A,B)(y) = D_{\mathcal{C}}(E_{\mathcal{I}}(A,B),-B)(y) = \sup_x C(B(y-x), \inf_z I(B(z-x),A(z))).$$

Note that the reflection $-B$ of a n-dimensional fuzzy set B is defined by $-B(x) = B(-x)$, for all $x \in \mathbb{R}^n$.

Obviously, we can use conjunctive uninorms and related implications to define fuzzy morphological operators following the previous definitions. We investigate in [12] which conjunctive uninorms need to be chosen in order to preserve the algebraic and morphological properties satisfied by the classical morphological operators. Moreover, going farther than De Baets in [6], it is given in [12] sufficient and/or necessary conditions on the conjunctive uninorms in order to guarantee these properties.

Given a strong negation $\mathcal{N}$, we define by $(co_{\mathcal{N}}A)(x) = \mathcal{N}(A(x))$ the $\mathcal{N}$-complement $co_{\mathcal{N}}A$ of a fuzzy set A. Two fuzzy morphological operations P and Q are called $\mathcal{N}$-dual if for any two gray-scale objects A and B it holds that $P(A,B) = co_{\mathcal{N}}Q(co_{\mathcal{N}}A,B)$.

All results in this paper are concerning to a left-continuous conjunctive uninorm and its residual implication $\mathcal{I}_U$. However, it is known that the fuzzy dilation and fuzzy erosion are $\mathcal{N}$-dual if and only if $\mathcal{I} = \mathcal{I}_{\mathcal{C},\mathcal{N}}$ (or equivalently $\mathcal{C} = \mathcal{C}_{\mathcal{I},\mathcal{N}}$). Moreover, if the fuzzy dilation and fuzzy erosion are $\mathcal{N}$-dual, then also the fuzzy closing and fuzzy opening are $\mathcal{N}$-dual [10]. Hence, to have duality between our fuzzy morphological operators, we need to use conjunctive uninorms satisfying

$$\mathcal{I}_U = \mathcal{I}_{U,\mathcal{N}}.$$

Two special kinds of uninorms are the only ones, among representable and idempotent uninorms, that satisfy the previous property (see [3] and [20]).

Proposition 2. *The identity*

$$\mathcal{I}_U = \mathcal{I}_{U,\mathcal{N}}$$

is satisfied in each one of the following situations

(i) When $U \equiv \langle h,e\rangle_{\text{rep}}$ is a conjunctive representable uninorm and $\mathcal{N}$ is the strong negation obtained from the additive generator h of U by

$$\mathcal{N}(a) = h^{-1}(-h(a)).$$

(ii) When $\mathcal{N}$ is any strong negation and U is the corresponding conjunctive, left-continuous, idempotent uninorm $U^{\mathcal{N}} \equiv \langle \mathcal{N}, e \rangle_{\text{ide}}$.

Thus, these two kinds of conjunctive uninorms guarantee duality between fuzzy morphological operators. Consequently, they are the most suitable in our framework.

The algebraic properties of the fuzzy morphological operators defined from uninorms are studied in [18], and in [12]. We summarized in the following the algebraic properties of the fuzzy morphological operators needed in the next section. Recall that given two fuzzy objects A, B, the fuzzy inclusion $A \subseteq B$ means that $A(x) \leq B(x)$ for all x.

Proposition 3. *Let U be a left-continuous, conjunctive uninorm and I_U its residual implication. Let A_1 and A_2 be two gray-scale images and let B be a gray-scale structuring element. Then it holds:*

(a) E_{I_U}, D_U, $C_{U,\mathcal{I}_U}$ and $O_{U,\mathcal{I}_U}$ are increasing in the first place.
(b) Moreover, they satisfy:

$$\begin{aligned} E_{I_U}(A_1 \cap A_2, B) &= E_{I_U}(A_1, B) \cap E_{I_U}(A_2, B) \\ E_{I_U}(A_1 \cup A_2, B) &\supseteq E_{I_U}(A_1, B) \cup E_{I_U}(A_2, B) \\ D_U(A_1 \cup A_2, B) &= D_U(A_1, B) \cup D_U(A_2, B) \\ D_U(A_1 \cap A_2, B) &\subseteq D_U(A_1, B) \cap D_U(A_2, B). \end{aligned}$$

The extensivity of the fuzzy dilation and the anti-extensivity of the fuzzy morphological operators are ensured by next propositions.

Proposition 4. *Let U be a conjunctive uninorm with neutral element $e \in (0,1)$, let I_U be its residual implication and let B be a gray-scale structuring element such that $B(0) = e$. Then the following inclusions hold:*

$$E_{\mathcal{I}_U}(A, B) \subseteq A \subseteq D_U(A, B).$$

Proposition 5. *Let U be a left-continuous conjunctive uninorm and I_U its residual implication, let A be a gray-scale image and let B be a gray-scale structuring element, then it holds*

1. The fuzzy closing $C_{U,\mathcal{I}_U}$ is extensive and the fuzzy opening is anti-extensive:

$$O_{U,\mathcal{I}_U}(A, B) \subseteq A \subseteq C_{U,\mathcal{I}_U}(A, B).$$

2. The fuzzy closing and the fuzzy opening are idempotent, i.e.:

$$\begin{aligned} C_{U,\mathcal{I}_U}(C_{U,\mathcal{I}_U}(A, B), B) &= C_{U,\mathcal{I}_U}(A, B), \\ O_{U,\mathcal{I}_U}(O_{U,\mathcal{I}_U}(A, B), B) &= O_{U,\mathcal{I}_U}(A, B). \end{aligned}$$

8.4 Closed and Open Fuzzy Objects

The idempotence of the fuzzy closing and opening, when we get U a left-continuous conjunctive uninorm and I_U its residual implication, motivates as in the classical mathematical morphology, the following definitions.

Definition 8. Let A and B be two gray-scale images. We say that A is B-closed (resp. B-open) if $C_{U,\mathcal{I}_U}(A,B) = A$ (resp. $O_{U,\mathcal{I}_U}(A,B) = A$).

Observe that, using Prop. 5, it is clear that $C_{U,\mathcal{I}_U}(A,B)$ is B-closed and $O_{U,\mathcal{I}_U}(A,B)$ is B-open. Moreover, all B-open and B-closed objects are the opening and the closing of some image, respectively, as we can see in the following proposition that was advanced in [2], without proof. We include it for the sake of clarity.

Proposition 6. *If U is left-continuous, then it holds:*

(a) A is B-open if and only if there exists a fuzzy object F such that $A = D_U(F,-B)$.
(b) A is B-closed if and only if there exists a fuzzy object F such that $A = E_{\mathcal{I}_U}(F,-B)$.

Proof. Let us assume that A is B-open. By definition of fuzzy opening, choosing $F = E_{\mathcal{I}_U}(A,B)$ we have a fuzzy object satisfying that $D_U(F,-B) = A$. Now suppose that A can be represented as $A = D_U(F,-B)$ for some fuzzy object F. From Prop. 5 we know that $O_{U,\mathcal{I}_U}(A,B) \subseteq A$. In order to prove the other inclusion, using Prop. 1 and that $\mathcal{I}_U$ is increasing in the second partial map, we have:

$$\begin{aligned} &E_{I_U}(A,B)(y) = \inf_x I_U(B(x-y),A(x)) = \inf_x I_U(B(x-y),D_U(F,-B)(x)) \\ &= \inf_x I_U(B(x-y),\sup_z U(B(x-z),F(z))) \geq \inf_x I_U(B(x-y),U(B(x-y),F(y))) \\ &\geq F(y). \end{aligned}$$

So, we have shown that $F \subseteq E_{I_U}(A,B)$. Then, by Prop. 3 we have that $A \subseteq O_{U,\mathcal{I}_U}(A,B)$. A similar argument proves b). □

As it was pointed out in [8] opening and closing operators only make sense if the opening always gives an open result, and the closing operator gives a closed result. Moreover, it is desirable to have *extremal properties*. We see now that this last requirement is also satisfied by our opening and closing fuzzy operators.

Proposition 7. *If U is left-continuous, then it holds:*

(a) $O_{U,\mathcal{I}_U}(A,B)$ is the largest B-open fuzzy subset of A.
(b) $C_{U,\mathcal{I}_U}(A,B)$ is the smallest B-closed fuzzy superset of A.

Proof. We only proof (a) since (b) follows similarly.

We know that $O_{U,\mathcal{I}_U}(A,B)$ is B-open and that $O_{U,\mathcal{I}_U}(A,B) \subseteq A$ from Prop. 5. Now let us assume that $E \subseteq A$ and E is B-open. Then, as E is B-open, we have $E = O_{U,\mathcal{I}_U}(E,B)$. By Prop. 3 $O_{U,\mathcal{I}_U}$ is increasing in the first argument, then $E = O_{U,\mathcal{I}_U}(E,B) \subseteq O_{U,\mathcal{I}_U}(A,B)$. □

Now let us consider the preservation of B-openness and B-closedness by intersections and unions, respectively.

Proposition 8. *Consider U a left-continuous uninorm, and A_1 and A_2 two gray-levels images. Then, it holds:*

(a) If A_1 and A_2 are both B-open then, $A_1 \cup A_2$ is B-open.
(b) If A_1 and A_2 are both B-closed then, $A_1 \cap A_2$ is B-closed.

Proof. (a) If A_1 and A_2 are B-open then, $O_{U,\mathcal{I}_U}(A_1,B) = A_1$ and $O_{U,\mathcal{I}_U}(A_2,B) = A_2$. By Prop. 3 we have that

$$\begin{aligned} A_1 \cup A_2 &= O_{U,\mathcal{I}_U}(A_1,B) \cup O_{U,\mathcal{I}_U}(A_2,B) \\ &= D_U(E_{\mathcal{I}_U}(A_1,B),-B) \cup D_U(E_{\mathcal{I}_U}(A_2,B),-B) \\ &= D_U(E_{\mathcal{I}_U}(A_1,B) \cup E_{\mathcal{I}_U}(A_2,B),-B) \\ &\subseteq D_U(E_{\mathcal{I}_U}(A_1 \cup A_2,B),-B) = O_{U,\mathcal{I}_U}(A_1 \cup A_2,B), \end{aligned}$$

thus $A_1 \cup A_2 \subseteq O_{U,\mathcal{I}_U}(A_1 \cup A_2,B)$. The other inclusion is a consequence of Prop. 5. Therefore, $A_1 \cup A_2 = O_{U,\mathcal{I}_U}(A_1 \cup A_2,B)$ and $A_1 \cup A_2$ is a B-open fuzzy set.
(b) The proof is similar. □

Previous propositions are valid for any left-continuous conjunctive uninorm. However, if we want to have duality between closed and open fuzzy objects we need again the two kinds of uninorms stated in Prop. 2.

Proposition 9. *Consider U satisfying the condition (i) or (ii) from Prop. 2. Then, A is B-open if and only if $co_{\mathcal{N}}A$ is B-closed.*

Proof. We will prove the implication from right to left. Assume that $co_{\mathcal{N}}A$ is B-closed, $C_{U,\mathcal{I}_U}(co_{\mathcal{N}}A,B) = co_{\mathcal{N}}A$. Complementing both sides, since $\mathcal{N}$ is an involutive operator, and closing and opening are $\mathcal{N}$-dual, then we obtain

$$O_{U,\mathcal{I}_U}(A,B) = co_{\mathcal{N}}C_{U,\mathcal{I}_U}(co_{\mathcal{N}}A,B) = A$$

thus A is B-open. The converse holds similarly □

8.5 Generalized Idempotence Law for Fuzzy Closing and Opening Using Left-Continuous Uninorms

Following with the study of the properties that meet the morphological operators, one of the important properties is the so-called generalized idempotence, which is satisfied by the opening and closing in the classical case. This section will show that, using left-continuous uninorms like in Prop. 2, the generalized idempotence is also satisfied by the fuzzy closing and fuzzy opening. First we need several previous results.

Proposition 10. *Let U be a uninorm with neutral element $e \in [0,1]$. The following statements are equivalent*

(i) U is left-continuous;
(ii) For all $(a,b) \in [0,1]^2$, $U(a,\mathcal{I}_U(a,b)) \leq b$;
(iii) For all $(a,b,c) \in [0,1]^3$, $U(\mathcal{I}_U(a,b),\mathcal{I}_U(b,c)) \leq \mathcal{I}_U(a,c)$.

Proof. The equivalence (i)$\Leftrightarrow$ (ii) is similar to a well known result for t-norms, see for instance [3].

To prove (iii)$\Rightarrow$ (ii), put (e,a,b) in (iii).

To show (ii)$\Rightarrow$ (iii), by definition of $\mathcal{I}_U$, we only need to prove that for all a,b, $c \in [0,1]$

$$U(a,U(\mathcal{I}_U(a,b),\mathcal{I}_U(b,c))) \leq c.$$

Now, using the associativity and the monotonicity of U and (ii), we have:

$$U(a,U(\mathcal{I}_U(a,b),\mathcal{I}_U(b,c))) = U(U(a,\mathcal{I}_U(a,b)),\mathcal{I}_U(b,c))) \leq U(b,\mathcal{I}_U(b,c)) \leq c.$$

□

Proposition 11. *Let U be a left-continuous uninorm. Then U and $\mathcal{I}_U$ satisfy*

$$\mathcal{I}_U(x,U(y,z)) \geq U(\mathcal{I}_U(x,y),z)$$

for all $x,y,z \in [0,1]$.

Proof. If we prove

$$U(x,U(\mathcal{I}_U(x,y),z)) \leq U(y,z)$$

for all $x,y,z \in [0,1]$, then the result follows.

Using (ii) from the previous proposition, we have that for all $x,y,z \in [0,1]$

$$U(x,U(\mathcal{I}_U(x,y),z)) = U(U(x,\mathcal{I}_U(x,y)),z) \leq U(y,z)$$

□

Proposition 12. *Let U be a left-continuous uninorm. For any $a,b,c,d,e,f \in [0,1]$, if $U(a,\mathcal{I}_U(b,c)) \geq d$, and $U(e,\mathcal{I}_U(f,b)) \geq a$ then*

$$U(e,\mathcal{I}_U(f,c)) \geq d.$$

Proof. Using Prop. 10, we have for all $a,b,c,d,e,f \in [0,1]$:

$$\begin{aligned} d &\leq U(a,\mathcal{I}_U(b,c)) \leq U(U(e,\mathcal{I}_U(f,b)),\mathcal{I}_U(b,c)) \\ &= U(e,U(\mathcal{I}_U(f,b),\mathcal{I}_U(b,c))) \leq U(e,\mathcal{I}_U(f,c)). \end{aligned}$$

□

Proposition 13. *Let U be a left-continuous uninorm, and $\mathcal{I}_U$ its residual implication. If for any real numbers $a,b,c,d,e,f,g,h \in [0,1]$, it is satisfied $U(a,\mathcal{I}_U(b,c)) \geq d$, $U(c,\mathcal{I}_U(e,f)) \leq g$ and $U(d,\mathcal{I}_U(e,f)) \geq h$, then*

$$U(a,\mathcal{I}_U(b,g)) \geq h.$$

Proof. Using Prop. 10 and 11, we have for all $a,b,c,d,e,f,g,h \in [0,1]$:

$$\begin{aligned} h &\leq U(d,\mathcal{I}_U(e,f)) \leq U(U(a,\mathcal{I}_U(b,c)),\mathcal{I}_U(e,f)) = U(a,U(\mathcal{I}_U(b,c),\mathcal{I}_U(e,f))) \\ &\leq U(a,\mathcal{I}_U(b,U(c,\mathcal{I}_U(e,f)))) \leq U(a,\mathcal{I}_U(b,g)). \end{aligned}$$

□

Now, using left-continuous uninorms like in Prop. 2, and taking into account the inequalities satisfied by uninorms and their residual implications, we obtain the so-called generalized idempotence laws for fuzzy closing and fuzzy opening. The proof of inclusions concerning fuzzy opening are quite similar to those given by De Baets in [2] for continuous t-norms, using in our case Props. 12 and 13. With respect to inclusions concerning fuzzy closing, their proofs follow from duality, guaranteed by Prop. 2.

Proposition 14. *Consider U satisfying the condition (i) or (ii) from Prop. 2. If A is B-open and rang(A) and rang(B) are finite sets, then for any fuzzy object F it holds:*

$$O_{U,\mathcal{I}_U}(F,A) \subseteq O_{U,\mathcal{I}_U}(F,B) \subseteq F$$

and dually, $F \subseteq C_{U,\mathcal{I}_U}(F,B) \subseteq C_{U,\mathcal{I}_U}(F,A)$.

Proposition 15. *(Generalized Idempotence) Consider U satisfying the condition (i) or (ii) from Prop. 2 and $\mathcal{I}_U$ its residual implication. If A is B-open and rang(A) and rang(B) are finite sets, then for any fuzzy object F it holds:*

$$O_{U,\mathcal{I}_U}(O_{U,\mathcal{I}_U}(F,B),A) = O_{U,\mathcal{I}_U}(O_{U,\mathcal{I}_U}(F,A),B) = O_{U,\mathcal{I}_U}(F,A)$$

and dually for the fuzzy closing.

Remark 1. The previous propositions are valid for any left-continuous uninorm, and in particular, for any left-continuous t-norm, improving the results in [2].

In the next we try to omit, in the previous propositions (see Prop. 14), the condition of finite rang for the sets A and B when we work with continuous t-norms. Following a procedure similar to that used by De Baets in [2], but different in substance, we can generalize the results by eliminating this condition.

Proposition 16. *Let U be a continuous t-norm and let $\mathcal{I}_U$ be its residual implication. If A is B-open, for any fuzzy object F it holds:*

$$O_{U,\mathcal{I}_U}(F,A) \subseteq O_{U,\mathcal{I}_U}(F,B) \subseteq F$$

and dually $F \subseteq C_{U,\mathcal{I}_U}(F,B) \subseteq C_{U,\mathcal{I}_U}(F,A)$.

Proof. Since A is B-open then for all $y \in [0,1]$ we have that

$$A(y) = O_{U,\mathcal{I}_U}(A,B)(y) = \sup_x U\left(B(y-x), \inf_z \mathcal{I}_U(B(z-x),A(z))\right)$$
$$= \sup_x \inf_z U\left(B(y-x), \mathcal{I}_U(B(z-x),A(z))\right).$$

This implies that given $\varepsilon > 0$, there exists x_ε such that

$$\inf_z U\left(B(y-x_\varepsilon), \mathcal{I}_U(B(z-x_\varepsilon),A(z))\right) > A(y) - \varepsilon.$$

Therefore, there exists x_ε satisfying

$$U\left(B(y-x_\varepsilon), \mathcal{I}_U(B(z-x_\varepsilon),A(z))\right) > A(y) - \varepsilon \quad \text{for all } z. \tag{8.1}$$

We want to see that $O_{U,\mathcal{I}_U}(F,A) \subseteq O_{U,\mathcal{I}_U}(F,B)$, it will be enough to see that

"for all $\alpha \in]0,1]$, if $O_{U,\mathcal{I}_U}(F,A)(y) = \alpha$ then $O_{U,\mathcal{I}_U}(F,B)(y) \geq \alpha$"

Let $\alpha \in]0,1]$, arbitrary but fixed and let y be such that

$$O_{U,\mathcal{I}_U}(F,A)(y) = \alpha.$$

In a similar way as we obtained (8.1) we have that there exists x'_ε such that

$$U\left(A(y-x'_\varepsilon), \mathcal{I}_U(A(z-x'_\varepsilon),F(z))\right) > \alpha - \varepsilon \quad \text{for all } z. \tag{8.2}$$

We will now show that

$$O_{U,\mathcal{I}_U}(F,B)(y) = \sup_x U\left(B(y-x), \inf_z \mathcal{I}_U(B(z-x),F(z))\right) \geq \alpha - \varepsilon.$$

In order to do it, it suffices to show that there exists an $\bar{x}$ such that

$$U\left(B(y-\bar{x}), \inf_z \mathcal{I}_U(B(z-\bar{x}),F(z))\right) \geq \alpha - \varepsilon$$

or simply,

$$U\left(B(y-\bar{x}), \mathcal{I}_U(B(z-\bar{x}),F(z))\right) > \alpha - \varepsilon \quad \text{for all } z.$$

Applying (8.2) and taking into account that U is left-continuous, given $\varepsilon > 0$, there exists $\delta > 0$ such that

$$U\left(A(y-x'_\varepsilon) - \delta, \mathcal{I}_U(A(z-x'_\varepsilon),F(z))\right) \geq \alpha - \varepsilon \quad \text{for all } z. \tag{8.3}$$

Now, applying (8.1) to $y - x'_\varepsilon$ and $\delta > 0$ we obtain an x_δ such that

$$U\left(B(y-x'_\varepsilon-x_\delta), \mathcal{I}_U(B(z-x_\delta),A(z))\right) > A(y-x'_\varepsilon) - \delta \quad \text{for all } z. \tag{8.4}$$

Applying (8.4) to $z - x'_\varepsilon$, we obtain

$$U\left(B(y-x'_\varepsilon-x_\delta),\mathcal{I}_U(B(z-x'_\varepsilon-x_\delta),A(z-x'_\varepsilon))\right) > A(y-x'_\varepsilon)-\delta. \tag{8.5}$$

Proposition 12 applied to (8.3) and (8.5) implies

$$U\left(B(y-x'_\varepsilon-x_\delta),\mathcal{I}_U(B(z-x'_\varepsilon-x_\delta),F(z))\right) \geq \alpha-\varepsilon \quad \text{for all } z. \tag{8.6}$$

Note that (8.6) is verfied for all z and then it ensures the existence of an $\bar{x}$ such that

$$U\left(B(y-\bar{x}),\mathcal{I}_U(B(z-\bar{x}),F(z))\right) \geq \alpha-\varepsilon \quad \text{for all } z$$

from which we obtain

$$U\left(B(y-\bar{x}),\inf_z \mathcal{I}_U(B(z-\bar{x}),F(z))\right) \geq \alpha-\varepsilon$$

and

$$\sup_x U\left(B(y-x),\inf_z \mathcal{I}_U(B(z-x),F(z))\right) \geq \alpha-\varepsilon.$$

That is

$$O_{U,\mathcal{I}_U}(F,B)(y) \geq \alpha-\varepsilon \quad (\text{for all } \varepsilon>0)$$

or equivalently

$$O_{U,\mathcal{I}_U}(F,B)(y) \geq \alpha.$$

This concludes the proof for the opening part. The closing part follows by duality. □

Thanks to the above proposition we can enunciate the *generalized idempotence* for continuos t-norms avoiding the restrictive condition of finite rang, that doesn't appear in the *crisp case.*

Theorem 6. *Let U be a continuous t-norm and let $\mathcal{I}_U$ be its residual implication. If A is B-open, for any fuzzy object F it holds:*

$$O_{U,\mathcal{I}_U}(O_{U,\mathcal{I}_U}(F,B),A) = O_{U,\mathcal{I}_U}(O_{U,\mathcal{I}_U}(F,A),B) = O_{U,\mathcal{I}_U}(F,A)$$

and dually

$$C_{U,\mathcal{I}_U}(C_{U,\mathcal{I}_U}(F,B),A) = C_{U,\mathcal{I}_U}(C_{U,\mathcal{I}_U}(F,A),B) = C_{U,\mathcal{I}_U}(F,A).$$

Proof. The proof of this theorem is identical to that carried out in the *crisp case*, but we include it for the sake of clarity. Let F be a fuzzy object, Prop. 5 ensures that $O_{U,\mathcal{I}_U}(F,B) \subseteq F$. Taking into account that the opening is increasing in its first argument (see Prop. 3), we have

$$O_{U,\mathcal{I}_U}(O_{U,\mathcal{I}_U}(F,B),A) \subseteq O_{U,\mathcal{I}_U}(F,A).$$

As A is B-open, Prop. 16 yields $O_{U,\mathcal{I}_U}(F,A) \subseteq O_{U,\mathcal{I}_U}(F,B)$. The idempotence of the fuzzy opening, done by the Prop. 5, ensures that

$$O_{U,\mathcal{I}_U}(F,A) = O_{U,\mathcal{I}_U}(O_{U,\mathcal{I}_U}(F,A),A) \subseteq O_{U,\mathcal{I}_U}(O_{U,\mathcal{I}_U}(F,B),A).$$

From both inclusions we obtain the equality. This concludes the proof. □

8.6 Experimental Results

In this section we present some experiments showing the differences between basic fuzzy morphological operators using different uninorms. The examples presented in this section illustrate the influence of the choice of the pair $(U,\mathcal{I}_U)$ using both, idempotent and representable conjunctive uninorms as in Prop. 2.

One of the first applications that we have implemented is the *fuzzy morphological gradient*

$$D_U(A,B) \setminus E_{\mathcal{I}_U}(A,B)$$

which is a useful tool for edge detection. Indeed, from Prop. 4 if $B(0) = e$, then it holds $E_{\mathcal{I}_U}(A,B) \subseteq A \subseteq D_U(A,B)$ and therefore the *fuzzy morphological gradient* will serve as edge-image of the fuzzy object A.

Some of our input images, A, are depicted in Fig. 8.1. We will present the results following the approach of Nachtegael and Kerre in [18]. The structuring element B used for the fuzzy operators is represented by the matrix

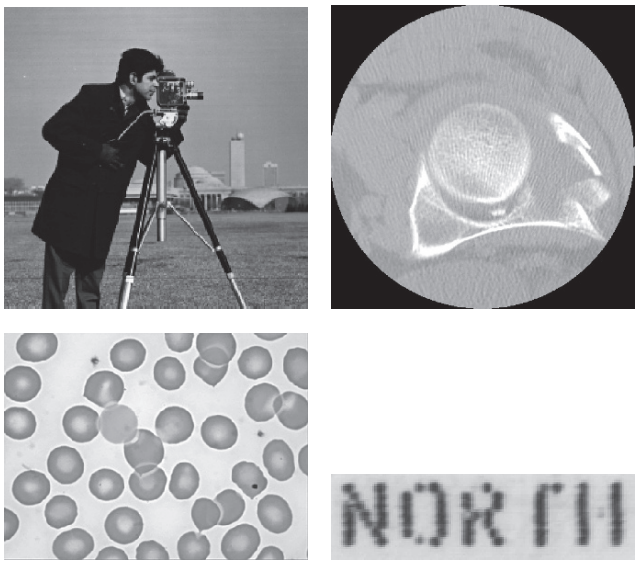


Fig. 8.1 Input images used in the experiments

$$B = e \cdot \begin{pmatrix} 0.86 & 0.86 & 0.86 \\ 0.86 & 1.00 & 0.86 \\ 0.86 & 0.86 & 0.86 \end{pmatrix} \tag{8.7}$$

where e is the neutral element of the uninorm. Following the definitions given in [18], we compare the edge images with those obtained using the classical gray-scale morphology based on the umbra approach, and those obtained by the fuzzy approach based on the Łuckasiewicz t-norm, $T_L(x,y) = \max\{0, x+y-1\}$ for all $x, y \in [0,1]$.

In the next figures, Figs. 8.2–8.5, we show the fuzzy gradient operator corresponding to each one of the input images displayed in Fig. 8.1, using several left-continuous representable and idempotent uninorms, compared with the fuzzy gradient using the pair $(T_L, \mathcal{I}_{T_L})$, the Łuckasiewicz t-norm and its residual implication, and the umbra approach, with the same structuring element but taking $e = 1$ in (8.7). Recall that the pair $(T_L, \mathcal{I}_{T_L})$ is the representative of the only class of t-norms (nilpotent ones) that guarantees the fulfillment of all the properties in order to have a good fuzzy mathematical morphology, including duality [18].

In Fig. 8.2 we display some of the results obtained using the classical cameraman image. While the hard edges (see the person in the foreground and some buildings in the background) are detected very well in all the cases, it can be observed that the gradient obtained using conjunctive uninorms (top) detect some soft edges better than the gradient obtained with $(T_L, \mathcal{I}_{T_L})$ and the umbra approach (bottom) (see some of the buildings in the background).

Fig. 8.2 Top left: using $U^{\mathcal{N}} \equiv \langle \sqrt{1-x^2}, \frac{\sqrt{2}}{2} \rangle_{\text{ide}}$. Top right: using $U \equiv \langle \ln\left(\frac{x}{1-x}\right), 0.5 \rangle_{\text{rep}}$. Down left: using $(T_L, \mathcal{I}_{T_L})$. Down right: umbra approach

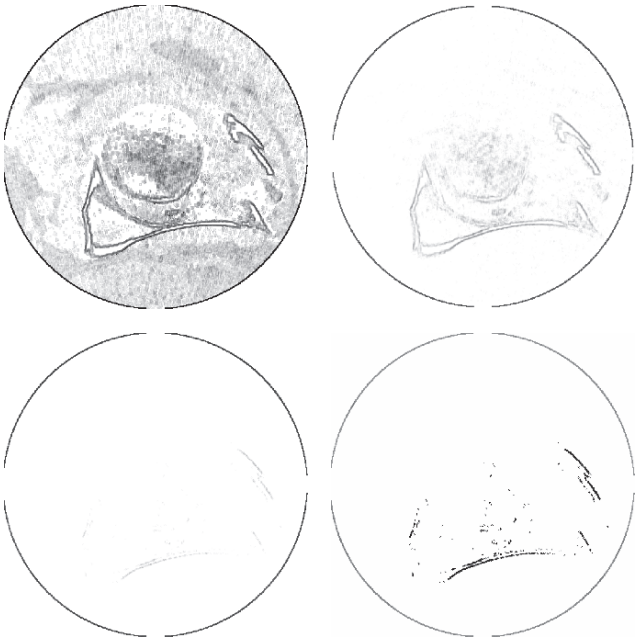


Fig. 8.3 Top left: using $U^{\mathcal{N}} \equiv \langle \sqrt{1-x^2}, \frac{\sqrt{2}}{2} \rangle_{\text{ide}}$. Top right: using $U \equiv \langle \ln\left(\frac{x}{1-x}\right), 0.5 \rangle_{\text{rep}}$. Down left: using $(T_L, \mathcal{I}_{T_L})$. Down right: umbra approach

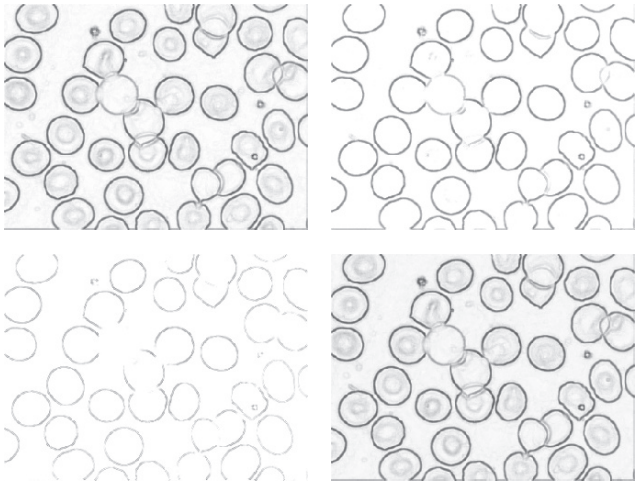


Fig. 8.4 Top left: using $U^{\mathcal{N}} \equiv \langle 1-x, 0.5 \rangle_{\text{ide}}$. Top right: using $U \equiv \langle \ln\left(\frac{x}{1-x}\right), 0.5 \rangle_{\text{rep}}$. Down left: using $(T_L, \mathcal{I}_{T_L})$. Down right: umbra approach

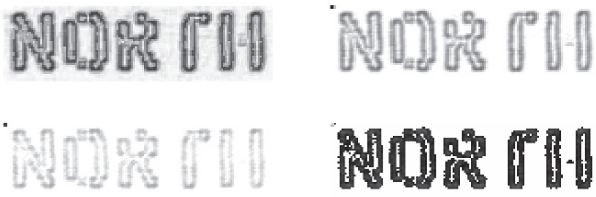


Fig. 8.5 Top left: using $U^{\mathcal{N}} \equiv \langle 1-x, 0.5\rangle_{\text{ide}}$. Top right: using $U \equiv \langle \ln\left(\frac{x}{1-x}\right), 0.5\rangle_{\text{rep}}$ Down left: using $(T_L, \mathcal{I}_{T_L})$. Down right: umbra approach

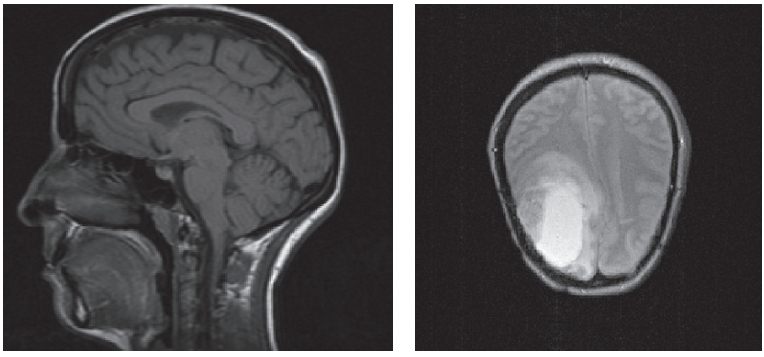


Fig. 8.6 MR images used as input images in the experiments

Figure 8.3 shows the edges obtained for the hip image displayed in Fig. 8.1 (top-right). In case of uninorms (top), we can observe how the bone structure and a little fracture is preserved. Otherwise, in the umbra and t-norms approaches, these structures are lost. So, we cannot use umbra approach and t-norms if we want to make a subsequent analysis.

The edge-images displayed in Fig. 8.4 correspond to the red bloods cells image shown in Fig. 8.1, also known as erythrocytes image. The edge-images obtained using idempotent and representable uninorms (top) improve the results corresponding to t-norms and are quite similar to umbra approach results (bottom). Observe that the boundary and structure of some cells disappear when we use t-norms, but in the boundary images obtained with uninorms these structures are preserved.

Figure 8.5 is an artificial image. We can observe, again, that the results based on uninorms improve the results based on t-norms. In the last case, we can observe a very soft boundary, although it is darker in the umbra approach.

In Fig. 8.6 we show two slices of different MRI images. On the left we display a saggital slice of an MRI and, on the right, we display an axial slice. In this last image the existence of a pathology can be observed. The gradients obtained from these two images are displayed in Figs. 8.7 and 8.8 respectively. As we can see, using uninorms (top) we obtain better results than using the Łuckasiewicz t-norm (bottom left). Note also that, in the case of idempotent uninorm (top left) the shape of the pathology and the brain are better delimited than in the umbra approach case.

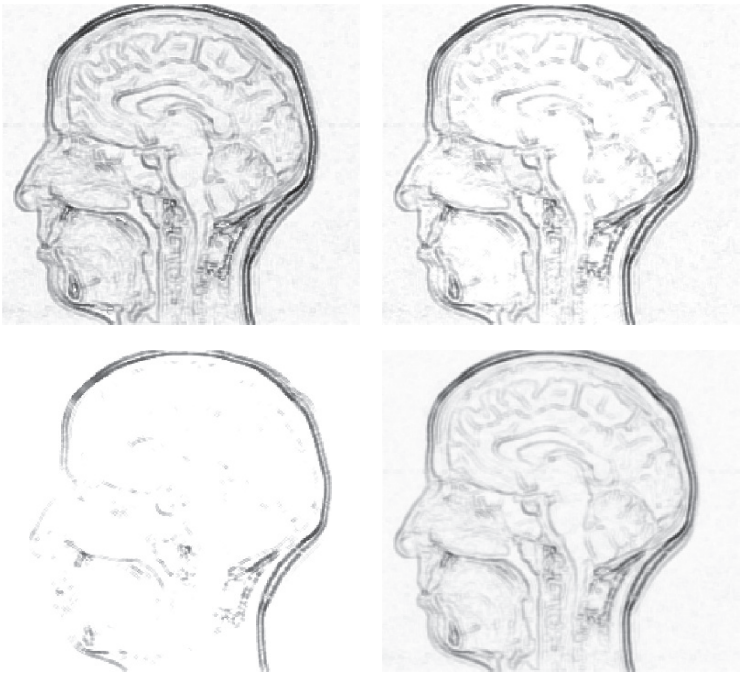


Fig. 8.7 Top left: using $U^{\mathcal{N}} \equiv \langle 1-x, 0.5\rangle_{\text{ide}}$. Top right: using $U \equiv \langle \ln\left(-\frac{1}{\ln 2}\cdot \ln(1-x)\right), 0.5\rangle_{\text{rep}}$. Down left: using $(T_L, \mathcal{I}_{T_L})$. Down right: umbra approach

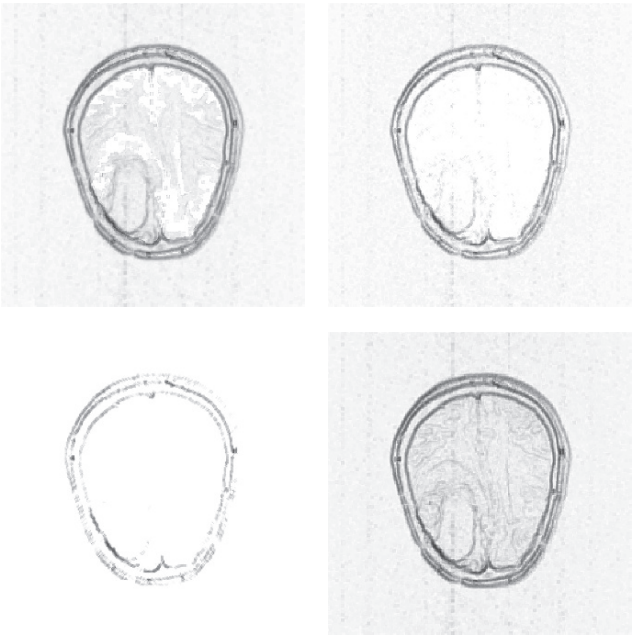


Fig. 8.8 Top left: using $U^{\mathcal{N}} \equiv \langle 1-x, 0.5\rangle_{\text{ide}}$. Top right: using $U \equiv \langle \ln\left(\frac{x}{1-x}\right), 0.5\rangle_{\text{rep}}$. Down left: using $(T_L, \mathcal{I}_{T_L})$. Down right: umbra approach

8.7 Conclusions and Future Work

We have proposed a fuzzy mathematical morphology based on left-continuous conjunctive uninorms, that fulfills the same good morphological properties that the one based on nilpotent t-norms. Some of this properties, known for continuous t-norms are generalized to left-continuous uninorms.

Respect to the experimental results, it is important to note that, in all cases, the results obtained in edge-detection with the approach presented here, are equal to or better than the ones obtained with the umbra approach and the nilpotent t-norm fuzzy approach, because they detect some soft edges more accurately. Depending of the application, this feature can be either an advantage or a drawback. On the other hand, note that in general, the edge-image with uninorms contains the same (or more) information as the edge image derived from the umbra approach. Note also that, from a practical point of view, the choice of the uninorm and its residual implication will also depend upon the specific problem or application. The same remark holds for the choice of the structuring element. The size and the isotropy of the structuring element is another think to take into account in practical applications. We are adressing currently some experiments in this aspect. We are also implementing the opening and closing operators, and it is our intention to use them for defining top-hat transformations and fuzzy morphological filters, in order to follow with our comparative study with the umbra and t-norms approaches.

Acknowledgements This work has been supported by the project MTM2006-05540, of the Spanish Government.

References

1. B. De Baets. Idempotent uninorms. *European Journal of Operational Research*, 118:631–642, 1999.
2. B. De Baets. Generalized idempotence in fuzzy mathematical morphology. In Etienne E. Kerre and Mike Nachtegael, editors, *Fuzzy Techniques in Image Processing*, number 52 in Studies in Fuzziness and Soft Computing, chapter 2, pages 58–75. Physica-Verlag, New York, 2000.
3. B. De Baets and J.C. Fodor. Residual operators of uninorms. *Soft Computing*, 3:89–100, 1999.
4. B. De Baets, E.E. Kerre, and M. Gupta. The fundamentals of fuzzy mathematical morfologies part i: basics concepts. *International Journal of General Systems*, 23:155–171, 1995.
5. B. De Baets, E.E. Kerre, and M. Gupta. The fundamentals of fuzzy mathematical morfologies part ii: idempotence, convexity and decomposition. *International Journal of General Systems*, 23:307–322, 1995.
6. B. De Baets, N. Kwasnikowska, and E. Kerre. Fuzzy morphology based on uninorms. In *Proceedings of the Seventh IFSA World Congress*, pages 215–220, Prague, 1997.
7. I. Bloch and H. Maître. Fuzzy mathematical morphologies: a comparative study. *Pattern Recognition*, 28:1341–1387, 1995.
8. U. Bodenhofer. A unified framework of opening and closure operators with respect to arbitrary fuzzy relations. *Soft Computing*, 7:220–227, 2003.
9. H.D. Cheng, X.H. Jiang, and J. Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34:2259–2281, 2000.

10. B. De Baets. Fuzzy morphology: a logical approach. In B. M. Ayyub and M. M. Gupta, editors, *Uncertainty Analysis in Engineering and Science: Fuzzy Logic, Statistics, and Neural Network Approach*, pages 53–68. Kluwer Norwell, 1997.
11. J.C. Fodor, R.R. Yager, and A. Rybalov. Structure of uninorms. *International Journal of Uncertainty, Fuzziness, Knowledge-Based Systems*, 5:411–427, 1997.
12. M. González, D. Ruiz-Aguilera, and J. Torrens. Algebraic properties of fuzzy morphological operators based on uninorms. In *Artificial Intelligence Research and Development*, volume 100 of *Frontiers in Artificial Intelligence and Applications*, pages 27–38, IOS Press. Amsterdam, 2003.
13. M. González, D. Ruiz-Aguilera, and J. Torrens. Opening and closing operators in fuzzy morphology using conjunctive uninorms. In *IPMU'2004, Tenth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 13–14, Perugia, Italy, 2004. (Extended version: Technical Report DMI-UIB A-01-2004).
14. M. González-Hidalgo, A. Mir-Torres, D. Ruiz-Aguilera, and J. Torrens. Fuzzy morphology based on uninorms: Image edge-detection. opening and closing. In João Tavares and Natal Jorge, editors, *Computational Vision and Medical Image Processing*, pages 127–133. Taylor & Francis, London, 2008.
15. E.E. Kerre and M. Nachtegael. *Fuzzy Techniques in Image Processing*, volume 52 of *Studies in Fuzziness and Soft Computing*. Springer, New York, 2000.
16. E.P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer, London, 2000.
17. M. Nachtegael, D. Van der Weken, D. Van De Ville, and E.E. Kerre, editors. *Fuzzy Filters for Image Processing*, volume 122 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag, 2003.
18. M. Nachtegael and E.E. Kerre. Classical and fuzzy approaches towards mathematical morphology. In Etienne E. Kerre and Mike Nachtegael, editors, *Fuzzy Techniques in Image Processing*, number 52 in Studies in Fuzziness and Soft Computing, chapter 1, pages 3–57. Physica-Verlag, New York, 2000.
19. D. Ruiz-Aguilera. *Contribució a l'estudi de les uninormes en el marc de les equacions funcionals. Aplicacions a la morfologia matemàtica*. PhD thesis, University of Balearic Islands, 2007.
20. D. Ruiz-Aguilera and J. Torrens. Residual implications and co-implications from idempotent uninorms. *Kybernetika*, 40:21–38, 2004.
21. D. Ruiz-Aguilera and J. Torrens. Distributivity and conditional distributivity of a uninorm and a continuous t-conorm. *IEEE Transactions on Fuzzy Systems*, 14:180–190, 2006.
22. J. Serra. *Image Analysis and Mathematical Morphology, Vols. 1, 2*. Academic, London, 1982, 1988.

Chapter 9
A Tissue Relevance and Meshing Method for Computing Patient-Specific Anatomical Models in Endoscopic Sinus Surgery Simulation

M.A. Audette, I. Hertel, O. Burgert, and G. Strauss

Abstract This paper presents on-going work on a method for determining which subvolumes of a patient-specific tissue map, extracted from CT data of the head, are relevant to simulating endoscopic sinus surgery of that individual, and for decomposing these relevant tissues into triangles and tetrahedra whose mesh size is well controlled. The overall goal is to limit the complexity of the real-time biomechanical interaction while ensuring the clinical relevance of the simulation. Relevant tissues are determined as the union of the pathology present in the patient, of critical tissues deemed to be near the intended surgical path or pathology, and of bone and soft tissue near the intended path, pathology or critical tissues. The processing of tissues, prior to meshing, is based on the Fast Marching method applied under various guises, in a conditional manner that is related to tissue classes. The meshing is based on an adaptation of a meshing method of ours, which combines the Marching Tetrahedra method and the discrete Simplex mesh surface model to produce a topologically faithful surface mesh with well controlled edge and face size as a first stage, and Almost-regular Tetrahedralization of the same prescribed mesh size as a last stage.

9.1 Introduction

Endoscopic sinus surgery is an intervention whereby the surgeon inserts a surgical instrument in one of the patients nasal passages, in a manner guided by an endoscopic view, to get to as well as resect a pathology that occupies one or more sinuses in that patients cranium. This procedure, illustrated in Fig. 9.1, also entails

M.A. Audette and O. Burgert
Innovation Center Computer Assisted Surgery (ICCAS), University of Leipzig, Leipzig, Germany
e-mail: michel.audette@medizin.uni-leipzig.de

I. Hertel and G. Strauss
Department of Otorhinolaryngology/Plastic Surgery, University of Leipzig, Leipzig, Germany

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

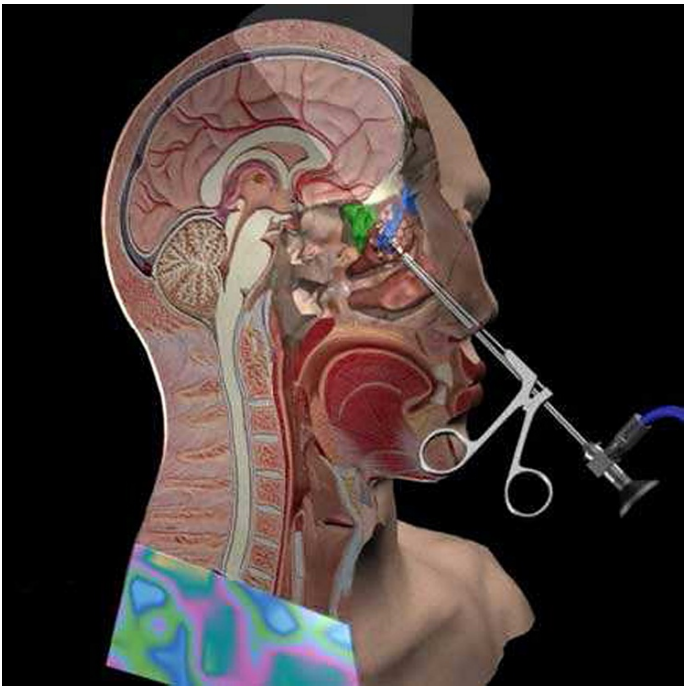


Fig. 9.1 Illustration of transnasal sinus surgery (Reproduced with permission from AAO-HNS [1])

the avoidance of critical tissues, such as the optic nerve(s) and eyeball(s) near the pathology at hand. In order to train future ENT surgeons on this procedure, we are developing methods for computing patient-specific models from routine Computed Tomography data. These models express the tissues relevant to biomechanical interaction in terms of simple shapes, or elements, typically triangles and tetrahedra, in order to make computationally tractable the numerical simulation of this interaction.

Currently, the anatomical models used in our simulation are obtained from manually labeled Tissue Maps, i.e.: an ENT surgeon manually identifies in CT relevant anatomical structures, namely the tumor, eyeball(s), optical nerve(s), and bone. Other, non-critical, soft tissues can be obtained by complementarity from CT, within a HU range of 500–1,500. Together, these tissues comprise a Tissue Map, which is the starting point of the methods described herein. The objective of this paper is to propose a method for selecting the tissue subvolumes relevant to the procedure, and therefore to the simulation, and for tesselating these tissues so as to express them in terms of triangular and tetrahedral elements for subsequent real-time interaction. The method proposed here does not preclude a Tissue Map obtained through minimally supervised methods, the development of which is underway [6].

Tissues deemed relevant to the simulation, to be modeled as elastic and therefore updated in real-time, are determined as *the union*

- *Of the pathology present in the patient*
- *Of critical tissues deemed to be near the planned surgical path or pathology*
- *Of soft tissue near these critical structures, or near a linearized, planned surgical path*

The relevant tissues must be tesselated and biomechanically modeled to allow a simulated interaction, where structures whose modeling is necessarily volumetric, such as the pathology to be resected, are decomposed into tetrahedral elements, while other structures are described in terms of triangular shell elements.

9.2 Materials and Methods

9.2.1 Distance-Based Biomechanical Relevance

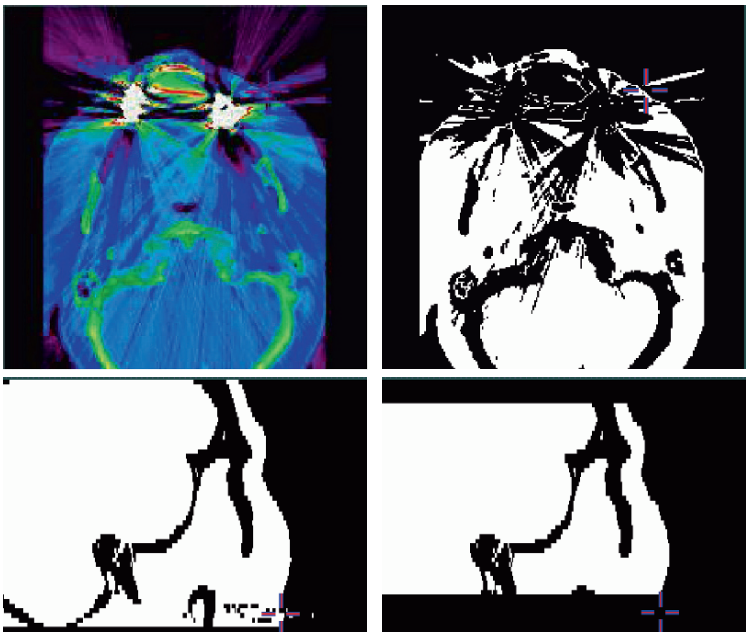
The first stage is the elaboration of a *Biomechanically Relevant Tissue Map*, which makes explicit those subvolumes of the Tissue Map that will be modeled for real-time interaction, with a view to excluding from consideration as much tissue as possible, for computational considerations, while keeping the simulation clinically meaningful. We make the assumption that non-pathological non-critical soft tissues that are relevant are relatively stiff and imbedded in bone, so that restricting our attention to a subvolume and modeling the rest as rigid is close to constitutive reality.

The first step in producing the Biomechanically Relevant Tissue Map is a pre-processing of the CT data, as shown in Fig. 9.2a, to ensure that the image on which the Tissue Map is based is free of artefacts, namely blackout artefacts caused by tooth fillings [18]. Currently, this is a simple cropping of the image volume: restricting the visible anatomical skin surface is viewed as preferable, from a simulation standpoint, to a slightly larger surface that is distorted by such an artefact. Furthermore, to make isosurface extraction well behaved, this stage in turn benefits from zero-padding the Tissue Map at the boundaries coinciding with tissue, as shown in Fig. 9.2b. Critical tissues not relevant to the procedure, such as those in the orbital region opposite to the pathology, are not considered further.

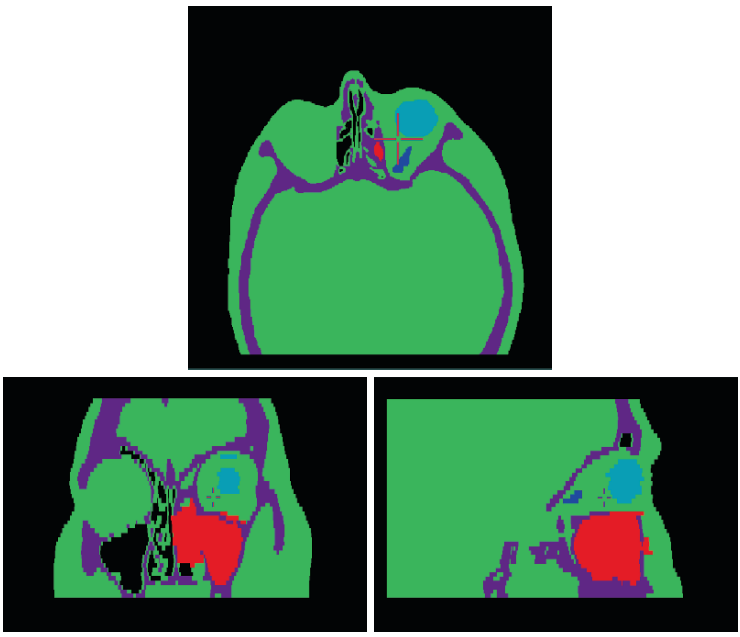
In order to produce a final Biomechanically Relevant Tissue Volume, the processing of tissues, prior to meshing volumetric processing is achieved with a combination of two approaches: MINC-based 3D image processing software from the Montreal Neurological Institute [11], and the well-known Fast Marching (FM) [15] method applied under various guises. First, *mincresample* and *mincreshape* are used to crop, zero-pad as well as orient a the sampling of a volume along *xyz*. Second, the FM method is used to restrict tissues under consideration, based on a **flexible notion of distance**, as idealized in Fig. 9.3. The basic idea here is that *useful notions of proximity include distances to a linearized path and to a blob, and that FM can implement all of these, even with anisotropically spaced data.*

The Fast Marching method numerically simulates the Eikonal equation in an efficient manner:

$$F(x)\| \bigtriangledown T(x)\| = 1 \quad x \in \Omega \tag{9.1}$$



(a)



(b)

Fig. 9.2 Preprocessing of CT data prior to determining biomechanically relevant tissues. (**a**) Elimination of blackout artefacts. Top: blackout artefact and distortive effect on soft tissue segmentation; bottom: sagittal illustration of cropping. (**b**) Tissue Map prior to determining biomechanically relevant subvolumes

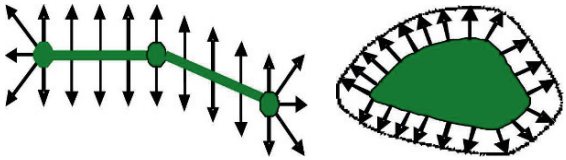


Fig. 9.3 Illustration of flexible distance, as defined from relevant primitives: (**a**) linearized path and (**b**) blob

which in turn estimates the time T of arrival of a monotonic front $\partial\Omega$ over a domain Ω, whose speed of propagation is $F(x)$. When $F(x)$ is unity, this method computes a distance map. In our case, this front $\partial\Omega$ can be initialized (at T = 0) as either the boundary of a given tissue (identification of the boundary is based on a non-zero gradient of a binarization of the tissue class vs all other tissues), or as the rasterization of a linearized path through user-provided points. Moreover, the method can deal with anisotropic spacing, as the upwind discretization of the gradient in Eq. (9.1) [13],

$$F_{i,j,k}\begin{bmatrix} max(D^{-x}_{i,j,k}T, -D^{+x}_{i,j,k}, 0)^2 + \\ max(D^{-y}_{i,j,k}T, -D^{+y}_{i,j,k}, 0)^2 + \\ max(D^{-z}_{i,j,k}T, -D^{+z}_{i,j,k}, 0)^2 \end{bmatrix} = 1\ , \tag{9.2}$$

makes use of voxel spacing h_x, in computing $D^{-x}_{i,j,k} = \frac{T_{i,j,k} - T_{i-1,j,k}}{h_x}$, $D^{+x}_{i,j,k} = \frac{T_{i+1,j,k} - T_{i,j,k}}{h_x}$, as well as h_y and h_z respectively for the other difference operators.

Furthermore, we can place tissue-based constraints on the motion of the front. For example, if we are interested only in the subset of non-critical non-pathological ("generic") soft tissue that is within a distance ε_{csp} of critical structures and pathology, *but not beyond bone tissue modeled as rigid*, we binarize and combine ("OR") together critical structures and pathology, determine their joint boundary, and use this boundary to initialize a *front enabled strictly on non-critical soft tissue*. In other words, we constrain the front to halt at bone or air tissue, under the assumption that most bone tissue will be modeled as rigid, and that therefore soft tissue on the other side of it is unlikely to be biomechanically relevant. Similarly, if we are interested in soft tissue near the planned surgical path, in order to model it as elastic, in contrast with the rest of the soft-tissue boundary, we can consider a *linearized intended surgical path S* determined by user points $P = \{\mathbf{p}_i\}$: $S = \{E_i(\mathbf{p}_i, \mathbf{p}_{i+1})\}$, where E_i is an edge linking successive points. The distance of voxel $\mathbf{x}$, $d_{SP}(\mathbf{x})$, to this surgical path is given by:

$$\begin{aligned} d_{SP}(\mathbf{x}) &= min_{E_i \in S} d_{edge}(\mathbf{x}, E_i) \ \text{ where} \\ d_{edge}(\mathbf{x}, E_i) &= min_{u+v=1} \left\| (u\mathbf{p}_i + v\mathbf{p}_{i+1} - \mathbf{x} \right\|. \end{aligned} \tag{9.3}$$

In practice, this distance is just a FM-based computation using an initial front coinciding with the voxels overlapping the edges E_i.

The end result is a refinement of the Tissue Map that discriminates between *biomechanically relevant* generic soft tissue, based on proximity to the planned surgical path or to critical or pathological tissue, and the *rest* of the generic soft tissue, which along with bone can be modeled as rigid in the simulation.

9.2.2 Tissue-Guided Surface Meshing of Controlled Mesh Resolution

Once we have our Biomechanically Relevant Tissue Map, featuring critical, pathological, as well as generic elastic and rigid tissues, our next step is to express them in terms of simple shapes, or elements, for biomechanical simulation. Our philosophy is to **use as few and as simple elements as required by the simulation**.

- In particular, the rigid tissue subvolume can be modeled exclusively as a collection of surface elements, since by definition no force will modify its interior.
- Likewise, we are not interested in how critical tissues, like the optic nerve, interact volumetrically with a surgical tool: we merely want the simulation to appropriately penalize a gesture that compromises its boundary, whereby surface meshing is once again sufficient.
- Last, the tumor that must be resected as well as elastic soft tissue surrounding the tumor and critical structures, whose interaction with a surgical tool should be inherently volumetric, are modeled as tetrahedral meshes.

As a result of the requirement for flexibility of choosing between surface and volumetric elements, the meshing is a *topologically faithful surface meshing stage followed by a tetrahedralization stage*, based on our prior work [2], as opposed to a method that tetrahedralizes a volume directly [8]. The surface meshing stage combines *Marching Tetrahedra*-based [3,12] anatomical boundary identification (or *contouring*) and an iterative *Simplex* mesh-based decimation [5]. The former provides our triangular surface mesh with fidelity to boundary and to topology, while the latter affords explicit control over triangular face area and edge size through the action of edge deletion and insertion. Equally important, the decimation preserves the topology (or *genus*) of the surface throughout each iteration.

In the past, we used the Marching Cubes (MCub) method [10] as our tissue boundary meshing method. However, Marching Cubes has been shown to exhibit limitations in topological fidelity and in regularity, which we can corroborate and which motivated us to opt for Marching Tetrahedra (MTet). The basic idea of these isosurface extraction methods is to proceed as follows.

1. First, tile the volume spanned by a scalar function, with regularly shaped cells such as cubes or tetrahedra.
2. Detect the nature of the intersection of an n-valued isosurface with each cell on the basis of the combination of relative values of vertices in relation to n (if $n = 0$ this equates with determining the $+$ and $-$ signs of the scalar function at the cell vertices) to ascertain which cell edges are split by this isosurface.

3. Next, express the intersection of the cell as a polygon, composed of 1–5 triangles depending on the combination of + and − vertices.
4. Finally, stitch together the faces of these cell traversals into a polygonal surface.

We might have opted for Regularized Marching Tetrahedra [20], which produces smoother surface boundaries, were it not for the fact that the Simplex model gives us control over surface continuity.

In [2], we showed that such an surface meshing method could be applied to binarized tissues volumes in a topologically faithful manner, preserving the ventricles within the brain boundary mesh, prior to controlled-resolution decimation. However, we then applied an image blurring stage to a binary image to produce a scalar function describing tissue occupancy, to stabilize MCub-based isosurface extraction. However, tight regions such as a nasal passage preclude blurring prior to surface meshing, so we have replaced this blurring stage by a *Signed Distance Map* computed from the tissue boundary, where *voxels inside the boundary are assigned a negative sign*, the *boundary is at a 0-distance*, and *points outside are given a positive distance*. This Signed Distance Map also makes use of the Fast Marching method described above: the sign is attributed through an appropriate test of the input tissue binary image at each voxel.

In general, these surface meshing methods produces dense results that need to be *decimated*, or simplified, in a manner that leads larger triangles still coincident with tissue boundaries, to support practical real-time applications such as surgery simulation. A survey of mesh decimation methods is given in [4], and the authors discriminate between coplanar facets merging, controlled vertex/edge/face decimation, which includes [14], re-tiling, energy function optimization, vertex clustering, wavelet-based methods and methods based on intermediate hierarchical representation. Many of the methods surveyed do not preserve topology (i.e.: can inappropriately introduce holes or plug existing holes in the surface), which makes them unusable in our application.

Our approach to decimation is of the edge-decimation type (and edge insertion if needed), through explicit logic based on local edge and face characteristics of the Simplex mesh at any given iteration. The Simplex is a *discrete active surface model, characterized by each vertex being linked to each of 3 neighbours by an edge*, and it is the *dual of a triangulated surface*, as shown in Fig. 9.4a. An image force can bind the model to the boundary of interest, even halting a model subject to a balloon inflation force, while other internal forces that improve face quality and continuity. Prior to decimation, the initialization of the Simplex from the MTet-based triangular surface is achieved through the geometric duality between triangular and Simplex meshes:

- Each triangular vertex is converted to a Simplex face
- Each triangular face leads to a Simplex vertex and
- Three Simplex edges link each Simplex vertex to its neighbours

We decimate the Simplex mesh with the T_1 and T_2 operators [5] that act on each face, where the former deletes an edge in order to fuse two faces into one, while the latter adds an edge to a face, subdividing it into two faces, as illustrated in Fig. 9.4b.

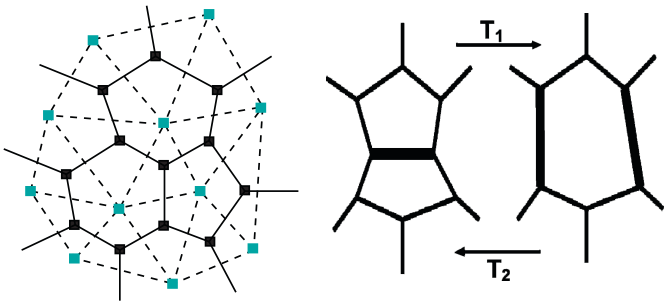


Fig. 9.4 Illustration of Simplex mesh: (**a**) close-up of Simplex, shown in black with solid line, with dual triangular mesh shown as broken line with blue vertices; (**b**) illustration of T_1 and T_2 operators

This logic allows us to *specify when to remove an edge, given a face that is too small in relation to a size objective, and in particular which choice of edge is best for that given face, in a manner to is conducive to biomechanical simulations*, whereas existing decimation methods tend to be optimal for surface rendering (i.e.: a mechanically static scene), producing dense results at high curvature points. Meshing that is dense at high-curvature points typically leads to an ill-conditioned finite elements stiffness matrix [16]. Moreover, the Simplex model is endowed with internal forces that enforce face quality and surface continuity characteristics, and external image forces that allows the surface to adhere to the tissue boundary, at each decimation iteration.

Our decimation strategy involves a penalty function computed for every face, as well as the sorting of all these faces by minheap or maxheap according to this penalty value, in order to "fix" the worst faces first. T_1 is applied to faces that are considerably smaller than desired and have a small (<6) number of edges, while we subdivide with T_2 those faces with a large number of edges (>7) and whose size is in excess of the desired resolution. The final Simplex boundaries can be converted to triangulated surfaces by duality.

9.2.3 Almost-Regular Volumetric Meshing with Resolution Control

The last stage in our procedure partitions each volume bounded by a triangulated mesh, coinciding with a tissue class or contiguous subset of tissue classes, into tetrahedral elements consistent with the FE method. The volumetric meshing stage is based on a technique published by Fuchs [7] that automatically produces an optimal tetrahedralization from a given polygonal boundary, such as a triangulated surface. In this case, optimality is defined as *near-equal length of the tetrahedral edges*, along with a *sharing of each inner vertex by a nearly consistent number of edges and tetrahedra*.

This method features the optimal positioning of inner vertices followed by a Delaunay tetrahedralization. The resulting near-regularity is important for FE stability and efficiency [16]. More importantly, and in keeping with the philosophy of our method, vertex positioning is governed by a prescribed density function $\rho(\mathbf{x})$, which can be spatially varying and correlates with the edge length objective pursued so far.

The positioning of vertices is divided into two stages:

- Construction of an initial configuration of vertices, based on the notion of a *canonical tetrahedron*, and in accordance the specified density function $\rho(\mathbf{x})$
- Adjustment of vertex positions by minimization of a penalty functional

A canonical tetrahedron leads to a partioning of R^3 with *congruent tetrahedra*, whose intersection is either empty, or a vertex, an edge or a face. In addition, it is invariant under subdivision.

Based on the relationship between the number of simplex and triangle vertices $V_t \approx V_{sm}/2$ [5] and a corresponding target simplex mesh size of $\mathcal{L}_{sm}(\mathbf{x})$ works out to a triangle or a tetrahedral mesh size of $\mathcal{L}_t(\mathbf{x}) \approx \sqrt{2}\mathcal{L}_{sm}(\mathbf{x})$. In summary, we modify the Fuchs tetrahedralization technique by specifying the target edge length $\rho(\mathbf{x}) = \mathcal{L}_t(\mathbf{x})$.

9.3 Results and Discussion

Figure 9.5 illustrates the results of distance-based tissue relevance strategy. First, Fig. 9.5a illustrates the consideration of a path-based distance to identify air voxels within $\varepsilon_{SP} = 10$ mm of the linearized path S.

Our tissue-discriminating FM-based approach offers advantages over a combination of logical and traditional morphological operators acting only on soft tissue, which could include soft tissues within a zone of influence based on the morphological structuring element's size, but on the other side of a thin bone or air duct. Our approach is also more useful than a generic region of influence approach, being more restrictive of the volume considered elastic. Proceeding this way gives us the flexibility to eventually model curviplanar tissues, such as membranes in the nasal passage, as relatively sparse "thick shell" elements.

In an earlier implementation, shown in Fig. 9.5b, we simply considered soft tissue voxels within $\varepsilon_{SP-air} = 4$ mm of this subset of air voxels in the nasal passage, as well as close to tumor (shown) and critical tissues ($\varepsilon_{csp} = 8$ mm in this case), with the subvolume of non-critical soft tissues modeled as elastic shown in light green. This approach resulted in a partition of tissue subvolumes, elastic and rigid respectively, that had a discontinuous outward appearance in the meshing, in the area of the eye socket and part of the nasal passage. We now endorse a slightly more subtle approach: we want to model volumetrically the non-critical soft tissues surrounding the tumor, eye ball and optic nerve, while at the same time not contained in some dilation band around air voxels (Figs. 9.5c, d). The nasal passage is handled as in the

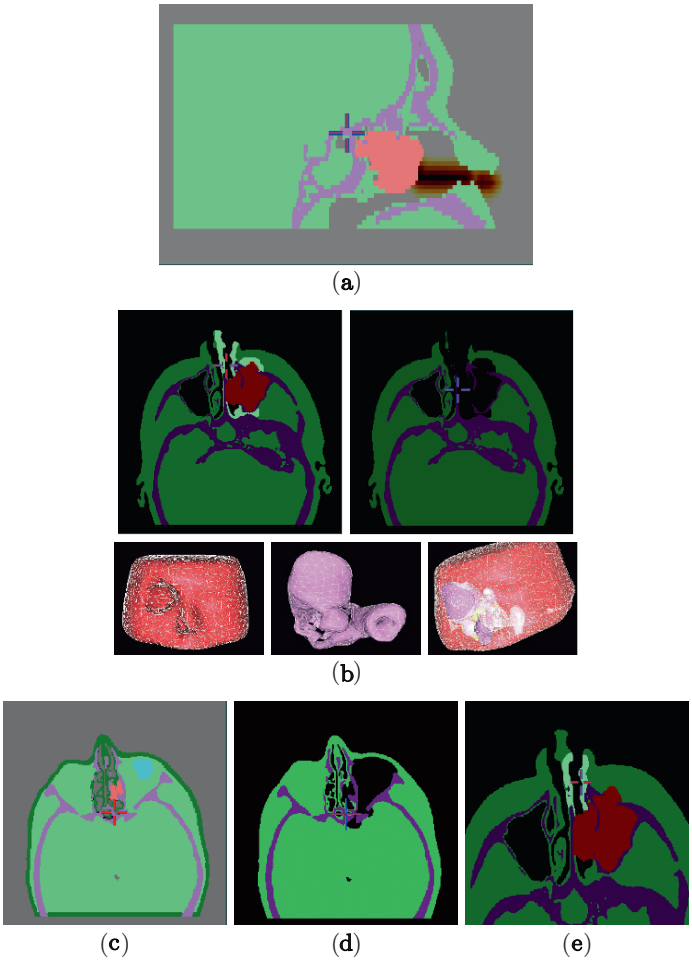


Fig. 9.5 Use of Fast Marching method for restricting tissues. (**a**) Identification of air voxels near linearized surgical path, $d_{SP} < 10$ mm, displayed as hot colour map. (**b**) Early implementation. Top left- axial plane images of idenfication of soft tissues near surgical path (close to air voxels in (**a**)), critical tissues and tumor, based on air- and bone-inhibited distance. Top right-soft tissue and bone modeled as rigid. Bottom: resulting surface meshing of (left) rigid tissue, (middle) elastic tissue, (right) with the two shown together. (**c**) Use of soft-tissue-enabled dilation of air voxels (dark green), to (**d**) restrict non-critical elastic volume to maintain integrity of rigid tissue, for continuous outer skin surface. (**e**) Elastic tissue in nasal passage

earlier implementation, albeit with a somewhat shorter path that models elastically only tissues inside. At all times, the algorithm processes information in 3D, not on a slice-by-slice basis.

Typical meshing results are presented in Figs. 9.6 and 9.7. Figure 9.6 provides a contrast between the decimation method of [14], which is optimized for static scenes as well as produces dense results in high curvature areas, and ours which produces

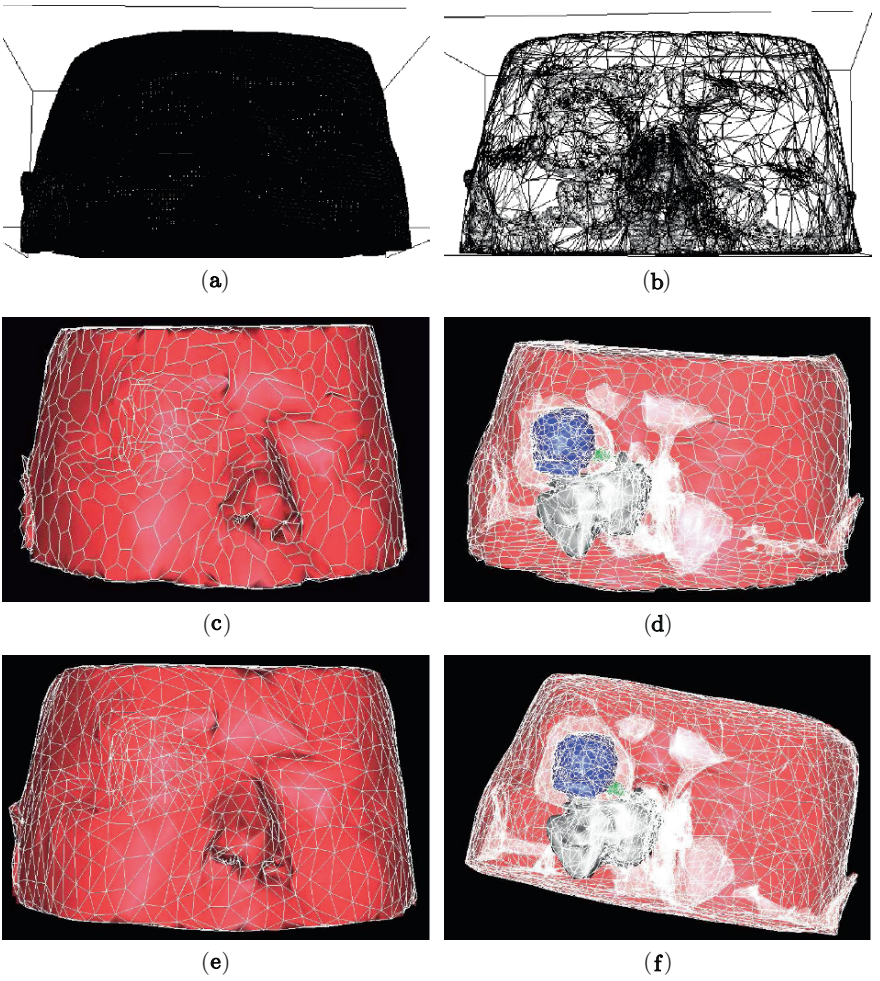
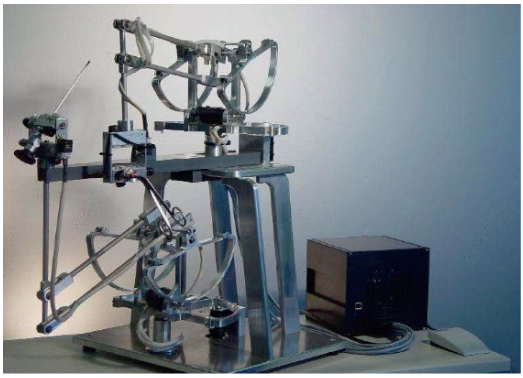


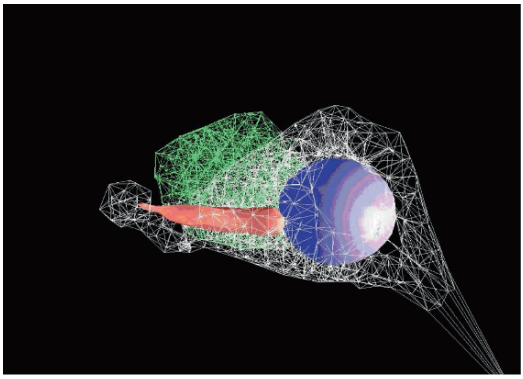
Fig. 9.6 Illustration of surface meshing strategy. (**a**) Original Marching Tetrahedra result of rigid tissue, featuring 3637412 triangles (too dense to discriminate between them visually); (**b**) typical decimation results with method [14]. Rigid tissue mesh, after simplex mesh decimation: (**c**) shown with wireframe and opaque surface rendering, and (**d**) semi-transparent rendering, and with wireframe-surface renderings of tumor (grey), optic nerve (green) and eyeball (blue). Elastic tissue surrounding critical tissues not shown, and some sinuses are also visible. Final triangulated surfaces, duals of (**c**) and (**d**) respectively

more regular shapes. Figure 9.7 illustrates tetrahedral meshing results, and their integration on the SPRING platform [17], along with haptic interaction provided by the IOMaster-7D device [19]. The mesh size currently is chosen arbitrarily, 8 mm for rigid surface, and 4 mm for other surface and tetrahedral meshing.

The combination of Marching Tetrahedra, Simplex and Almost-regular tetrahedralization provides topologically faithful meshing with controlled mesh size. The



(**a**)



(**b**)

Fig. 9.7 Tetrahedral mesh, capable of real-time interaction: (**a**) 2-handed haptic device; (**b**) non-critical elastic and tumor tissues, with non-critical tissue stretched haptically

parameters, such as distances and edge lengths, are currently set arbitrarily, but are being optimized based on on-going simulations and clinical interaction. This research also includes a method for simulating the bite of the Blakesley forceps [9], based on the 7 degrees of freedom of the dominant hand of the IOMaster-7D, whose prelinary results are illustrated in Fig. 9.8.

9.4 Conclusions

This paper presents a new method for selecting and meshing subvolumes of interest for ENT surgery simulation. It emphasizes the usefulness of surface models in particular, as Fast Marching and Simplex models are applied in tissue subvolume refinement and in meshing respectively. The tissue preprocessing uses the Fast Marching method under various guises to exploit distance to the surgical path and

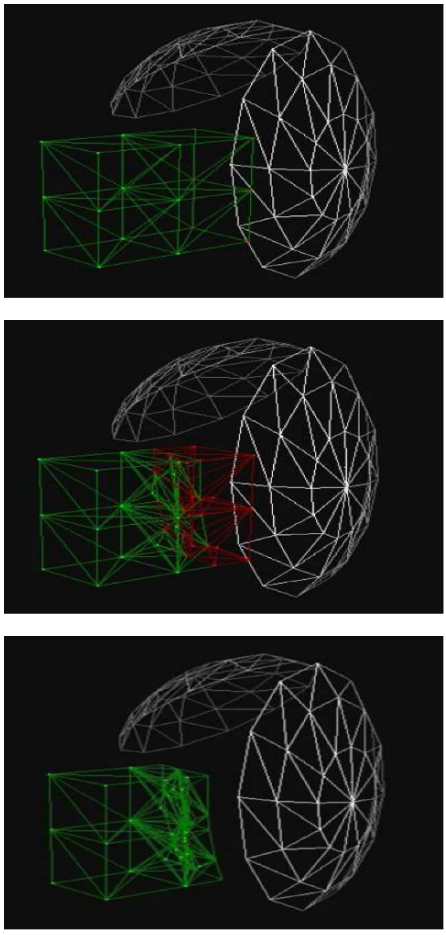


Fig. 9.8 On-going work on forceps bite simulation, in conjunction with a tetrahedral mesh. From top to bottom: initial configuration; tetrahedral mesh contained in bite illustrated in red; removal of contained mesh with subtetrahedral precision

to highly relevant tissues, to exclude from consideration irrelevant soft tissues. The meshing also features Almost-regular Tetrahedralization, and both surface and volume meshing afford explicit control over mesh size.

References

1. American Academy of Otolaryngology-Head and Neck Surgery (AAO-HNS), *http://www.entnet.org*, 2008.
2. M. A. Audette, H. Delingette, A. Fuchs, O. Burgert, and K. Chinzei, A topologically faithful, tissue-guided, spatially varying meshing strategy for computing patient-specific head models

for endoscopic pituitary surgery simulation, *Journal of Computer Aided Surgery*, 12:1, 43–52, Jan. 2007.
3. S.L. Chan and E.O. Purisima, A new tetrahedral tesselation scheme for isosurface generation, *Computers & Graphics*, 22:1, 83–90, Feb. 1998.
4. P. Cignoni, C. Montani, and R. Scopigno, A comparison of mesh simplification algorithms, *Computers and Graphics*, 22:1, 37–54, 1998.
5. H. Delingette, General object reconstruction based on simplex meshes, *Internal Journal of Computer Vision*, 32:2, 111–146, 1999.
6. F. Li, G. Strauss, C. Trantakis, and M.A. Audette, An iterative classification method of 2D CT head data based on statistical and spatial information, *Computer Aided Surgery Around the Head, CAS-H*, Feb. 2008.
7. A. Fuchs, Almost regular triangulations of trimmed NURBS-Solids, *Engineering with Computers*, 17, 55–65, 2001.
8. P.J. Frey, H. Borouchaki, and P.-L. George, Delaunay tetrahedralization using an advancing-front approach, *Proc. 5th Int. Meshing Roundtable*, Pittsburgh, 1996.
9. C. Grühser, N. Ritter, G. Strauss, H. Maass, and M. Audette, Development of a tool-centered collision model for volumetric resection in ENT surgery simulation, *EuroHaptics*, 2008.
10. W. Lorensen and H. Cline, Marching cubes: a high resolution 3D surface construction algorithm, *Computer Graphics*, 21:(4), pp. 163–170, July 1987.
11. Montreal Neurological Institute, Brain Imaging Software Toolbox, *http://www.bic.mni.mcgill.ca/software/*, 2008.
12. P. Ning and J. Bloomenthal, An evaluation of implicit surface tilers, *IEEE Computer Graphics and Applications*, 13:6, 33–41, 1993.
13. Elisabeth Rouy and Agnés Tourin, A viscosity solutions approach to shape-from-shading, *SIAM Journal on Numerical Analysis*, 29:3, 867–884, 1992.
14. W.J. Schroeder, J.A. Zarge, and W.E. Lorensen, Decimation of triangle meshes, *Computer Graphics*, 26:2, 65–70, 1992.
15. J.A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, 2nd ed., 1999.
16. J. Shewchuk, What is a good linear element? Interpolation, conditioning, and quality measures, *Eleventh International Meshing Roundtable*, 115–126, Sept. 2002.
17. SPRING Surgical Simulation Platform, Stanford University, *http://spring.stanford.edu/*.
18. P. Svendsen, L. Quiding, and I. Landahl, Blackout and other artefacts in computed tomography caused by fillings in teeth, *Neuroradiology*, 19:5, 229–234, July 1980.
19. C. Trantakis, J. Meixensberger, G. Strauss, E. Nowatius, D. Lindner, H. K. Çakmak, H. Maass, C. Nagel, and U.G. Kühnapfel, IOMaster 7D - a new device for virtual neuroendoscopy, *Computer Assisted Radiology and Surgery*, pp. 707–712, 2004.
20. G.M. Treece, R.W. Prager, and A.H. Gee, Regularised marching tetrahedra: improved iso-surface extraction, *Computers & Graphics*, 23:4, 583–598, Aug. 1999.

Chapter 10
A Robust Eye Tracking Procedure for Medical and Industrial Applications

Alberto De Santis and Daniela Iacoviello

Abstract An efficient eye tracking procedure is presented providing a non-invasive method for real time detection of a subject pupils in a sequence of frames captured by low cost equipment. The procedure can be easily adapted to any application relying on eye tracking. The eye pupil identification is performed by a hierarchical optimal segmentation procedure: a contextual picture zoning yielding the eyes position, and a further binarization extracting the pupils coordinates. No eye movement model is required to predict the future eyes position to restrict the image searching area, since the procedure first step is fast enough to obtain a frame to frame eyes position update.

10.1 Introduction

The process of visual search has a close relation to attentional mechanisms, providing insight into cognitive processes including memory, decision making, language comprehension [20, 21]. Recording and analyzing eye movements allow researchers to investigate how individuals make use of visual information. This is important in fields such as neuro-physiology [3], ergonomics [12], advertising and web design [1], education [23], disabled people technology [24], etc.

Dyslexic children phonological impairments are revealed by eye tracking during an auditory word recognition task [6]. The children were asked to look to a target item between distractor pictures; the experiment showed that although dyslexic children showed clear cohort competitor effects, they did not demonstrate delayed eye movements due to the presence of rhyme competitors. Eye position monitoring techniques are evaluated in [15] to test the perception/decision making process in

A.D. Santis and D. Iacoviello
Department of Informatica e Sistemistica "A. Ruberti" "Sapienza" University of Rome Via Ariosto 25, 00185 Rome, Italy
e-mail: desantis@dis.uniroma1.it, iacoviel@dis.uniroma1.it

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

radiology. In this study, visual search parameters such as the time to hit the location of the lesion for the first time, the amount of dwell time in the location, the total time analyzing the image were measured. The general characteristics of eye movements have been also studied in great depth during the process of reading. In [7] an eye tracking method is used to find a similarity based interference during language comprehension. On line measures of processing of critical regions of the sentences are reported, such as gaze durations, right-bounded reading times, rereading times, first-pass regression ratios and regression path reading times. An application of eye tracking technology (ETT) in the study of autism is presented in [2]. Normal adults show specific patterns when gazing at faces, fixating mainly on the eyes, nose and mouth, whereas subjects with autism spend shorter time in examining these core features. Eye tracking deficits are validated behavioural markers of risk of schizophrenia; in [18] eyes movements were monitored using an infrared limbus tracker during smooth pursuit and antisaccade tasks, determining a positive correlation between deficits on oculomotor task and latent liability to schizophrenia.

Meaningful indications on ergonomic criteria in user computer interface design can be obtained by the objective measurement of direction and time profile of gaze, duration of fixation on specific lookzones. The success of modern graphic user interfaces (GUI) relies on the performance and comfort of pointing devices. In [31] an evaluation of simple point-and-select techniques, as compared to standard mouse operation, is performed, measuring pupil and corneal reflection to compensate for head movement. Ergonomic design improvement can take advantage of the relation between eye movements and cognitive processing; explication of the role of attentional and memory processes in oculomotor control is discussed and suggestions on how to use these findings in display design, workload assessment, training and selection, is provided in [12].

The spread of internet has brought the attention on how users read through the web pages to get the information they need. The interesting articles in [1, 23] point out the importance of ETT to obtain suggestion on how to design an on line article to make the news easily accessible, or how to design a web page to label clearly the main information to direct the users to it. A well designed multimedia presentation can dramatically improve the class audience active learning. ETT again plays a prominent role to investigate the viewing of text and graphics [27]. Intelligent learning environments are designed to provide the students a tailored support to enforce learning from free exploration and capability to self-explain instructional material [16]. A vast area of eye tracking application is relevant to the design of human computer interfaces (HCI) for disabled people. The purpose consists in determining with sufficient accuracy the point of gaze of a subject exploring a PC screen to provide a smart tool improving the subject autonomy both in terms of communication and environment interaction [14, 24].

An eye tracking facility is a complex system composed of physical devices (such as video cameras, brain activity sensors, head movement sensors, special contact lenses etc.) and signal analysis algorithms. In [29] high-tech solutions are proposed; they have different degree of invasiveness and, depending on the user application, may result in the user stress and fatigue. Nevertheless the use of sophisticated

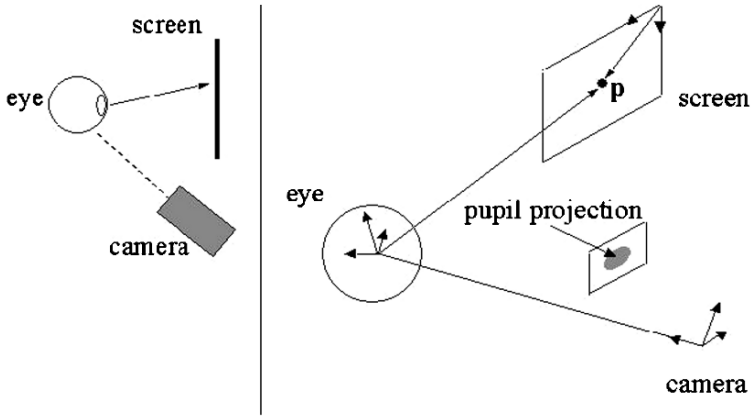


Fig. 10.1 General eye tracking set up and coordinate references

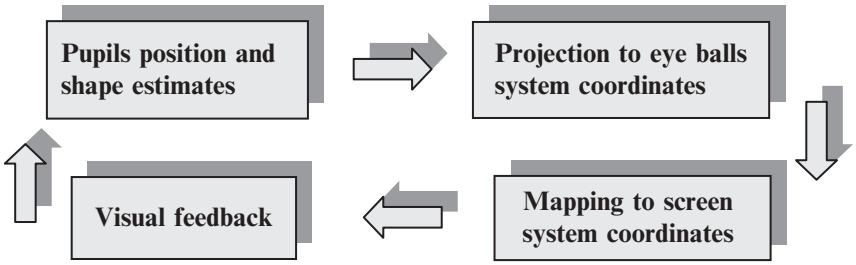


Fig. 10.2 Flow chart of the eye tracking system

technology considerably simplifies the signal analysis. As opposite, the use of non-invasive low cost technology demands complex algorithms able to deal with the real time constraint. Our work refers to the latter situation: the subject is positioned in front of the screen of a general purpose PC endowed with a low cost video camera. Generally three main reference systems can be identified [26]: the eye reference system, the camera reference system and the screen reference system, Fig. 10.1.

The tracking system analyzes the image captured by the camera, determines the line of gaze of the subject watching the screen and identifies the point of the screen the subject is looking at. The position of the pupil centre and the pupil shape on the image plane depend on the point on the screen the pupil is directed to, on the location of the user head, on the parameters of the eye model, on the position of the screen with the respect to the camera and on the camera intrinsic parameters. In Fig. 10.2 a general eye tracking functional flow chart is displayed. In this paper the focus is on the first block, presenting a robust and efficient procedure of image analysis to identify the pupils in a video sequence; the other blocks are just standard and a possible implementation can be found in [26].

Eye detection procedures usually exploit the pupil reflectance power to perform image zoning to separate the subject head from the environment and therefore

identify the pupil shape and center by template matching. In [13, 30], a deformable template is designed to fit at best the eye shape in the image; the eye identification procedure is usually accurate once properly initialized. Images with good contrast are required for better performances, whereas the computational load may be high. Papers [9, 10] present supervised learning algorithms to train a classifier with eye photometric appearance in different subjects under different face orientations. The quality of the performance depends on the completeness of the training set, therefore a large amount of data is required. Other methods rely on some eye characteristics such as dark pupil, white sclera, circular iris, eye corners, etc, to distinguish the human eye from the context. In [17], the eye tracking is performed by means of the Lucas Kanade feature tracker [28]; this method requires the initial location of the eye features points that are provided by picture zoning (Turk method [25]) and blink detection; all these steps require a high contrast image to detect and track eye corners. In [8], after a picture zoning is performed, the iris lower semi circle is detected as the curve that maximizes the normalized flow of the luminance gradient. In [32] eye detection and tracking make use of active remote IR illumination to generate bright/dark pupil images and therefore obtain the pupil identification by simple thresholding. The usual drawbacks due to non-perfect illumination, occlusion and eye closure are avoided by using a support vector machine; consequently eye tracking is performed by a Kalman filtering approach reinforced by a mean shift tracking algorithm. In [22] gaze direction and blinking are determined by tracking and motion analysis of eye corners, eyelids and irises. Template matching is exploited to identify the eye elements whereas their motion is determined by the optical flow associated with their edge segments; head motion is estimated to detect head-independent eye movements such as saccades or smooth pursuit. In [11], the *between-the-eyes* pattern is considered for feature tracking, and head movement cancellation is performed for easier eyelids movement detection.

In this paper a new method for pupils identification in video sequences is proposed; it relies on appearance and shape features of the eyes. The method is composed of two steps: a contextual frame zoning and a region growing image segmentation process, based on a discrete level set formulation, first presented in a more general setting in [4, 5] for still images. In the general set up we refer to, the pupils occupy a very small portion of the frame, making their detection hard to accomplish since no additional IR illumination is considered. This problem is usually avoided by frame zoning, obtained by cropping the frame around the estimated eye position. Nevertheless the eye elements and the other face elements (browses, nostrils, lips) still need to be separated for the detection of the pupils positions; moreover an eye motion predictor is required to update the area to be cropped from frame to frame. In this work the contextual frame zoning proposed performs a fast four levels segmentation by Otsu method [19], and therefore the eye can be easily separated by Boolean operations. This results in a binary image where the eyes are well separated from the other elements, and constitutes a mask that contextually selects the "zones" of the frame to be further analyzed. This solution does not require the motion predictor.

On the selected areas a further segmentation is performed obtaining a reliable pupils position estimation. This is accomplished by a level set procedure that is

proved to be accurate and robust especially in case of low contrast [4, 5]. The segmentation method presented in [5] is here adapted to the case of video sequences by considering a simpler cost functional consisting just in the error between data and their piece-wise constant approximation; moreover the interconnection between frames is obtained by updating the current frame segmentation starting from the segmentation of the previous frame.

The paper is organized as follows. The contextual zoning procedure is presented in Section 10.2, whereas in Section 10.3 the optimal discrete level set segmentation procedure is outlined. The features extraction and pupils tracking method is presented in Section 10.4 and the application of the proposed procedure to a real video sequence is performed in Section 10.5. Conclusions can be found in the last section.

10.2 Contextual Zoning

The general set up we refer to considers a user sitting in front of a PC screen endowed with a videocamera. In Fig. 10.3a, a typical subject snapshot is displayed. Compared to the picture size, the pupils extent over very small area regions with very low contrast, so to be hardly distinguishable from the other eye elements. In this situation any histogram based method would fail since the pupils poorly contribute to the global signal statistics. Picture zoning may solve this problem by cropping the image around the estimated eye position, thus making the pupil pixels to contribute more to the signal statistics. Nevertheless, depending on the signal quality in the selected image area, there can still be problems in a reliable pupil detection. Moreover, in the relevant literature, all methods presented propose the use of an eye movement predictor to basically estimate the next frame cropping zone.

In this work a more efficient zoning procedure is proposed; instead of an estimated region of interest, it selects elements of interest to be further segmented. These elements can be characterized by signal as well as shape features. Since in video sequence analysis the real time is important, only the gray level information is exploited in a fast four-levels segmentation based on a hierarchical thresholding procedure. The image is first binarized by Otsu method [19], and then the regions obtained are again binarized, thus obtaining a four level segmentation, Fig. 10.3b. In this simpler image representation the eyes are well separated objects among those with the lowest gray level; therefore they can be easily identified, with some other distinct elements, by a Boolean operation, Fig. 10.3c.

The result provides a mask selecting elements where the pupils can be more easily identified. Nevertheless, depending on the subject and/or on the illumination conditions, in the segmented image the eyes need not be completely separated in the ocular zone, Fig. 10.4a–c. In these cases a further Otsu binarization is required to obtain in an obvious way a mask as in Fig. 10.3c, see Figs. 10.4d–e.

Note that no eyes motion prediction is required since the Otsu thresholding is extremely efficient and therefore can be applied to the full size frame with negligible cpu time consumption.

Fig. 10.3 Contextual zoning; (**a**) original; (**b**) Otsu 4 levels segmentation; (**c**) mask of darkest elements

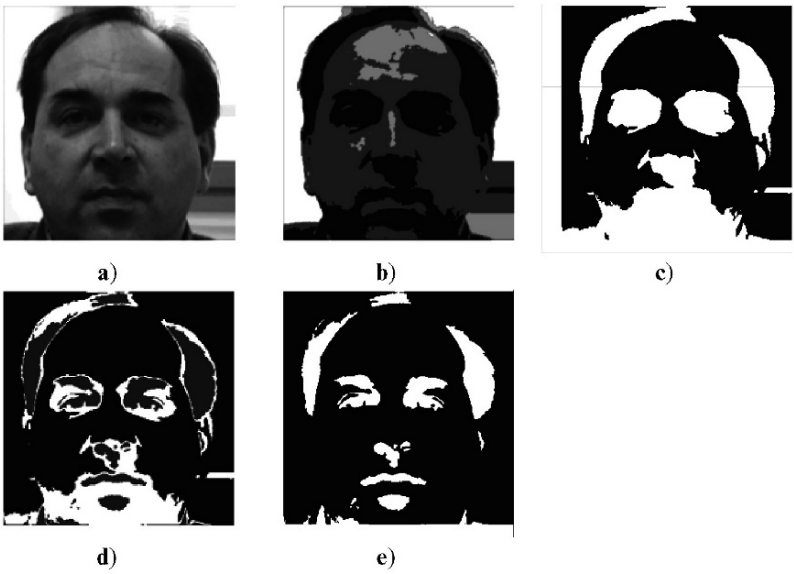


Fig. 10.4 Contextual zoning; (**a**) original; (**b**) Otsu 4 levels segmentation; (**c**) early mask; (**d**) binarization of the original on the domain selected by the early mask; (**e**) mask of darkest elements

10.3 The Optimal Segmentation Procedure

In this section a brief outline of a simplified version of the discrete level set segmentation procedure proposed in [5] is presented. Consider a simple 2D monochromatic image I with just one object over the background; the object boundary can be represented by the *boundary set* ϕ_0 of a function $\phi : D \rightarrow \mathrm{R}$, where $D \subset \mathrm{N}^2$ is a grid of points (pixels) representing the image domain. The boundary set is defined as follows

$$\phi_0 = \{(i,j) : sign\left(\phi_{h,k}\right) \neq sign\left(\phi_{i,j}\right) \\ for\ at\ least\ one\ (h,k) \in [i \pm 1,\ j \pm 1]\}$$

Let us assume that the region $\{(i,j):\phi_{i,j}\geq 0\}$ coincides with the object; the pixels not belonging to ϕ_0 are either in the interior of the object or in the background. In this case it is easy to obtain a binary representation I_s of the original picture

$$I_s = c_1\chi_{(\phi\geq 0)} + c_2\chi_{(\phi<0)} \tag{10.1}$$

where

$$\chi_{(\phi\geq 0)} = \begin{cases} 1 & \phi_{i,j}\geq 0 \\ 0 & otherwise \end{cases}, \qquad \chi_{(\phi<0)} = \begin{cases} 1 & \phi_{i,j}<0 \\ 0 & otherwise \end{cases} \tag{10.2}$$

and c_1, c_2 are two different positive constant gray level values. Function ϕ is called the *level set function* and, according to (10.1), operates the image segmentation. Segmentation (10.1) can be obtained by solving an optimal approximation problem defined as follows:

$$\begin{aligned} \min_{(c_1,c_2,\phi)} E(c_1,c_2,\phi) &= \min_{(c_1,c_2,\phi)} \left[\lambda\|I-I_s\|^2+\alpha\|\phi\|^2\right] \\ &= \min_{(c_1,c_2,\phi)} \left[\lambda\sum_{i,j}(I_{i,j}-c_1)^2 H(\phi_{i,j}) \right. \\ &\qquad \left. +\lambda\sum_{i,j}(I_{i,j}-c_2)^2(1-H(\phi_{i,j}))+\alpha\sum_{i,j}\phi_{i,j}^2\right] \end{aligned} \tag{10.3}$$

where H is the Heaviside function, λ and α are two positive parameters. The first two terms represent the fit error between the original data and the piece-wise constant approximation; the third term is a regularity term that makes the cost function convex [5].

Following the argument in [5], a smooth version of the cost function is advisable and it is obtained by substituting the Heaviside function in (10.3) by a smooth approximant

$$H_\varepsilon(\phi) = \frac{1}{1+\exp(-\phi/\varepsilon)} \tag{10.4}$$

It can be proved that the smooth version of problem (10.3) admits necessary and sufficient conditions for a unique global minimum that, by standard calculus, can be obtained by the following numerical scheme

$$c_1^n = \frac{\sum_{i,j} H_\varepsilon\left(\phi_{i,j}^n\right) I_{i,j}}{\sum_{i,j} H_\varepsilon\left(\phi_{i,j}^n\right)}, \qquad c_2^n = \frac{\sum_{i,j}\left[1-H_\varepsilon\left(\phi_{i,j}^n\right)\right] I_{i,j}}{\sum_{i,j}\left[1-H_\varepsilon\left(\phi_{i,j}^n\right)\right]} \tag{10.5}$$

$$\alpha\phi_{i,j}^{n+1} + \lambda\left[(I_{i,j}-c_1^n)^2-(I_{i,j}-c_2^n)^2\right]\delta_\varepsilon\left(\phi_{i,j}^n\right) = 0 \tag{10.6}$$

where δ_ε is the derivative of function H_ε.

The level set method has the amenable property that, during the level set evolution (10.6), the boundary set ϕ_0^n, starting from *any* initial shape ϕ_0^0, can *merge* and *split* in order to easily deal with the complex topology of the real world images.

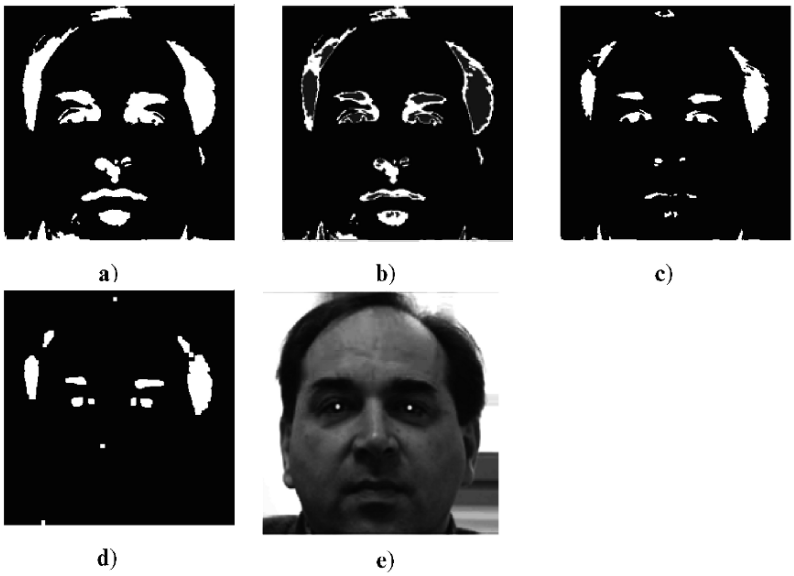


Fig. 10.5 Pupils identification; (**a**) mask from contextual zoning; (**b**) level set binarization; (**c**) darkest elements of (**b**); (**d**) morphological processing; (**e**) identified pupils marked on the original subject picture

In this work, the level set segmentation is applied to the frame regions selected by the zoning presented in Section 10.2. The result is shown in Fig. 10.5

A better centering of the pupil in the iris may be obtained after a morphological erosion of the level set binarization result.

10.4 Features Extraction and Pupil Tracking

The contextual zoning along with the level set binarization allows the selection of a set of eye features that can be reliably tracked in a video sequence: the principal elements of a human face are preserved and well separated one to another and from the rest of the scene. The eye browses, the eyelids, the irises, the pupils, the nostrils, the mouth and so for, can be individually analysed and their shape characterized by a proper set of quantities. The eye tracking system of Fig. 10.1 requires the measure of the pupil position and deformation to correctly determine the line-of-gaze of the subject and therefore the point of the screen the subject is looking at. Among the parameters that can be used to describe the position and deformation of the pupil we can consider: the centroid coordinates, the area, the major and minor axes and the inclination of the ellipse that best fits the pupil, the distance between the centroids of the right and left pupil. These quantities can be accurately estimated and stacked in two features vectors that represent the *eyes signature,* Fig. 10.6. Since

	left	right
centroid	176.8741 137.6643	91.3576 136.4636
area	143	151
major axis	15.8543	16.2065
minor axis	11.8402	12.2086
orientation	14.0832	14.0832

Fig. 10.6 Pupils signature vectors: shape and position parameters values

even in low cost video cameras the acquisition rate is at least 15 f/s, the scenes captured in adjacent frames do not differ significantly; therefore, moving from one frame to another, false detections can be avoided by simply matching the current frame features vectors to those of the eyes signatures of the previous frame. This matching procedure allows also to resolve the pupils occlusion due to either wide head motion or blinking: the last pupils position recorded is updated only if a pair of objects with matched characteristics are found in the current frame.

The eye-tracking algorithm requires that the current frame be segmented quickly so to obtain in real time the features vectors to be matched to the eyes signature of the previous frame. This can be accomplished if the segmentation is suitably fast. This is indeed the case since the early four levels segmentation is performed by the very fast Otsu thresholding procedure, whereas the successive step is just a binarization, accurately obtained by very few runs of algorithm (10.5), (10.6).

10.5 Experiments

The videotaping occurs in a regular room illumination; no additional light sources are considered, like IR lamps. The tracking procedure is composed of the following steps:

- Video acquisition and signal preprocessing; conditioning may be required for signal equalization, like gamma correction or histogram equalization.
- Otsu four levels segmentation; it is performed according to the hierarchical thresholding described in Section 2. It provides a *cartoon image* of the original data where the eyes always belong to the darkest part, whatever the colour of the eyes and the race of the subject. A simple boolean operation provides a mask, highlighting the dark elements (and therefore the eyes) and leaving the rest of the picture in the black background.
- Contextual zoning and binarization; the mask obtained in the previous step selects the frame zones where a binarization is performed according to the algorithm presented in Section 10.3.

- Features extraction; once the pupils are well separated their signatures vectors are built by collecting the following shape parameters: centroid coordinates, area, major and minor axes of the best fitting ellipse, orientation as the angle formed by the major axis and the horizontal axis, the distance between the pupils.

In the sequence analysed, the subject performs a wide head movement: saccades and partial/total eye occlusion occur. On each frame gaussian filtering with $\sigma = 1$ and gamma correction with $\gamma = 0.5$ are used. The Otsu method is then applied for the early four levels segmentation; this step does not require any parameter choice. After the contextual zoning the level set binarization is performed with $\lambda = 10^3$, $\alpha = 1$, $\varepsilon = 1$. The high value of λ ensures the segmentation accuracy and a fast convergence of Eq. (10.6). The difference in magnitude between λ and α makes the two terms of the cost functional E comparable; indeed, the value of the level set function ϕ are of two orders greater than those of the fit error.

In Fig. 10.7 nine snapshots of the analysed video sequence are displayed. In the first row the subject performs a smooth pursuit (like during the reading of a text

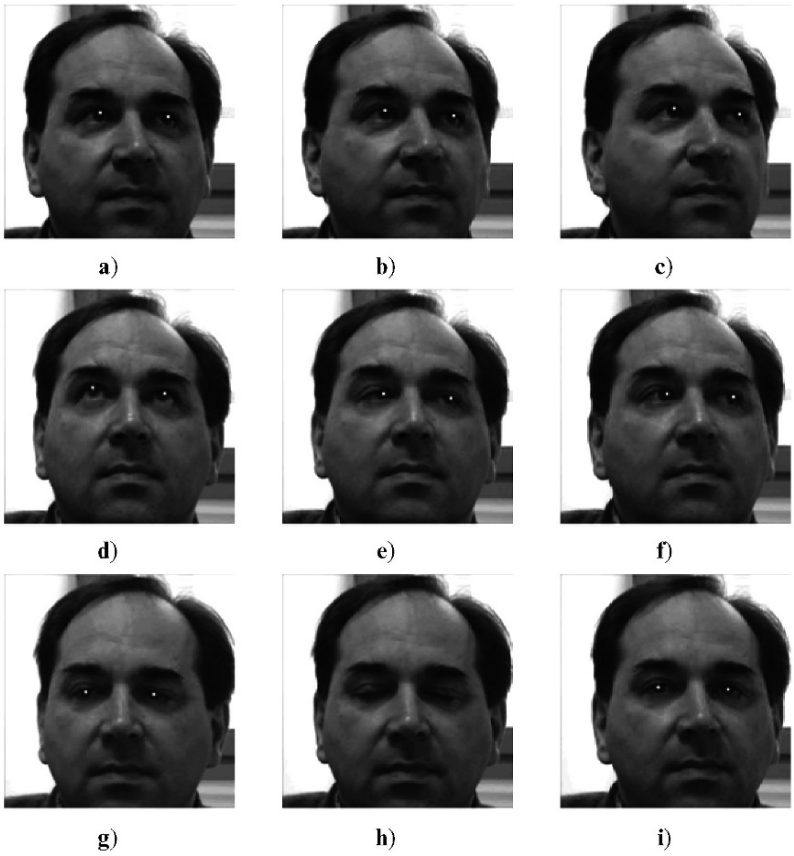


Fig. 10.7 Samples of the sequence analysed; (**a**)–(**c**) smooth pursuit; (**d**)–(**e**) saccade; (**g**)–(**i**) occlusion

line) and the pupils are well detected. In the second row an instance of a saccade is represented (the eyes are suddenly pointed in a direction different from the head orientation); also in this case the proposed procedure denotes a good performance. Finally, the third row of Fig. 10.7 shows the ability of the tracking method in resolving the occlusions: in the sequence samples the subject closes the eyes; the tracker records the last pupils signatures and does not update them until a couple of objects with similar signatures are found in the successive frames. The similarity is of course measured by the Euclidean norm of the difference between the signature vectors.

10.6 Conclusions

Eye tracking systems should have some key characteristics: non-invasive low cost equipment, extremely simple calibration and robustness with respect to changes in illumination conditions. The authors refer to a standard set up in which the subject sits in front of a workstation endowed with a regular video camera, with no specific source of illumination. The proposed eye tracking procedure first uses the Otsu method to build a mask separating the eyes from the other face elements. Then a further optimal level set binarization identifies the pupils so that some of their shape parameters can be recorded in two signatures vectors. These are used to track the pupils motion throughout the video sequence. This procedure is reliable: it enjoys the accuracy and robustness of the region based segmentation procedures, whereas high numerical efficiency is obtained due to the discrete formulation.

Acknowledgements The Authors thank Prof. Qiang Ji, at Renssealer Polytechnic Institute, Troy, New York, who granted them the use of pictures taken from the Intelligent System Lab data base. http://www.ecse.rpi.edu/homepages/cvrl/database/database.html

References

1. Bojko A (2006) Using Eye tracking to Compare Web Page Designs: a case study, Journal of Usability Studies. Journal of Usability Studies (1) 3: 112–120
2. Boraston Z, Blakemore S J (2007) The application of the eye-tracking technology in the study of autism. The Journal of Physiology 581: 893–898
3. Chen Y, Levy D L, Sheremata S, Holzman P S (2006) Bipolar and schizophrenic patients differ in patterns of visual motion discrimination. Schizophrenia Research 88: 208–216
4. De Santis A, Iacoviello D (2006) Optimal segmentation of pupillometric images for estimating pupil shape parameters. Computer Methods and Programs in Biomedicine, special issue on Medical Image Segmentation 84: 174–187
5. De Santis A, Iacoviello D (2007) Discrete level set approach to image segmentation. Signal, Image and Video Processing (1) 4: 303–320
6. Desroches A S, Joanisse M F, Robertson E K (2006) Specific phonological impairments in dyslexia revealed by eyetracking. Cognition 100: B32–B42

7. Gordon P C, Hendrick R, Johnson M, Lee Y (2006) Similarity-based interference during language comprehension: evidence from eye tracking during reading. Journal of Experimental Physiology,: Learning, Memory and Cognition (32) 6: 1304–1321
8. Hammal Z, Massot C, Bedoya G, Caplier A (2005) Eyes Segmentation Applied to Gaze Direction and Vigilance Estimation. International Workshop on Pattern Recognition for Crime Prevention, Security and Surveillance, LNCS 3687, Springer, Berlin, Heidelberg, pp 236–246
9. Huang J, Wechsler H (1999) Eye detection using optimal wavelet packets and radial basis functions. International Journal of Pattern Recognition and Artificial Intelligence (13) 7: 1009–1025
10. Huang W M, Mariani R (2000) Face detection and precise eyes location. Proceedings of the 15th International Conference on Pattern Recognition 4: 722–727
11. Kawato S, Tetsutani N (2004) Detection and tracking of eyes for gaze-camera control. Image and Vision Computing 22: 1031–1038
12. Kramer A F, McCarley J S (2003) Oculomotor behaviour as a reflection of attention and memory processes: neural mechanism and applications to human factors. Ergonomics Science (4) 1–2: 21–55
13. Lam K M, Yan H (1996) Locating and extracting the eye in the human face images. Pattern Recognition 29: 771–779
14. Majaranta P, Raiha K J (2002) Twenty Years of Eye Typing: System and Design Issues. In Eye Tracking Research & Applications: Proceedings of the Symposium on ETRA, New York, 15–22
15. Mello-Thoms C, Gur D (2007) Remote vs. head-mounted eye-tracking: a comparison using radiologists reading mammograms. Medical Imaging 2007: Image Perception, Observer Performance, and Technology Assessment, Proceedings of the SPIE, 6515: 65150C
16. Merten C, Conati C (2006) Eye-Tracking to Model and Adapt to user Meta-Cognition in Intelligent Learning Environments. International Conference on Intelligent User Interfaces, Proceedings of 11th International Conference on Intelligent User Interfaces, Sidney, Australia, pp 39–46
17. Morris T, Blenkhorn P, Zaidi F (2002) Blink detection for real-time eye tracking. Journal of Network and Computer Applications 25: 129–143
18. O'Driscoll G A, Lenzenweger M F, Holzman P S (1998) Antisaccades and smooth pursuit Eye Tracking and Schizotypy. Archives of General Psychiatry 55: 837–843
19. Otsu N (1979) A threshold selection method for gray level histogram. IEEE Transactions on Systems, Man and Cybernetics 9: 62–69
20. Radach R, Hyona J, Deubel H (2003) The mind's eye: cognitive and applied aspects of the eye movement research. Elsevier, North Holland
21. Richardson D C, Spivey M J (2004) Eye-Tracking: Characteristics and Method. In: Wnek G, Bowlin G (eds) Encyclopedia of Biomaterials and Biomedical Engineering. Marcel Dekker, New york, pp 568–572
22. Sirohey S, Rosenfeld A, Duric Z (2002) A method of detecting and tracking irises and eyelids in video. Pattern Recognition 35: 1389–1401
23. Slykhuid D A, Wiebe E N, Annetta L A (2005) Eye-Tracking Students' Attention to PowerPoint Photographs in a Science Eudcation Setting. Journal of Science Education and Technology (14) 5–6: 509–520
24. Su M C, Wang K C, Chen G D (2006) An eye tracking system and its applications in aids for people with severe disabilities. Biomedical Engineering: Applications, Basis & Communications (18) 6: 319–327
25. Turk M A (1991) Interactive-Time Vision: Face Recognition as a Visual Behaviour, Doctoral Thesis, MIT
26. Villanueva A, Cabeza R, Porta S (2006) Eye tracking: Pupil orientation geometrical modelling. Image and Vision Computing 24: 663–679
27. Wiebe E N (2005) Eye tracking Methods for improving Engineering Graphics Instruction" Proceedings of the 2005 American Society for Engineering Education Annual Conference Exposition, Washington, DC

28. Xie X, Sudhakar R, Zhuang H (1994) On improving eye feature extection using deformable templates. Pattern Recognition 27: 791–799
29. Yu L H, Eizenman M (2004) A new methodology for determining point-of-gaze in head-mounted eye tracking systems. IEEE Transactions on Biomedical Engineering (10) 51: 1765–1773
30. Yuille A, Hallinan P, Cohen D (1992) Feature extraction from faces using deformable templates. International Journal of Computer Vision (8) 2: 99–111
31. Zhang X, MacKenzie I S (2007) Evaluating Eye Tracking with ISO 9241 – Part 9. In Human Computer Interaction, Part III, LNCS 4552, Springer, Berlin, Heidelberg, pp 779–788
32. Zhu Z, Ji Q (2005) Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. Computer Vision and Image Understanding 98: 124–154

Chapter 11
3D Reconstruction of the Retinal Arterial Tree Using Subject-Specific Fundus Images

D. Liu, N.B. Wood, X.Y. Xu, N. Witt, A.D. Hughes, and Thom SAMcG

Abstract Systemic diseases, such as hypertension and diabetes, are associated with changes in the retinal microvasculature. Although a number of studies have been performed on the quantitative assessment of the geometrical patterns of the retinal vasculature, previous work has been confined to 2 dimensional (2D) analyses. In this paper, we present an approach to obtain a 3D reconstruction of the retinal arteries from a pair of 2D retinal images acquired in vivo. A simple essential matrix based self-calibration approach was employed for the "fundus camera-eye" system. Vessel segmentation was performed using a semi-automatic approach and correspondence between points from different images was calculated. The results of 3D reconstruction show the centreline of retinal vessels and their 3D curvature clearly. Three-dimensional reconstruction of the retinal vessels is feasible and may be useful in future studies of the retinal vasculature in disease.

11.1 Introduction

The retina, which lies at the posterior fundal surface of the eye, has the highest oxygen requirement per unit weight of any tissue in the body [10] and this makes it particularly vulnerable to vascular insults impairing oxygen and nutrient supply. Retinal vascular anatomy and net-work structure are adversely affected by high blood pressure, diabetes mellitus, ageing and atherosclerosis. Diabetic eye disease is one of the commonest causes of blindness in UK. A number of studies have shown that generalized arteriolar narrowing and retinopathy are associated with in-creased risk of stroke, ischaemic heart disease, heart failure, renal dysfunction and cardiovascular mortality [13]. Therefore, quantitative assessment of the retinal vascular network is very important.

D. Liu, N.B. Wood, and X.Y. Xu
Department of Chemical Engineering, Imperial College London, UK

N. Witt, A.D. Hughes, and Thom SAMcG
International Centre for Circulatory Health, Imperial College London, UK

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

The geometric patterns of the retinal microvascular network are readily observed in vivo using fundal photography [12]. However, quantitative analysis of the geometrical patterns requires vessel segmentation and reconstruction. The reconstruction of the retina, especially the area of optic disc, has been performed by several researchers [6, 15], but most pathological changes in the microvasculature occur away from this region.

3D reconstruction of the retinal vascular tree from fundal images is a considerable challenge and only a few such attempts have been made so far [9]. When subject-specific retinal images are obtained with a fundus camera, the intrinsic parameters of the fundus camera-eye system will be altered by the relative displacement between the camera and the eye of the subject. These changes can be reduced by acquiring retinal images with relatively small displacements of the camera in a plane which may be assumed parallel with the surface of the lens. Consequently the intrinsic parameters may be assumed to be fixed. The retinal vessels of interest can be segmented using a semi-automatic approach [8] and the point-by-point correspondence between different images can also be calculated. In order to acquire a metric reconstruction result, an essential matrix based self-calibration approach was performed to estimate the intrinsic parameters of the "fundus camera-eye" optical system. For this the pixels of the camera-eye system are assumed to be nearly perfectly rectangular (which means that the aspect ratio is considered to be one and there is no skew) and the principal point of the camera-eye system is assumed to lie at the centre of the final retinal image. With these assumptions the self-calibration approach can be reduced to a simple system and a final metric 3D reconstruction can be recovered after retrieving the projection matrix from the essential matrix.

11.2 Methodology

11.2.1 Image Acquisition

The retinal images for reconstruction were obtained in a normal subject following mydriasis with tropicamide (1% eye drops) using a commercial retinal fundus camera (Zeiss FF 450+ with a 30° field of view (Fig. 11.1). The fundus camera is a specialized low power microscope with an attached camera designed to photograph the interior surface of the eye, including the retina, optic disc, macula, and other structures. Digitized images were captured using a CCD camera and transferred to a PC for analysis. The principle of the paired image acquisition is illustrated in Fig. 11.2 and a stereo (approximate) pair of retinal images is shown in Fig. 11.3.

Although the eye and fundus camera are very complex, they are combined and simplified as one single lens in the analysis described here. Because this special system combines the eye and the fundus camera, the displacement between them, such as the change in relative distance and rotation, will alter the intrinsic parameters of the combined eye-camera system (Fig. 11.2). In order to minimize these changes, the distance between the camera and eye was held approximately constant and only a

Fig. 11.1 Retinal imaging

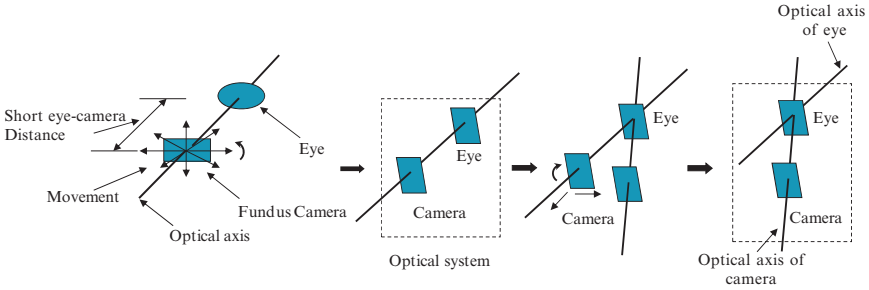


Fig. 11.2 Schematic diagram illustrating the approach to retinal imaging

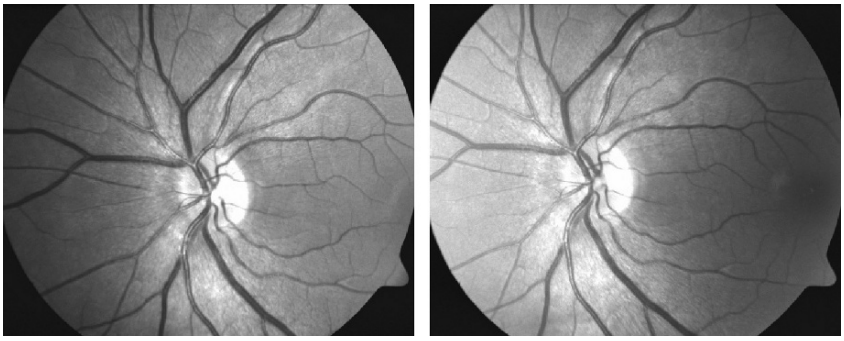


Fig. 11.3 A stereo pair of retinal images for reconstruction

small relative displacement of the fundus camera was made when the retinal images were acquired. Retinal images were acquired with a resolution of $1{,}280 \times 1{,}030$ pixels but were reduced to 499×402 pixels prior to analysis in order to reduce the computational time.

11.2.2 Feature Points Extraction and Vessel Segmentation

Depending on the theory of 3D reconstruction from 2D images, two images of a single scene are related by the epipolar geometry, which may be represented by a 3×3 singular matrix called the fundamental matrix F. It captures all geometric information contained in the two images, and must first be estimated for the reconstruction. A standard linear camera calibration matrix K has the following entries:

$$K = \begin{bmatrix} f & s & u_0 \\ 0 & \delta f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where f is the focal length in pixels and δ is the aspect ratio. $(u_0,\ v_0)$ are the coordinates of the principal points, and s is the skew factor which is zero for rectangular pixels.

Generally in order to obtain the fundamental matrix F, at least 7 corresponding fiducial points (i.e. at bifurcations) should be obtained. In fact, more than seven matches are required for an accurate estimation. Therefore a semi-automatic method [8] was used to perform the feature points extraction and vessel segmentation by scale-space analysis of the 1st and 2nd derivatives of the image intensity profiles (Fig. 11.4). The coordinates of the individual pixels corresponding to the centrelines of the segmented vessels are recorded. Because of the limited resolution of the fundus camera, only trunk branches of retinal vessels such as the 1st to 4th generations were clear enough to be analyzed (Fig. 11.5). Matching points were selected by an operator and are marked out in Fig. 11.6. If the same vessel were segmented from different images, the correspondence between them could be obtained.

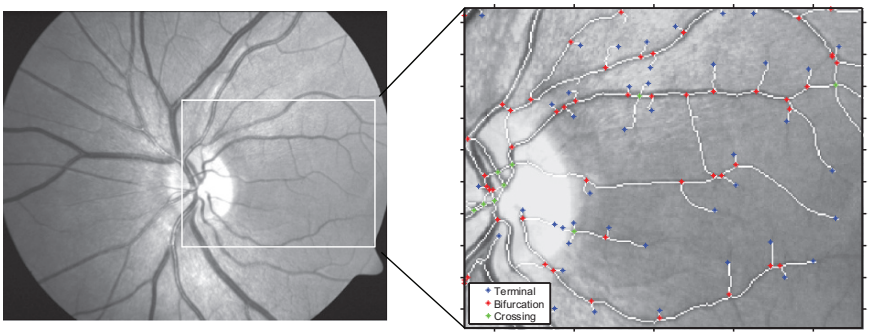


Fig. 11.4 Feature points extraction. Left: original retinal image; right: analyzed vessels

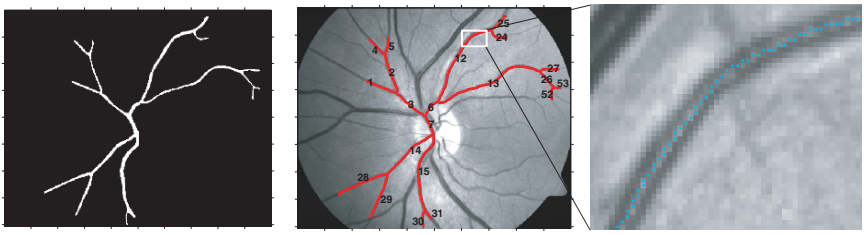


Fig. 11.5 Vessel segmentation. Middle: original retinal images; left: the segmentation result of the red marked vessels; right: the recorded vascular centreline points showing the vessel sect in the white rectangle area

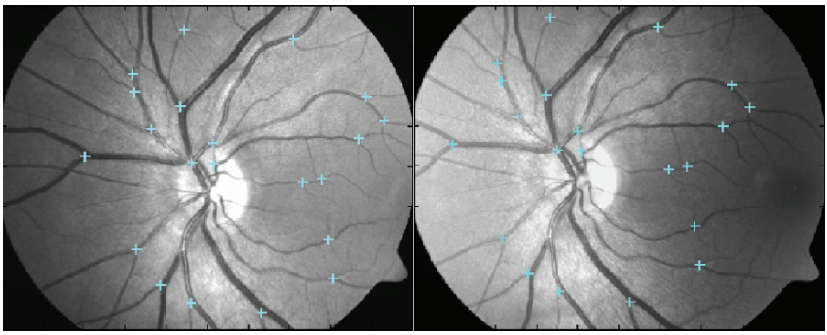


Fig. 11.6 Retinal images with marked corresponding points

11.2.3 Estimation of Epipolar Geometry

Based on the marked corresponding points (crossing and bifurcation points), the fundamental matrix F was recovered by applying the gold standard algorithm developed by Hartley and Zisserman [5] by minimizing the re-projection geometric error:

$$\sum_i d(x_i, \hat{x}_i)^2 + d(x_i', \hat{x}_i')^2 \tag{11.1}$$

where $x_i(u_i,\ v_i) \leftrightarrow x_i'(u_i', v_i')$ are the marked correspondences, and $\hat{x}_i \leftrightarrow \hat{x}_i'$ are the estimated correspondences that satisfy $\hat{x}_i F \hat{x}_i' = 0$ exactly for rank-2 estimated fundamental matrix F.

The gold standard algorithm was implemented by taking the following steps:

1. Obtain the initial estimated rank 2 fundamental matrix $\hat{F}$ using the normalized 8-points algorithm [3]:
   - Normalization: transform the corresponding points according to $\hat{x}_i = T x_i$ and $\hat{x}_i' = T' x_i'$, where T and T' are normalized transforms consisting of a translation and scaling.

- Find the fundamental matrix $\hat{F}$ corresponding to the matches $\hat{x}_i \leftrightarrow \hat{x}'_i$ with a linear solution and enforce the rank 2 constraint to it by Singular Value Decomposition (SVD).
- Set final fundamental matrix $F = T'^{\mathrm{T}}\hat{F}T$, where superscript T represents the transpose of the vector.

2. Compute an initial estimated subsidiary variables $\{\hat{x}_i \leftrightarrow \hat{x}'_i\}$ as follows:
   - Define two projection matrices as $P = [I\,|0]$ and $P' = [[e']_\times F\,|e']$, where e', the epipole of the second image, could be obtained from F.
   - From the correspondence $x_i \leftrightarrow x'_i$, the 3D points $\hat{X}_i$ are obtained by an iterative linear-eigen triangulation method [4] in order to save computational cost.
   - The projective correspondence consistent with F is obtained by $\hat{x}_i = P\hat{X}_i$, $\hat{x}'_i = P'\hat{X}_i$.
3. Minimize the cost function in Eq. (11.1) by varying $P' = [M|t]$ and $\hat{X}_i$ with the Levenberg-Marquardt algorithm.
4. Compute the fundamental matrix F as $F = [t]_\times M$.

This process of estimating the fundamental matrix is effectively equivalent to a projective reconstruction. Epipolar lines of the two retinal images calculated by the above algorithm are shown in Fig. 11.7.

11.2.4 Self-Calibration

In principle, a projective reconstruction can be obtained without the calibration matrix based on the fundamental matrix, F, but, in practice, due to the ambiguity of projective reconstruction, results may not be sufficiently accurate. Therefore the 3D reconstruction was made on the basis of a metric projection. It is known that a metric reconstruction of a scene may be computed by using the essential matrix E which could be derived from the calibration matrix K [5]:

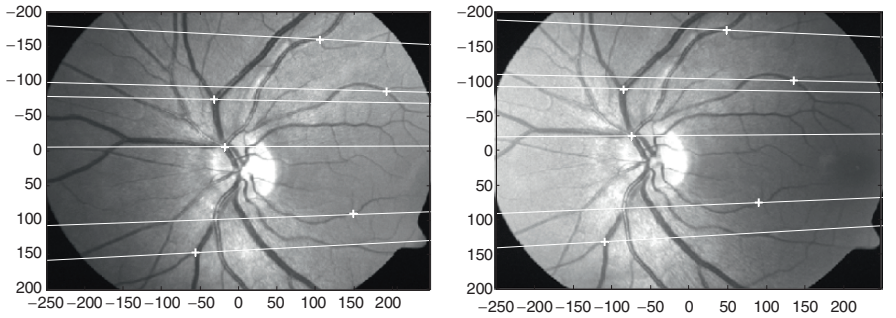


Fig. 11.7 The epipolar lines of the two retinal images

$$E = K^T F K \tag{11.2}$$

For the special fundus camera-eye system, a general photogrammetric calibration method, which depends on a calibration object with a known 3-D geometry [14,16], is not available. Instead a self-calibration method was employed. Since the image acquisition process was specially designed to minimize alteration of the camera-eye system, we can assume that the intrinsic parameters of the camera-eye system remained constant, and the pixels could be considered as nearly perfectly rectangular with an aspect ratio of 1 and no skew. The principal point of the camera-eye system is assumed to be at the centre of the final retinal image. Therefore the only unknown parameter of the calibration matrix is the focal length.

It has been recognized that if n_k is the number of intrinsic parameters known in all views and n_f is the unknown but constant intrinsic parameter, the number of views, m, required for self-calibration will be:

$$mn_k + (m-1)n_f \geq 8 \tag{11.3}$$

Therefore, a minimum of two views is required in this case [11].

A self-calibration method, based on the characteristics of the essential matrix E, was used to recover the unknown focal length: two of the three singular values of E should be identical and the other should be zero [11]. The cost function:

$$C = \omega_{12} \frac{\sigma_1 - \sigma_2}{\sigma_2} \tag{11.4}$$

was minimized by a direct search algorithm. $\sigma_1 > \sigma_2$ are the non-zero singular values of $E = K^T F K$, and ω_{12} is a weighting factor which represents the degree of confidence in the estimation of the fundamental matrix F. There are several possible choices for ω_{12}: (i) the residual of the estimation of F – the inverse of the mean geometric distance between the image points and their corresponding epipolar lines, (ii) the number of points used in the computation of F, and (iii) simply set it to one [1]. After obtaining the focal length by this self-calibration method, the essential matrix E could be calculated from Eq. (11.2).

Apart from sensitivity to the noise of images, the application of self-calibration is always dependent on the issue of initialization. Since the nonlinear minimization is used for self-calibration, convergence to the global minimum can be guaranteed only if the algorithm is initialized properly. Although this algorithm has a good global convergence according to Fusiello [2], we still employed a planar chessboard based photogrammetric calibration approach [7] to generate initial values for self-calibration. A series of images of a 2-D chessboard plane were acquired by moving and rotating the fundus camera (Fig. 11.8). The chessboard should not be exactly parallel to the image plane, because the calibration method applied here depends on the vanishing points. Otherwise, the vanishing points for the horizontal and vertical lines of the planar chessboard would both be at infinity, and no solution would exist (Fig. 11.9). The guessed values for K here are based only on the optics of the fundus camera. Therefore, it does not include the optics of eye, but should be close enough to the values for the whole optical system.

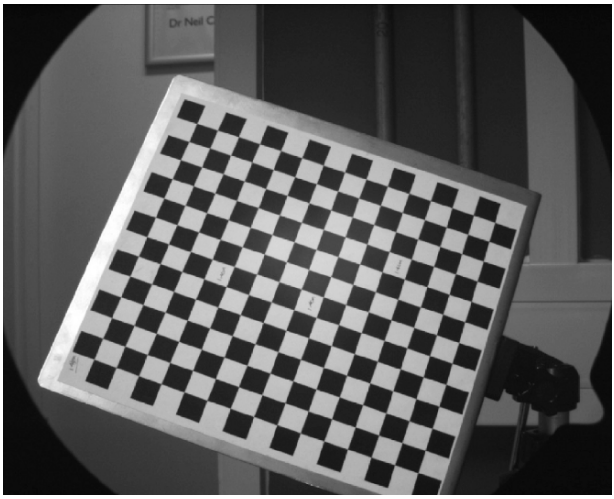


Fig. 11.8 Planar chessboard used for calibration

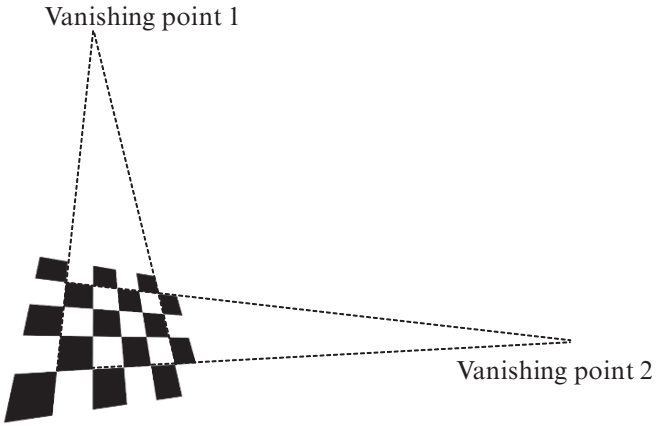


Fig. 11.9 Vanishing points of a perspective chessboard

11.2.5 Recovery of the Projection Matrix

The corresponding points $x_i \leftrightarrow x_i'$ in the 2D images and the unknown 3D points, X_i, on the object have the relationship:

$$x_i = PX_i, \quad x_i' = P'X_i \tag{11.5}$$

here P and P' are the two projective matrices. If the essential matrix E of the camera-eye system was obtained, the matrices P and P' could be retrieved from E as follows [5]:

1. Suppose the origin of the coordinate system is at the first camera centre, then the two projection matrices could be factorized as $P = K[I|0]$ and $P' = K[R|t]$, where R and t are the 3D displacements (rotation and translation) from the global coordinate system to the camera coordinate system. If the calibration matrix K was known, its inverse can be applied to the points x_i and x_i' to obtain their normalized coordinates:

$$\bar{x}_i = K^{-1}x_i = [I \mid 0]X_i, \quad \bar{x}_i' = K^{-1}x_i' = [R \mid t]X_i$$

Thus the corresponding normalized projection matrices are

$$\bar{P} = [I \mid 0], \quad \bar{P}' = [R \mid t]$$

2. The essential matrix can be calculated from Eq. (11.2), or

$$\text{expressed as } E = [t]_\times R = SR$$

where S is the skew-symmetric matrix of t.
Let the SVD of E be UDV^{T} where $D = \text{diag}(k,\ k,\ 0)$, then the possible factorization of $E = SR$ is one of the following:

$$S = UZU^T;\ R = UWV^T \text{ or } UW^TV^T$$

where

$$Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Hence the normalized projection matrix $\bar{P}'$ has four possible choices based on SVD of E as follows:

$$\bar{P}' = [UWV^T|+u_3] \text{ or } [UWV^T|-u_3] \text{ or}$$
$$[UW^TV^T|+u_3] \text{ or } [UW^TV^T|-u_3]$$

here $u_3 = U(0,0,1)^{\mathrm{T}}$, the last column of U.
3. Finally the P and P' can be calculated by

$$P = K\bar{P} \text{ and } P' = K\bar{P}'.$$

Based on the fact that reconstructed points should be in front of both cameras, the correct solution may be determined by testing a single point if it is in front of both cameras.

At the same time the extrinsic parameters, the rotation axis l and the angle of rotation λ may be obtained as:

$$l = (R_{32} - R_{23}, R_{13} - R_{31}, R_{21} - R_{12})^T \tag{11.6}$$

$$\lambda = \arccos\left(\frac{trace(R) - 1}{2}\right) \tag{11.7}$$

The projection matrix P and P' are computed according to the essential matrix E obtained above.

11.2.6 Parameter Refinement

Based on the corresponding points obtained from vessel segmentation, the extrinsic parameters R and $\boldsymbol{t}$, and the intrinsic parameter f were refined by minimizing the first-order geometric error cost function using the Levenberg-Marquardt algorithm.

$$C_r = \sum_i \frac{(x_i' \tilde{F} x_i)^2}{(\tilde{F}x_i)_1^2 + (\tilde{F}x_i)_2^2 + (\tilde{F}^T x_i')_1^2 + (\tilde{F}^T x_i')_2^2} \tag{11.8}$$

Here the fundamental matrix $\tilde{F}$ was calculated as

$$\tilde{F} = K^{-T}[t]_\times R' K^{-1} \tag{11.9}$$

where R' is the rotation matrix calculated from Rodrigues' formula [5]

$$R' = I + \sin\lambda[\frac{l}{\|l\|}]_\times + (1 - \cos\lambda)[\frac{l}{\|l\|}]_\times^2. \tag{11.10}$$

After this refinement, the final projection matrix could be obtained from R, $\boldsymbol{t}$ and K. The corresponding epipolar lines are displayed in Fig. 11.10.

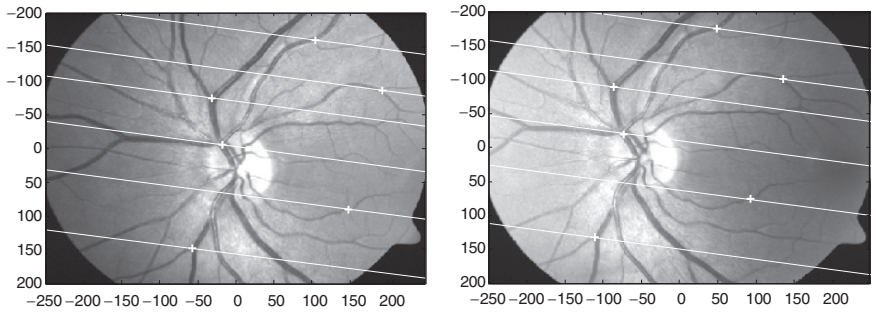


Fig. 11.10 The epipolar lines after parameter refinement

11.2.7 Reconstruction of 3D Points

Knowing the projection matrices for two images separately, the 3D coordinates of each point, X_i, can be calculated. In order to obtain a smooth reconstruction, the corresponding points from vessel segmentation were smoothed and interpolated using cubic splines.

An iterative linear method (Iterative-Eigen) [4] was used to perform the triangulation of 3D points. Equation (11.5) can be written as:

$$\begin{cases} \frac{1}{w_{i,j}}(u_i p^{3T} X_{i,j} - p^{1T} X_{i,j}) = 0 \\ \frac{1}{w_{i,j}}(v_i p^{3T} X_{i,j} - p^{2T} X_{i,j}) = 0 \\ \frac{1}{w'_{i,j}}(u'_i p'^{3T} X_{i,j} - p'^{1T} X_{i,j}) = 0 \\ \frac{1}{w'_{i,j}}(v'_i p'^{3T} X_{i,j} - p'^{2T} X_{i,j}) = 0 \end{cases} \tag{11.11}$$

where p^{iT} and p'^{iT} are the ith rows of P and P' respectively. $w_{i,j}$ and $w'_{i,j}$ are the weight factors at the jth step of iteration which have the form:

$$w_{i,j} = p^{3T} X_{i,j-1}, \quad w'_{i,j} = p'^{3T} X_{i,j-1} \tag{11.12}$$

At the beginning of the iteration $w_{i,0}$ and $w'_{i,0}$ were set to be 1 in order to find the initial solution of $X_{i,0}$.

All calculations were performed using Matlab (The Mathworks). The parameter refinement was based on the codes from the Oxford Brookes toolbox (http://cms.brookes.ac.uk/staff/PhilipTorr/). The flowchart of a complete reconstruction process is shown in Fig. 11.11.

11.3 Experiment and Results

A cylinder with a chessboard attached to its external surface was adopted to understand and obtain a preliminary validation of the metric projective reconstruction procedure. The focal length and aperture of the camera (Nikon D50) were fixed when pictures of the chessboard were taken, which means that the intrinsic parameters of the camera optical system were kept constant. Four images were acquired and used (Fig. 11.12 left) for self-calibration. The corners of the chessboard were extracted automatically for reconstruction (Fig. 11.12 right). The epipolar geometry between the image pairs was calculated by the normalized 8-points algorithm. After self-calibration, the 3D coordinates of the corners of the chessboard were calculated by the Iterative-Eigen triangulation (Fig. 11.13). Although the final triangulation of the 3D object coordinates was only based on two images, the self-calibration could utilize more images to improve the accuracy of the estimated camera intrinsic parameters. Quantitative comparison between the measurements and reconstruction results is summarized in Table 11.1. It is clear that with good estimation of the

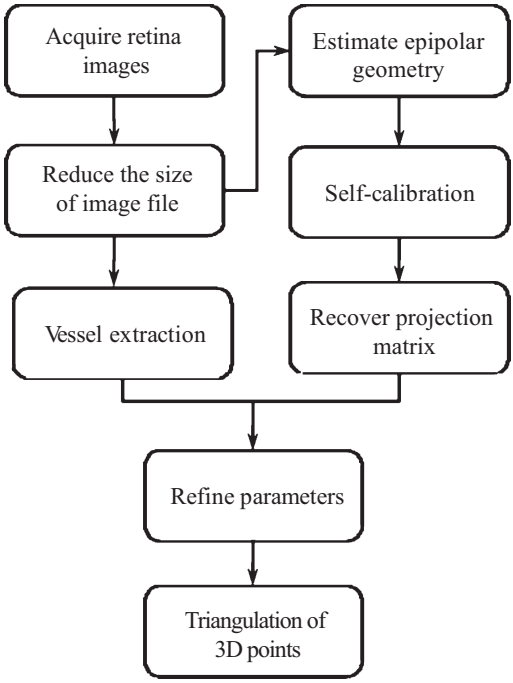


Fig. 11.11 The reconstruction process

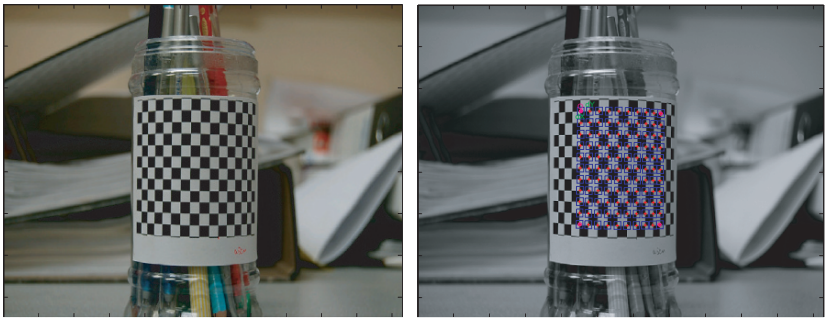


Fig. 11.12 Object image (left) and image with extracted corners (right)

epipolar geometry and self-calibration, the reconstruction procedure implemented here is capable of recovering very well most of the geometric features such as height to length ratio (a/b), angles (α and β) and curvature (r/b). The percentage error in curvature (r/b) seems to be larger than that in the other parameters, possibly due to the uncorrected distortion in the images and the smaller value for r making it more sensitive to errors.

Figure 11.14 shows the reconstruction results of the segmented vessels in the central region of the image presented in Fig. 11.5. Figure 11.14a is the 2-D view of

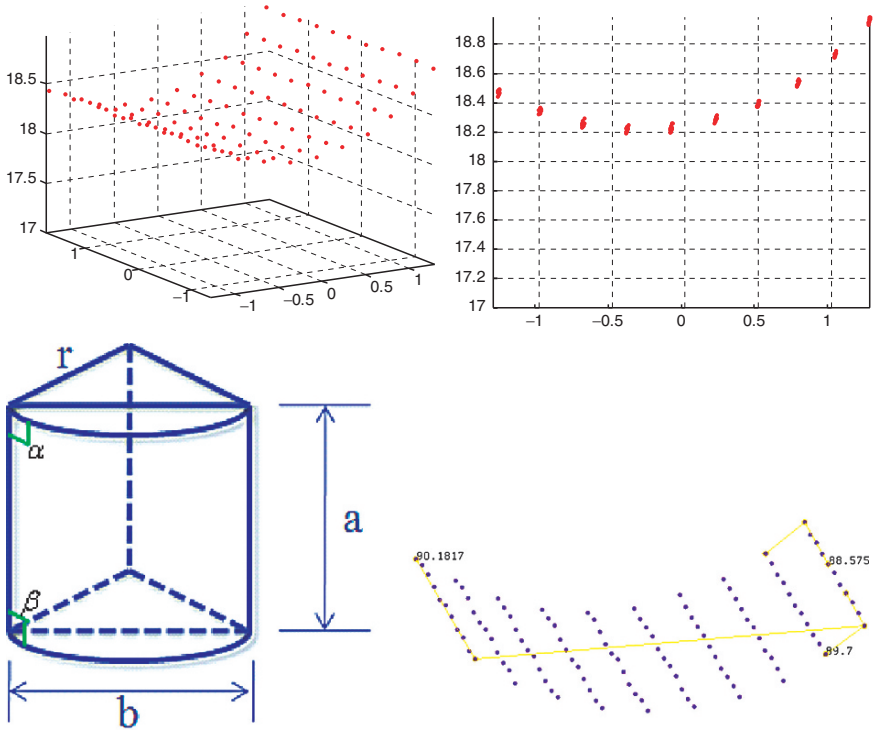


Fig. 11.13 Three-dimensional view of the reconstruction result (top left) and its side view (top right). Geometry model of the cylindrical chessboard (bottom left) and reconstruction result with angle measurements in ICEM (bottom right)

Table 11.1 Quantitative comparison of the reconstructed results and measurements. See Fig. 11.13 for definition of geometric parameters

	Measured (M)	Reconstructed (R)	Error $= \lvert(M-R)/M\rvert^*100\%$
a/b	1.1951	1.1778	1.45%
r/b	0.8415	0.7771	7.66%
α	90	89.7	0.3%
β	90	88.575	1.58%

the recovered centreline points of the marked 3-D vessels. The numbers (according to Fig. 11.5) indicate which segments of the marked vessels are retrieved. Figure 11.14b is the 3-D view of the reconstructed vessel centrelines. Figure 11.14c and d show the side projections of the centrelines of the reconstructed vessels. These preliminary results demonstrate that 3D reconstruction of the retinal vessels is feasible and may be useful in future studies of the retinal microvasculature in health and disease. Future studies will attempt to validate the 3-D reconstruction ex vivo using a model eye with known geometry and in vivo using ultrasound to measure geometric features.

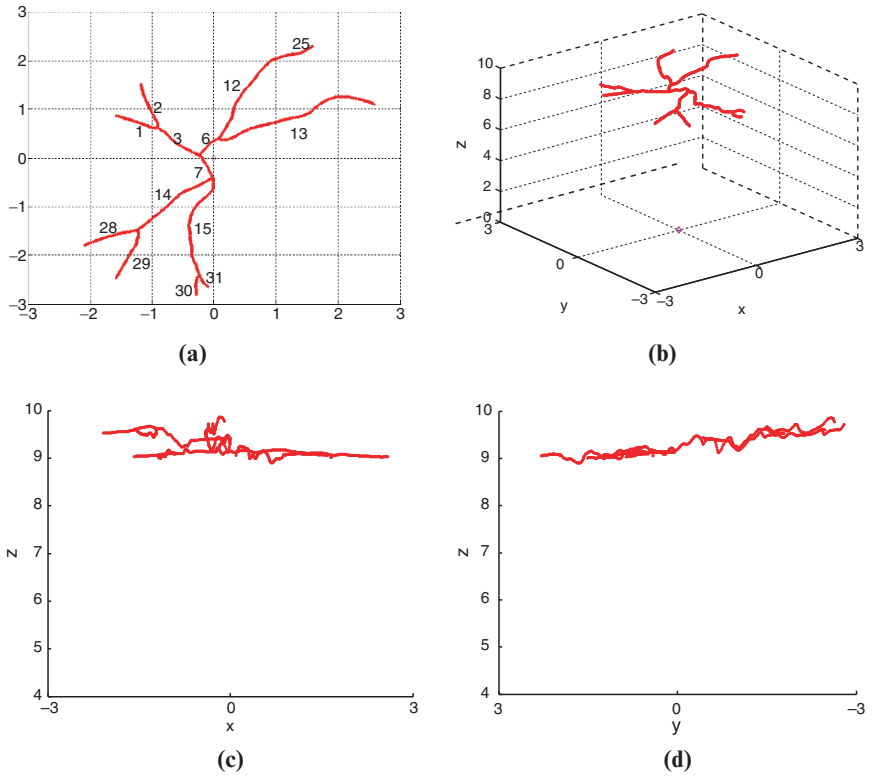


Fig. 11.14 Three-dimensional reconstruction results of the centre-lines of the marked retinal vessels: (**a**) top view of the reconstructed vessels; (**b**) 3D view; (**c**), (**d**) side projections

References

1. Dick A, Torr P, Cipolla R (2000) Automatic 3d modelling of architecture. In: Proceedings of 11th British Machine Vision Conference (BMVC'00), Bristol, pp 372–381.
2. Fusiello A (2001) A new autocalibration algorithm: experimental evaluation. Computer analysis of Images and Patterns 2001, Lecture Notes in Computer Science 2124: 717–724.
3. Hartley RI (1997) In defense of the eight-point algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 19: 580–595.
4. Hartley RI, Sturm P (1997) Triangulation. Computer Vision and Image Understanding 68(2): 146–157.
5. Hartley R, Zisserman A (2003) Multiple View Geometry in Computer Vision. Cambridge: Cambridge University Press.
6. Kai Z, Xu X, Zhang L, Wang GP (2005) Stereo matching and 3-D reconstruction for optic disk images. CVBIA, LNCS 3765: 517–525.
7. Liu Y, Wu YX, Wu MP, Hu XP (2004) Planar vanishing points based camera calibration. In: Proceedings of the Third International Conference on Image and Graphics (ICIG'04), Hong Kong, China, pp 460–463.

8. Martinez-Perez ME, Hughes AD, Stanton AV, Thom SA, Chapman N, Bharath AA, Parker KH (2002) Retinal vascular tree morphology: a semi-automatic quantification. IEEE Transactions on Biomedical Engineering 49(8): 912–917.
9. Espinosa-Romero A, Martinez-Perez ME (2005) Optical 3D reconstruction of retinal blood vessels from a sequence of views. In: Proceedings of SPIE 5776, Merida, Mexico, pp 605–612.
10. Masters B (2004) Fractal analysis of the vascular tree in the human retina. Annual Review of Biomedical Engineering 6: 427–452.
11. Mendonca PRS, Cipolla R (1999) A simple technique for self-calibration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, pp 500–505.
12. Stanton AV, Wasan B, Cerutti A, Ford S, Marsh R, Sever PP, Thom SA, Hughes AD (1995) Vascular network changes in the retina with age and hypertension. Journal of Hypertension 13:1724–1728.
13. Wong TY, McIntosh R (2005) Systemic associations of retinal microvascular signs: a review of recent population-based studies. Ophthalmic and Physiological Optics 25: 195–204.
14. Xu J, Chutatape O (2003) Comparative study of two calibration methods on fundus camera. In: Proceedings of the 25 Annual International Conference of the IEEE EMBS, Cancun, Mexico, pp 17–21.
15. Xu J, Chutatape O (2006) Auto-adjusted 3D optic disk viewing from low-resolution stereo fundus image. Computers in Biology and Medicine 36: 921–940.
16. Zhang Z (1999) Flexible camera calibration by viewing a plane from unknown orientations. In: Proceedings of the International Conference on Computer Vision, Corfu, Greece, pp 666–673.

Chapter 12
Microscale Flow Dynamics of Red Blood Cells in Microchannels: An Experimental and Numerical Analysis

R. Lima, M. Nakamura, T. Omori, T. Ishikawa, S. Wada, and T. Yamaguchi

Abstract The blood flow dynamics in microcirculation depends strongly on the motion, deformation and interaction of red blood cells (RBCs) within the microvessel. We present confocal micro-PTV measurements on the motion of individual RBCs through a circular polydimethysiloxane (PDMS) microchannel. The RBC radial displacement and dispersion calculated from these measurements show that the RBC paths are strongly dependent on the both Hct and plasma layer. In order to obtain more detailed information of the non-Newtonian property of blood a novel computational scheme is also described. The simulated flow dynamics were in good agreement with the Casson flow model and *in vivo* observations. In the near future by comparing both results we hope to clarify a variety of complex phenomena occurring at the microscale level.

12.1 Introduction

Approximately, the half volume of the blood is composed of red blood cells (RBCs) which is believed to strongly influence its flow properties. Blood flow in microvessels depends strongly on the motion, deformation and interaction of RBCs. Several experimental studies on both individual and concentrated RBCs have already been performed in the past [3, 6–8]. However, all studies used conventional microscopes

R. Lima, T. Omori, T. Ishikawa, and T. Yamaguchi
Department of Bioengineering & Robotics, Graduate School Of Engineering, Tohoku University, 6-6-01 Aoba, 980-8579 Sendai, Japan

R. Lima
Department of Mechanical Technology, ESTiG, Bragança Polytechnic, C. Sta. Apolonia, 5301-857 Bragança, Portugal

M. Nakamura and S. Wada
Department of Mechanical Science and Bioengineering, Graduate School of Engineering, Osaka University, Toyonaka, 560-8531 Osaka, Japan

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

and also ghost cells to obtain visible trace RBCs through the microchannel. Recently, considerable progress in the development of confocal microscopy and consequent advantages of this microscope over the conventional microscopes have led to a new technique known as confocal micro-PIV [13,15,18,21]. This technique combines the conventional PIV system with a spinning disk confocal microscope (SDCM). Due to its outstanding spatial filtering technique together with the multiple point light illumination system, this technique has the ability to obtain in-focus images with optical thickness less than 1 μm.

In a numerical context, blood flow in large arteries is usually modeled as a continuum however this assumption is not valid in small vessels such as arterioles and capillaries. In this way, we are developing an integrative multi-scale model to simulate the blood flow at mesoscopic level. This computational approach may provide important information on the rheology of blood in small vasculatures where non-Newtonian property of blood is not negligible.

The main purpose of this paper is to measure flow behavior of individual RBCs at different haematocrits (Hct) through a 75 μm circular polydimethysiloxane (PDMS) microchannel by means of confocal micro-PTV system. Moreover we introduce an integrative multi-scale model to simulate the blood flow behavior through microvessels in order to obtain more detailed insights about the blood rhelogical properties at cellular level.

12.2 Confocal Micro-PTV Measurements of RBCs

12.2.1 Materials and Methods

12.2.1.1 Working Fluids and Microchannel

Four working fluids were used in this study: dextran 40 (Dx40) containing about 3% (3 Hct), 13% (13 Hct), 23% (23 Hct) and 37% (37 Hct) of human RBCs. The blood was collected from a healthy adult volunteer, where ethylenediaminetetraacetic acid (EDTA) was added to prevent coagulation. The RBCs where separated from the bulk blood by centrifugation (1,500 RPM for 5 min) and aspiration of the plasma and buffy coat and then washed twice with physiological saline (PS). The washed RBCs were labeled with a fluorescent cell tracker (CM-Dil, C-7000, Molecular Probes) and then diluted with Dx40 to make up the required RBCs concentration by volume [10, 12]. All blood samples were stored hermetical at 4 °C until the experiment was performed at controlled temperature of about 37 °C. All procedures in this experiment were carried out in compliance with the Ethics Committee on Clinical Investigation of Tohoku University.

By using a soft lithography technique it is possible to fabricate easily precise and reproducible rectangular microchannels at low cost. Rectangular PDMS microchannel with a low aspect ratio seems to be appropriate to perform confocal micro-PIV measurements of *in vitro* blood [14]. However, this geometry may not reflect the

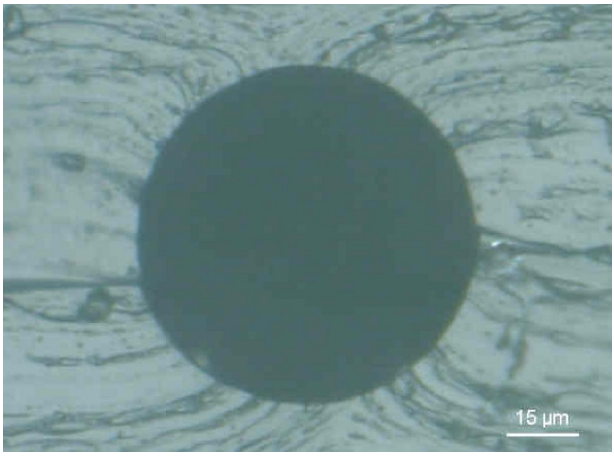


Fig. 12.1 Cross section of the PDMS microchannel

actual physiology of the microcirculation. Hence, very recently we have successfully fabricated straight circular PDMS microchannels by using a wire casting technique [12]. The microchannel used in this study was a PDMS circular microchannel with a diameter of 75 µm (Fig. 12.1).

12.2.1.2 Confocal Micro-PTV Experimental Set-Up

The confocal micro-PIV system used in our experiment consists of an inverted microscope (IX71, Olympus, Japan) combined with a confocal scanning unit (CSU22, Yokogawa, Japan) and a diode-pumped solid state (DPSS) laser (Laser Quantum Ltd, UK) with an excitation wavelength of 532 nm. Moreover, a high-speed camera (Phantom v7.1, Vision Research, USA) was connected into the outlet port of the CSU22 (see Fig. 12.2). The PDMS microchannel was placed on the stage of the inverted microscope where the flow rate of the working fluids was kept constant $(\mathrm{Re} \sim 0.004)$ by means of a syringe pump (KD Scientific Inc., USA). A thermo plate controller was set to 37 °C. All the confocal images were captured in the middle of the microchannels with a resolution of 640×480 pixels, at a rate of 100 frames/s with an exposure time of 9.4 ms. The recorded images were transferred to the computer and then evaluated in Image J (NIH) [1] by using a manual tracking MTrackJ [16] plugin. As a result it was possible to track single RBCs through the middle plane of the PDMS microchannel.

12.2.1.3 RBC Radial Displacement and Radial Dispersion Coefficient

The radial displacements $(\Delta \mathrm{R})$ of the tracked RBCs were determined by using the following equation:

$$\Delta R(t) = |R(t_0 + t) - R(t_0)| \tag{12.1}$$

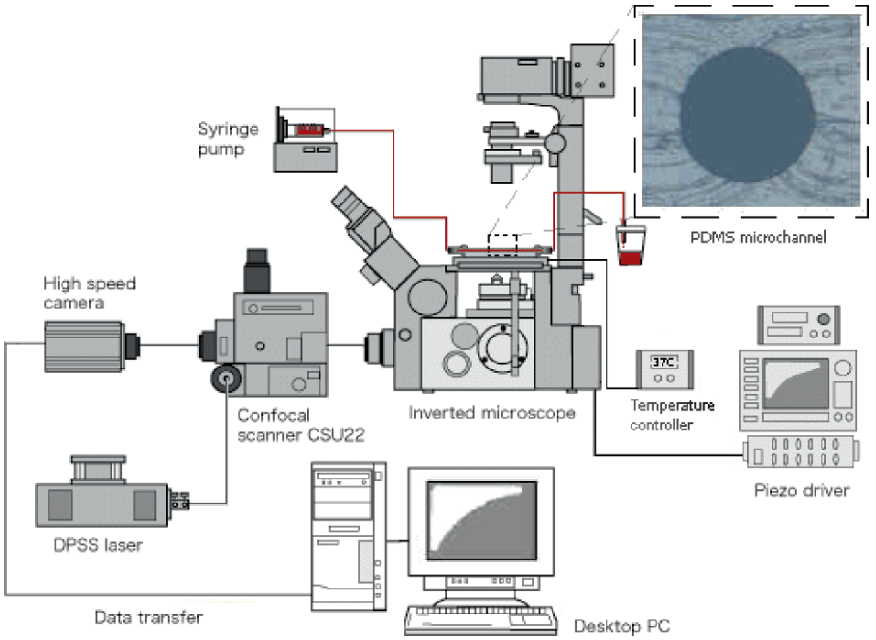


Fig. 12.2 Experimental set-up

where R(t) is the radial position at time t, t_0 is the initial time, and $\Delta R(t)$ is the radial displacement at time interval t.

Besides the calculation of the radial displacement (ΔR), we have also analyzed the motions of RBCs by using a radial dispersion coefficient (D_{yy}) [8, 9], given by:

$$D_{yy}(t) = \frac{1}{N}\sum_{i=1}^{N}\frac{\left\langle (R_{i,y}(t) - R_{i,y}(0))^2 \right\rangle}{2t} \tag{12.2}$$

where R, t, N are the radial displacement, time interval and number of RBCs respectively.

12.2.2 Results and Discussion

12.2.2.1 Blood Flow Visualization at Several Hcts

Figure 12.3 shows images with both non-labeled RBCs (halogen illumination) and labeled RBCs (laser-emitted light) measured in centre plane of 75 µm circular PDMS microchannel at several Hcts (from 3% to 37%) with Re from 0.004 to 0.005.

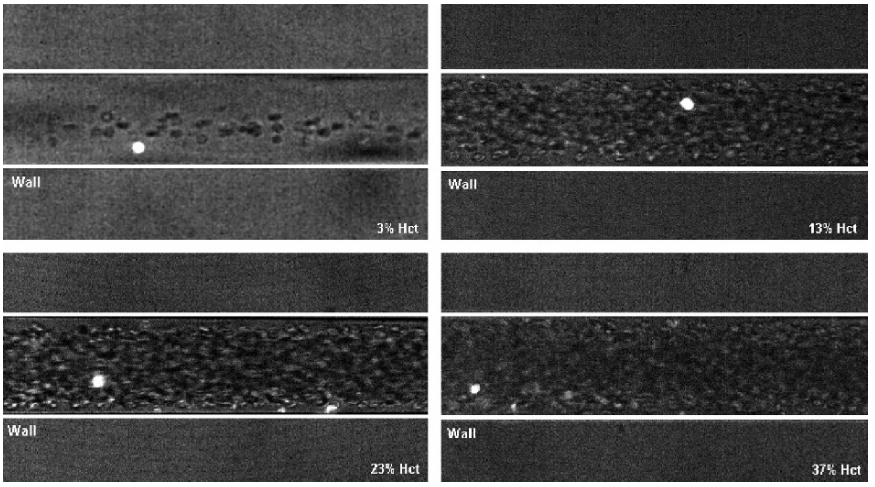


Fig. 12.3 Both normal and labeled rbcs (bright spots) with 3% Hct, 13% Hct, 23% Hct, 37% Hct (20×, 1.6 zoom)

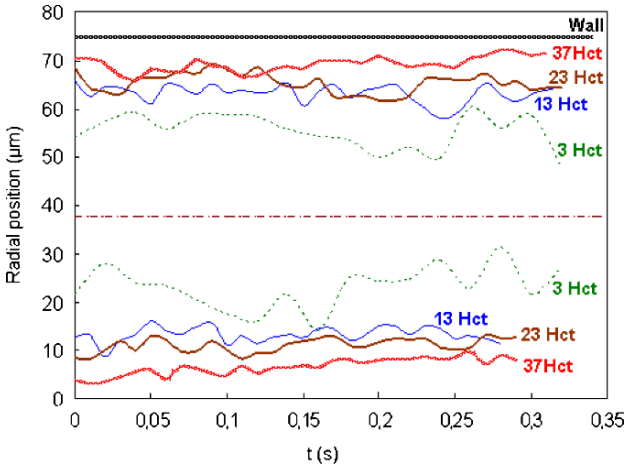


Fig. 12.4 The effect of the Hct on the plasma layer

From Fig. 12.3 it is possible to observe that the Hct decreases with the microchannel diameter which corroborates with the well known Faharaeus effect [3]. Moreover, it is also evident that the plasma layer tends to increase by decreasing the Hct (see Figs. 12.4 and 12.5). This latter phenomenon is related to the Fahareues-Lindqvist effect [5]. Although this phenomenon is still not completely understood [2], the most acceptable explanation is related to the tendency of the RBCs to migrate toward the microtube axis enhanced by the RBCs deformation and interations [6, 19]. A further consequence is that the apparent blood viscosity is reduced so that the flow resistance through the capillary also decreases.

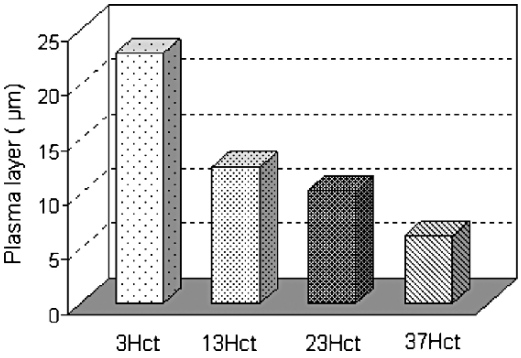


Fig. 12.5 Average thickness of the plasma layer at several Hcts

12.2.2.2 Radial Displacement in a 75 μm PDMS Microchannel

Figure 12.6a shows the effect of the radial position on the motion of labeled RBCs flowing through a 75 μm PDMS capillary, for a feed Hct of about 13%. In addition Fig. 12.6b shows the radial displacement of a RBC flowing through microchannel with Hct of about 3%. The correspondent radial displacements (ΔR) are shown in Fig. 12.7.

From Fig. 12.7 it is clear that the RBCs radial displacement (ΔR) for 13% Hct is higher than a RBC flowing within an Hct of 3%. In addition it is also possible to observe that the radial displacement (ΔR) of the RBC flowing close to the plasma layer (12 μm from the wall) is around three times higher than the RBC (36 μm from the wall) traveling around the middle of the microchannel. These results suggest that RBCs flowing within the boundary region of RBC core appear to undergo the largest radial displacements. We believe that the random like transverse motions happening in this region are mainly due to multiple hydrodynamic interactions with neighboring RBCs which flow with lower velocity adjacent to the wall or tend to migrate away from the microtube wall towards the RBC core region.

12.2.2.3 Radial Dispersion in 75 μm PDMS Microchannel at Several Hcts

By measuring the radial displacement of labeled RBCs flowing through the microchannel for a known time interval, it was possible to calculate the correspondent dispersion coefficient (D_{yy}). Figure 12.8 shows the RBC averaged dispersion coefficient at the middle plane (D_{yy}) for several Hcts (3% Hct, 13% Hct, 23% Hct and 37% Hct).

Figure 12.8 shows that the radial dispersion coefficient (D_{yy}) increases with the Hct. The RBC D_{yy} for Hcts from 23% to 37% have almost one order of magnitude greater than the D_{yy} with 3% Hct. These results clearly reflect the RBCs radial displacement obtained in Fig. 12.7.

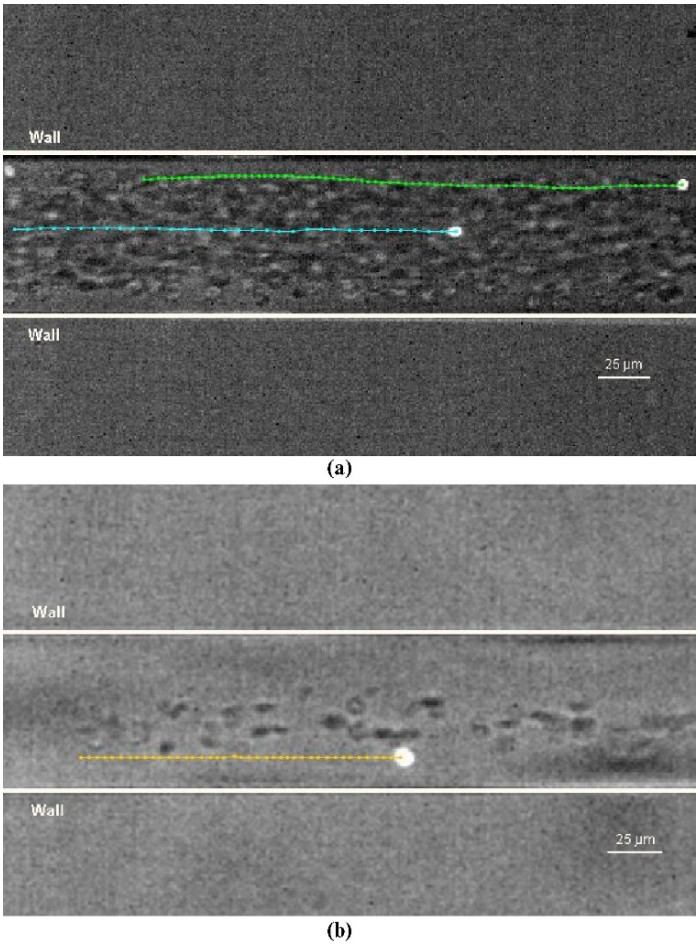


Fig. 12.6 Rbcs streamlines at several haematocrits: (**a**) 13% Hct, (**b**) 3% Hct

Generally, our results demonstrate that the RBCs at dense concentrations exhibit higher erratic radial displacement when compared with diluted suspensions of RBCs. Although it is evident that the RBC radial dispersion rises with the increase of Hct at Hcts of about 24% it tends to level off (see Fig. 12.9). Note that, these observations are consistent with several other measurements performed in glass microchannels [8, 11].

The results from Fig. 12.10 reinforce our previous measurements obtained from glass capillaries [11]. These data clearly demonstrate that RBC radial dispersion tend to decrease with the diameter. We believe that the main reason is due not only to Hct reduction with the diameter (Faharaeus effect) but also to the geometry constriction which limits the amplitude of the RBCs radial displacements.

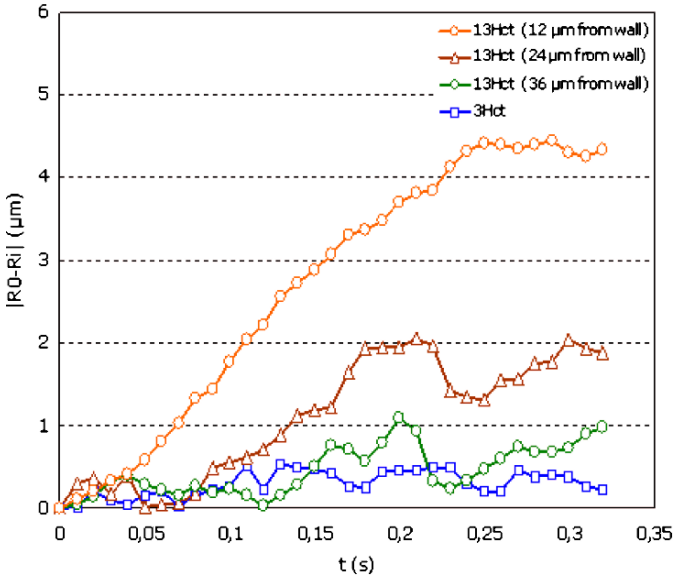


Fig. 12.7 Radial displacement (ΔR) of labeled rbcs at several haematocrits (3% Hct and 13% Hct) and at different radial positions

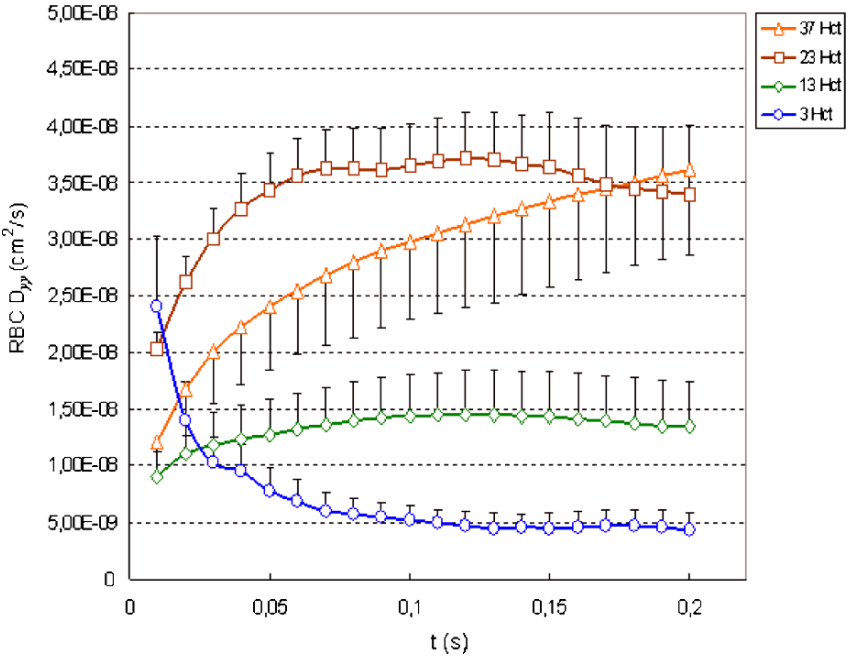


Fig. 12.8 RBC averaged dispersion coefficient at the middle plane (D_{yy}) for several Hcts: 3% Hct, 13% Hct, 23% Hct and 37% Hct

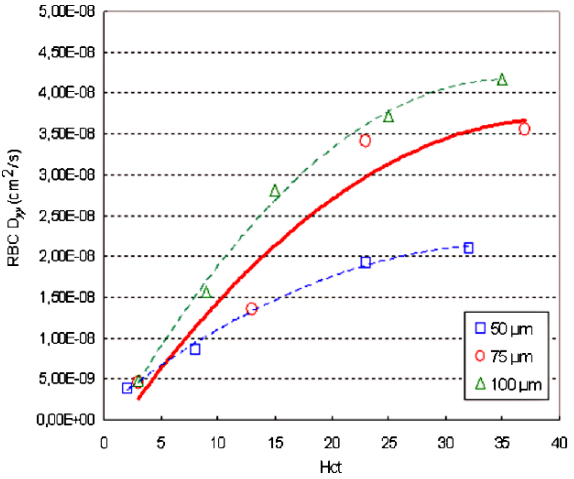


Fig. 12.9 Effect of the Hct on the RBC D_{yy} at 75 μm PDMS microchannel and 100 μm, 50 μm glass capillary [11]

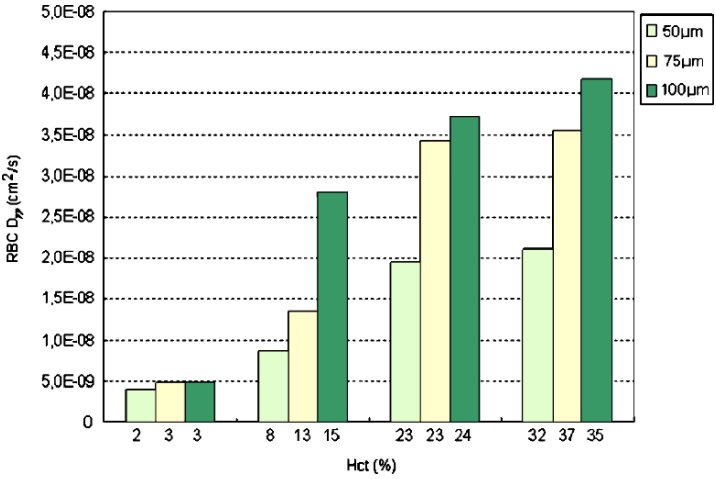


Fig. 12.10 The effect of the microchannel diameter on the RBC D_{yy}

12.3 Numerical Flow Model of Multiple RBCs

A simulation method for multiple RBCs in a micro-vasculature was proposed for understanding the rheological properties of blood from a viewpoint of multiscale mechanics. In the following context, blood flow is modeled at two different scales. A micro-scale flow is modeled as a particulate flow of RBCs. On the other hand, flow at a macro-scale is modeled as a continum expressed by the equations of continuity and Navier-Stokes. Then, a coupling method between those differently-scaled blood flow models is described. Finally, the simulation results are presented.

12.3.1 Materials and Methods

12.3.1.1 Microscale Blood Flow Model (RBC Flow Model)

An RBC model developed by Wada and Kobayashi [22] was adopted. This model is capable of expressing a biconcave shape of the RBC at rest as well as its elastic deformation while in motion. As shown in Fig. 12.11, the model was constructed by surrounding the internal liquid of RBC with RBC membrane consisting of N nodal points linked by a spring element.

Owing to deformation, elastic energies are generated and stored in RBC. The stretching energy W_s and bending energy W_b are modeled as

$$W_s = \frac{1}{2} k_s \sum_{l=1}^{N} (L_l - L_{l0})^2 \tag{12.3}$$

$$W_b = \frac{1}{2} k_b \sum_{l=1}^{N_l} L_l \tan^2 \left(\frac{\theta_l}{2} \right) \tag{12.4}$$

where k_s and k_b are spring constants, N, N_1 are the number of nodes and lines, L_{l0}, L_l are length of spring at the natural state and after deformation, θ_l is the contacting angle between neighboring elements. To maintain the total area of RBC (actually it is volume in 3D), a penalty functions W_A was introduced. Mathematically, it was defined by

$$W_A = \frac{1}{2} k_A \left(\frac{A - A_0}{A_0} \right)^2 A_0, \tag{12.5}$$

where subscript 0 denotes the natural state, N_e is the number of bending springs, k_a is a coefficient for the area constraint.

An interactive force working between two RBCs which come closer are expressed by a potential function Ψ assigned at each nodal point on the RBC membrane. The potential function Ψ for nodal point i is a summation of interactive forces from all neighboring nodal points as formulated by

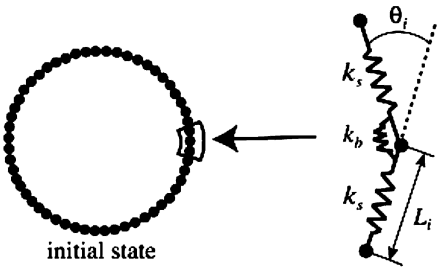


Fig. 12.11 Red blood cell model. Nodal points on the membrane were connected by a mechanical spring. The neighboring springs were also linked by a bending spring

$$\Psi = \sum_{j=1}^{N} \Psi_{ij} \tag{12.6}$$

where Ψ_{ij} is the potential function on nodal point i from nodal point j on neighboring RBC. It is given by

$$\Psi_{ij} = \begin{cases} k_r \left(\frac{\pi z_{ij}}{2} - \tan\left(\frac{\pi z_{ij}}{2} \right) \right) & for \quad -1 \le z_{ij} \le 0 \\ 0 & for \quad 0 \le z_{ij} \end{cases} \tag{12.7}$$

where $z_{ij} = d_{ij}/\delta - 1$, d_{ij} is a distance between nodal points i and j, and δ is equilibrium distance. According to this function, a repulsive force acts when they are come much closer. In the present simulation, the equilibrium distance δ is set to L_0 which is natural length of a spring.

An interactive force working between an RBC and a vessel wall is modeled in a way similar to the interactive force between two RBCs. The potential function Z for nodal point i is expressed by a summation of interactive forces from all neighboring nodal points as formulated by

$$\mathrm{Z} = \sum_{i=1}^{N} \mathrm{Z}_i \tag{12.8}$$

where Z_i is the potential function on nodal point i from nodal point j on neighboring RBC, which is given by

$$Z_i = \begin{cases} k_w \left(\frac{\pi z_i}{2} - \tan\left(\frac{\pi z_i}{2} \right) \right) & for -1 \le z_i \le 0 \\ 0 & for \quad 0 \le z_i \end{cases} \tag{12.9}$$

where $z_i = D_i/\delta_w - 1$, D_i is the distance between nodal point i and the wall, k_w is a parameter to express the magnitude of a repulsive force, and δ_w is an equilibrium distance. Here, δ_w was set to the same as δ.

Modeling of a fluid force $\mathbf{f}_i$ working on an RBC is made separately for forces $\mathbf{f}_{ni}$ and $\mathbf{f}_{ti}$ which respectively work in normal and tangential directions to a line element i of an RBC membrane. On the basis of the conservation of momentum, the normal force, $\mathbf{f}_{ni}$ working on a line element i of the RBC membrane is modeled as

$$\mathbf{f}_{ni} = \rho L_i \mathbf{v}_{ni}^2 \tag{12.10}$$

where ρ is a density of plasma, and $\mathbf{v}_{ni}$ is a normal component of the velocity difference between an RBC and a plasma flow. The tangential force $\mathbf{f}_t$ is modeled based on Newton's law of viscosity. If we assume that a flow velocity at a distance Δ from an RBC membrane is the same as a flow velocity at the RBC membrane, the tangential force $\mathbf{f}_{ti}$ that works on a line element i can be approximated as

$$\mathbf{f}_{ti} = \mu_p L_i \, \mathbf{v}_{ti}/\Delta \tag{12.11}$$

where μ_p is a viscosity of plasma, $\mathbf{v}_{ti}$ is a tangential component of $\mathbf{v}_i$. The equivalent distance Δ in Eq. (12.11) was estimated from Oseen flow theory where a

two-dimensional cylinder with a radius of a falls down in a viscous fluid. The distance Δ is therefore

$$\Delta = \left(2.0\left(\ln\frac{8.0}{\mathrm{Re}} - \gamma\right) + 1.0\right)\frac{a}{8.0} \tag{12.12}$$

where Re is the Reynolds number determined from a relative velocity of the cylinder to a fluid and a diameter of the cylinder and γ is Euler constant $(= 0.57721)$. In the present study, a is set to be comparable to the size of an RBC. Note that the fluid force is assumed not to work when a distance between two RBCs is less than δ for presuming the situation that they are in touch.

Given elastic energies and fluid forces, the motion of nodal point i placed on RBC membrane was determined from

$$m\ddot{\mathbf{r}}_i + \gamma(\dot{\mathbf{r}}_i - \dot{\mathbf{r}}_g) = \mathbf{F}_i + \mathbf{f}_i \tag{12.13}$$

where a dot means a time derivative, $\mathbf{r}_i$ is the position vector of nodal point i, m is mass of the nodal point, and $\mathbf{f}_i$ is a fluid force $(= \mathbf{f}_{ti} + \mathbf{f}_{ni})$. Based on the virtual work theory, an elastic force $\mathbf{F}_i$ is given by

$$\mathbf{F}_i = \frac{\partial W}{\partial r_i} \tag{12.14}$$

$$W = W_s + W_b + W_a + \Psi + \mathrm{Z}. \tag{12.15}$$

12.3.1.2 Macroscale Blood Flow Model (Continuum Flow Model)

A macroscale blood flow at a steady state is modeled as a continuum expressed by the Navier-Stokes and continuity equations;

$$u_j\frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\frac{1}{\mathrm{Re}}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] \tag{12.16}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{12.17}$$

where u_i is a flow velocity component and p is a pressure. In general, fluid viscosity is assumed to be spatially constant for the analysis of arterial blood flow. However, it is not true in small arteries due to spatial variation of RBC concentration. We here introduce a Hematocrit function $F(\mathrm{Hct})$ to the viscous term of the Navier-Stokes equation in order to express a local viscosity dependent on a local hematocrit.

Blood flow is often represented by Casson model. Shiga et al. [20] obtained an empirical formula of the relationship between Casson viscosity η_c and hematocrit Hct as

$$\ln\left(\frac{\eta_c}{\eta_p}\right) = k \cdot \mathrm{Hct} \tag{12.18}$$

where η_p is a plasma viscosity and k is a constant. Based on this equation, we express the hematocrit function F as

$$F = \frac{\eta_c}{\eta_p} = \exp\left[k \cdot \mathrm{Hct}\right], \tag{12.19}$$

whereby the Navier-Stokes equation is rewritten as

$$u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\frac{\exp\left[k \cdot \mathrm{Hct}\right]}{\mathrm{Re}}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] \tag{12.20}$$

In the actual simulation of a macroscale flow, we solve Eqs. (12.17) and (12.20) by a finite element method.

12.3.1.3 Geometry Model of a Small Vasculature

The present study solved blood flowing between a two-dimensional parallel plate flow channel. The x-axis and y-axis were respectively set in axial and radial directions of the channel. For the microscale simulation, 108 RBCs were put in the channel with height of 96 μm and length of 44 μm. In this case, Hct is 0.31. A periodic boundary condition was assumed for both x-ends of the channel. For the macroscale flow analysis, the channel is extended to 480 mm. The flow region was divided into triangular finite elements. The total numbers of nodal points and elements were 2,500 and 4,752, respectively.

12.3.1.4 Simulation Condition

Table 12.1 encapsulates the parameter of the microscale flow. The Reynolds number was 0.6. Parameter k in Eq. (12.19) was set to 2.85 based on the viscosity of the whole blood of 4.55 mPa · s and that of the plasma of 1.33 mPa · s at Hct of 0.44, a shear rate of 230 s^{-1}, and temperature of 37 °C.

Table 12.1 Parameters used for the simulation

Number of nodes, N	60
Mass, m	1.0×10^{-15} kg
Spring constant for stretching, k_s	1.0×10^{-17} J
Spring constant for bending, k_b	1.0×10^{-18} J
Spring constant for the area constraint, k_a	1.0×10^{-16} J
Viscosity of the red blood cell membrane, γ	1.0×10^{-10} kg/s

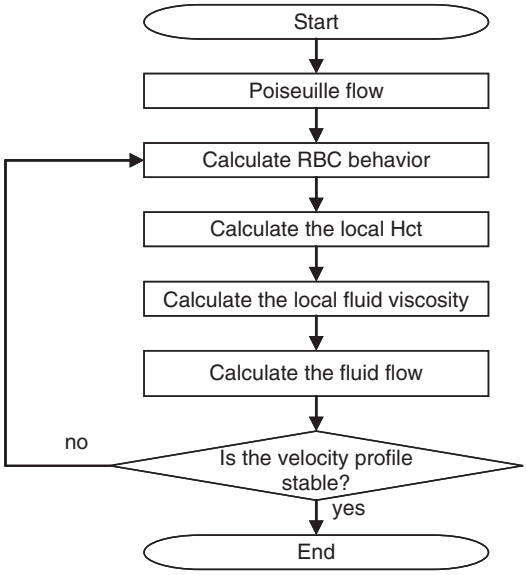


Fig. 12.12 Flowchart of the present simulation

12.3.1.5 Simulation Procedure

A flowchart of the simulation is shown in Fig. 12.12. Assuming with Poiseuille flow as an initial axial flow, the microscale flow is calculated. When the radial distribution of RBCs becomes stable, the calculation is stopped, and the Hct functions are obtained for flow segments defined by radially subdividing the flow channel by N to incorporate the influence of the local variation of an RBC distribution into the macroscale flow analysis. The macroscale flow simulation is performed until a fully-developed axial velocity profile is obtained. With this velocity profile, the simulation is back to the microscale simulation. This process is repeated until a change in the axial velocity profile after the microscale flow simulation is negligible. In the present simulation, $N = 8$.

12.3.2 Result

The results showed a drastic change in the distribution of RBCs with progress of the simulation. The spatial distributions of RBCs at the initial state and the converged state are shown in Fig. 12.13. On the right of each figure, the percentage of RBCs in each flow segment at each state is plotted. At the initial state, RBCs were distributed almost uniformly in a radial direction and were found near the wall the initial state. As RBCs were carried downstream by a fluid flow, they inclined and migrated to the center of the flow channel. As a result, RBCs were concentrated around the center and barely found near the wall, forming a plasma layer.

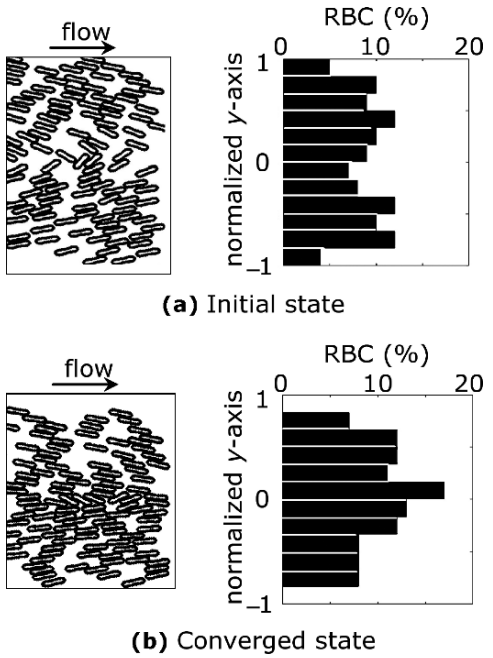


Fig. 12.13 Comparison of RBC distribution between the initial state (**a**) and the converged state (**b**)

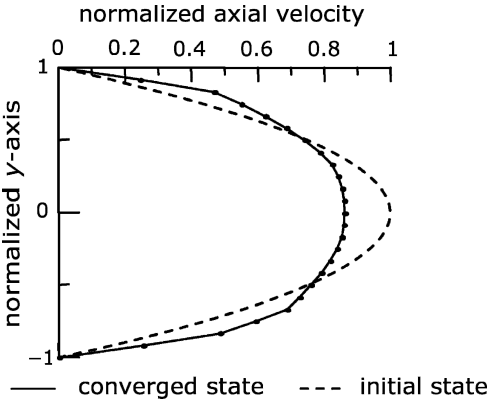


Fig. 12.14 Comparison of the axial velocity profile between the initial and converged states

The axial velocity profiles at the initial and converged states are depicted in Fig. 12.14. At the initial state, the velocity profile was parabolic as given. With progress of the simulation, there was a tendency that velocity at the center of the flow channel decreased while that near the wall increased. At the converged state, the axial flow took a rather flat velocity profile as seen in Fig. 12.14. The maximal velocity at the converged state decreased by 13.7% compared to that of the Poiseuille flow.

12.3.3 Discussion

A blood flow property in a small vasculature was analyzed by interactively simulating macro- and micro-scale blood dynamics. The results of the microscale flow simulation showed that RBCs gathered around the center of the flow channel and a plasma layer near the wall at the converged state. On the other hand, the macroscale flow showed a flat velocity profile. These flow features are quite similar to the in vivo observations [4, 17]. It is therefore considered that the simulation results represent the process of how the microscopic flow influences the macroscopic flow.

It is quite obvious that RBC behavior induced a flat velocity profile of the macroscale flow and *vice versa*. With an axial migration of RBCs, the RBC concentration became higher around the center of the channel while that near the wall became less, bringing about an increase in blood viscosity around the center and the decrease near the wall, respectively. As a consequence, the flow velocity around the center of the channel decreased and that near the wall increased, developing into a flat velocity profile.

12.4 Conclusions

Quantitative description of the flow behavior of labeled RBCs in both diluted and concentrated suspensions were studied under a confocal micro-PTV system. The experiments were performed in the middle plane of 75 μm circular PDMS microdevice at low Reynolds numbers ($\mathrm{Re} \sim 0.004$) by using Hcts from 3% up to 37%. Our experimental results suggest that the RBC paths are strongly dependent on the Hct and as a result both RBC radial displacement and RBC radial dispersion increase with the heamatocrict. Moreover, our results also indicate that RBCs flowing around the plasma layer appear to undergo the largest radial displacements. The confocal micro-PTV system used in the present work is proved to be a powerful technique to measure the motion of labeled RBCs at different Hcts through a circular PDMS microchannel.

A novel computational scheme for the analysis of the mesoscopic blood rheology was proposed. The scheme was applied for the analysis of blood flow in a small vasculature. The simulated flow dynamics were in good agreement with the Casson flow model and *in vivo* observations. These results addressed the potential of the present computational approach to the analysis of the rheology of blood in small vasculatures where non-Newtonian property of blood is significant.

By comparing both results we hope in the near future to clarify a variety of complex phenomena occurring at the microscale level.

Acknowledgements This study was supported in part by the following grants: International Doctoral Program in Engineering from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT), “Revolutionary Simulation Software (RSS21)” next-generation IT program of MEXT; Grants-in-Aid for Scientific Research from MEXT and JSPS Scientific

Research in Priority Areas (768) "Biomechanics at Micro- and Nanoscale Levels, Scientific Research (S) No. 19100008, Grant-in-Aid for Young Scientists (A) 19680024.

References

1. Abramoff M., Magelhaes P., Ram S. (2004) Image Processing with Image Journal of Biophotonics International 11: 36–42.
2. Caro C., Pedley, T., Schroter R., Seed W. (1978) The mechanics of the circulation. Oxford, Oxford University Press.
3. Chien S., Usami S., Skalak R. (1984) Blood flow in small tubes In: Handbook of physiology – the cardiovascular system IV, Microcirculation, Part I. Bethesda MD, American Physiological Society, pp 217–249.
4. Cloutier G., Qin Z., Durand L.G., Teh B.G. (1996) Power Doppler ultrasound evaluation of the shear rate and shear stress Dependences of red blood cell aggregation. IEEE Transactions on Biomedical Engineering 43: 441–450.
5. Fahraeus R., Lindqvist T. (1931) The viscosity of the blood in narrow capillary tubes. American Journal of Physiology 96: 562–568.
6. Goldsmith H. (1971) Red cell motions and wall interactions in tube flow. Federation Proceedings 30: 1578–1588.
7. Goldsmith H., Marlow J. (1979) Flow behavior of erythrocytes. II. Particles motions in concentrated suspensions of ghost cells. Journal of Colloid and Interface Science 71: 383–407.
8. Goldsmith H., Turitto V. (1986) Rheological aspects of thrombosis and haemostasis: basic principles and applications. ICTH-Report-Subcommittee on Rheology of the International Committee on Thrombosis and Haemostasis. Thrombosis and Haemostasis 55: 415–435.
9. Ishikawa T., Pedley T. (2007) Diffusion of swimming model micro-organisms in a semi-dilute suspensions. Journal of Fluid Mechanics 588: 437–462.
10. Lima R., Ishikawa T., Imai, Y., Takeda, M., Wada, S., Yamaguchi, T. (2008a) "Measurement of individual red blood cell motions under high hematocrit conditions using a confocal micro-PTV system" (under revision to Annals of Biomedical Engineering).
11. Lima R., Ishikawa T., Imai, Y., Takeda, M., Wada, S., Yamaguchi, T. (2008) Radial dispersion of red blood cells in blood flowing through glass capillaries: role of hematocrit and geometry. Journal of Biomechanics 41: 2188–2196.
12. Lima R. (2007) Analysis of the blood flow behavior through microchannels by a confocal micro-PIV/PTV system. Ph.D. thesis, Tohoku University, Japan.
13. Lima, R., Wada, S., Takeda, M., Tsubota, K., Yamaguchi, T. (2007) In vitro confocal micro-PIV measurements of blood flow in a square microchannel: the effect of the haematocrit on instantaneous velocity profiles. Journal of Biomechanics 40: 2752–2757.
14. Lima, R., Wada, S., Tanaka, S., Takeda, M., Ishikawa, T., Tsubota, K., Imai, Y., Yamaguchi, T. (2008) In vitro blood flow in a rectangular PDMS microchannel: experimental observations using a confocal micro-PIV system. Biomedical Microdevices 10: 153–167.
15. Lima, R., Wada, S., Tsubota, K., Yamaguchi, T. (2006) Confocal micro-PIV measurements of three dimensional profiles of cell suspension flow in a square microchannel. Measurement Science and Technology 17: 797–808.
16. Meijering E., Smal I., Danuser G. (2006) Tracking in molecular bioimaging. IEEE Signal Processing Magazine 23: 46–53.
17. Oka S. (1984) Biorehology. Shokabo, Tokyo.
18. Park J., Choi C., Kihm K. (2004) Optically sliced micro-PIV using confocal laser scanning microscopy (CLSM). Experiments in Fluids 37: 105–119.
19. Schmid-Schonbein, H., Wells, R. (1969) Fluid drop-like transition of erythrocytes under shear. Science 165: 288–291.
20. Shiga T., Maeda N., Suda T., Kon K., Sekiya M., Oka S. (1979) Rheological and kinetic dysfunctions of the cholesterol-loaded, human erythrocytes. Biorheology 16: 363–369.

21. Tanaami T., Otsuki S., Tomosada N., Kosugi Y., Shimizu M., Ishida H. (2002) High-speed 1-frame/ms scanning confocal microscope with a microlens and Nipkow disks. Applied Optics 41: 4704–4708.
22. Wada S., Kobayashi R. (2003) Numerical simulation of various shape changes of a swollen red blood cell by decrease of its volume. Transactions of the Japan Society of Mechanical Engineers 69A: 14–21 (in Japanese).

Chapter 13
Efficiency of Spherical Filters on Detection of Isotropic Defects in Textured Backgrounds

Céline Gouttière and Joël De Coninck

This paper concerns the detection of small defects inserted in various textured backgrounds with more or less spherical filters and wavelets. We have evaluated the detection efficiency of the filters when the controlled isotropic defects are first added in synthetic textured images, then in real reference textured images, the Brodatz textures, and finally in medical images, parts of digital mammographies. Three families of filters are involved: the less spherical family is the Gabor filters, the nearly isotropic wavelets ϕ and ψ, and the Mexican hat filters, which are totally spherical. We have also studied the influence of the defect amplitude by considering various truncations. To achieve this, the defect height was truncated at different percentages.

13.1 Introduction

Many studies have been carried out about texture analysis, like texture classification [1,2], feature extraction [3,4] or defect segmentation [5,6]. The work presented in this paper belongs to the defect detection field, particularly singularity detection in textured images.

It is not obvious to detect singularities in images because they are very small defects in size and in amplitude, sometimes they are not visible to the human eye. This kind of defects can be interesting to detect for various image processing applications, that it is in the medical world or in the industrial sector.

In the past few years, singularity detection has been addressed, notably for the development of a detection algorithm of particle contamination in reticle images [7],

C. Gouttière
University of Mons-Hainaut, C.R.M.M.
e-mail: celine.gouttiere@crmm.umh.ac.be

J. De Coninck
University of Mons-Hainaut, C.R.M.M.
e-mail: joel.de.coninck@crmm.umh.ac.be

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

to detect the characteristic positions in images in automatic inspection system [8], for automatic fingerprint identification system [9] or for image denoising [10], just to quote a few.

We have also contributed to the improvement of the singularity detection in comparing different families of filters and wavelets about conic defect detection in textured images. Our aim was to establish a link between the sphericity of the filters and wavelets and their efficiency to detect singularities, since we can assume that a singularity is a conic defect. We had to measure the sphericity of the filters, to create tunable singularities and to evaluate their detection efficiency.

In our first experiments, the families of filters and wavelets were investigated on the detection of conic defects in synthetic images, which have a tunable textured background [11,12]. Then we have tested other textured backgrounds, real reference textures and parts of mammographies, and the defects were slightly modified [13]. One of the focused applications is the detection of microcalcifications in digital mammographies. This work is in fact an overview of our studies about singularity detection in textured images.

This paper is organized as follows: we begin to describe the tested filters and wavelets in Section 13.2, and we compute their sphericity level. Then three sets of textured images, which are used as background for the experiments, are presented in Section 13.3. Section 13.4 present the two kinds of defects to detect with the tested filters and wavelets. These defects are inserted in the textured images. The methodology of detection is then explained in Section 13.5 and the detection efficiency of the filters and wavelets is evaluated in Section 13.6. Finally, the conclusions of the observations are drawn in Section 13.7.

13.2 Tested Filters and Wavelets

Three families of filters and wavelets were chosen in order to evaluate their capability to detect isotropic defects in textured images. Since we want to establish a link between the sphericity and the detection efficiency of the filters, these were selected on the basis of their sphericity: one filter family with a low sphericity level, which is the Gabor filter, another nearly spherical, the ϕ and ψ wavelets, and a last isotropic one, the Mexican hat filter. Each family is described in the following sections, but let us start with the sphericity measure.

13.2.1 Sphericity Measure

Let us recall the definition of the sphericity. A separable wavelet is said to be spherical if its 2D shape is isotropic, i.e.:

$$\forall x_1, x_2, y_1, y_2 \in R :$$
$$x_1^2 + y_1^2 = x_2^2 + y_2^2 \Longrightarrow G(x_1)G(y_1) = G(x_2)G(y_2) \tag{13.1}$$

The sphericity measure used in this paper was established beforehand for separable wavelets and filters in [14, 15]. It is based on the comparison of the function f with the Gaussian function. This last Gaussian function is chosen because it is the only multidimensional and separable function which is isotropic. We want to measure the closeness of the separable function to the Gaussian with σ specific for the investigated function. This leads us to find the maximum of the peak of the convolution of the wavelet with the Gaussian:

$$S_f = \max_{\sigma \in R_0^+} \left| \frac{\langle f, g_\sigma \rangle}{\| f \| \| g_\sigma \|} \right|, \tag{13.2}$$

where $g_\sigma = e^{-\frac{x^2}{\sigma}}$.

13.2.2 Mexican Hat Filters

The first family of filters is based on the Mexican hat function, which is expressed as:

$$\psi(x,y) = \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{13.3}$$

The Mexican hat filters are interesting for our study because they are isotropic. They were already studied in various applications where the sphericity was an important factor [16, 17].

The sphericity value of this function is the highest. But we know that this filter is non-separable and the sphericity measure is adapted for separable filters. We compute nevertheless S_f for the Mexican hat function and obtain 0.919. This value is very close to 1 and confirms the validity of the evaluation method of the sphericity.

13.2.3 Nearly Isotropic Wavelets

The nearly isotropic wavelets ϕ and ψ are also investigated in this paper. They were created in [14] in improving the sphericity factor of the scaling function for the ϕ wavelet and in optimizing the sphericity of the wavelet function for the ψ wavelet. The length of these wavelets are 8. Figure 13.1 represents the scaling functions of these two wavelet filters. More details about their construction can be found elsewhere [15].

We compute the sphericity measure S_f for the ϕ and ψ wavelets and obtain respectively the values 0.802 and 0.840. They are thus nearly isotropic.

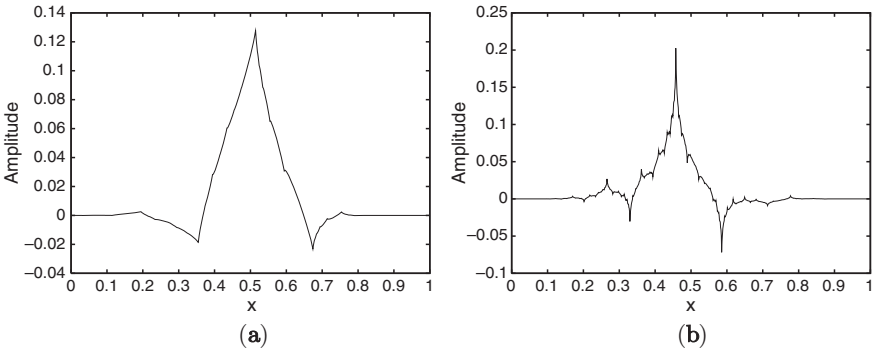


Fig. 13.1 Scaling function of wavelet filter ϕ (**a**) and ψ (**b**)

13.2.4 Gabor Filters

The Gabor filters are chosen because of their tuning properties. Indeed, they are tunable in scale and in orientation. The Gabor function is a sinusoid function modulated by a Gaussian envelope in the space domain. The bank of filters can be obtained by dilation and rotation of a mother function which has the following form :

$$\psi(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \cos\left(2\pi W x\right). \tag{13.4}$$

where W is the modulation frequency of the filter and σ_x, σ_y define the Gaussian envelope size.

The Gabor filters are often used in various applications of image processing like defect detection [18, 19] or texture analysis [20, 21].

These filters have the least spherical form in comparison to the two other families. The value of the sphericity S_f found is equal to 0.354 and is well the smallest.

13.3 Textured Backgrounds

We have tested three sets of textured images in our defect detection experiments. A first set was created by us in order to tune the level of roughness of the images [22], and thus its texture. The second set is composed of real textures from the well-known Brodatz database.[1] As a third set, we have chosen to test medical images, which have a textured background, *i.e.* parts of mammographies. The size of all images is 256×256 pixels. Examples of these three sets of images are shown in the following sections.

[1] http://www.ux.uis.no/~tranden/brodatz.html

13.3.1 Synthetic Textured Images

The advantage of working with synthetic images is the possibility to tune their parameters. In our case, we would like to vary the image background to obtain a textured image. The aim is to create synthetic images which are similar to real images like mammographies. To do this, we use the "Diamond-Square" algorithm which needs only one parameter related to the level of roughness we want. This parameter is called h and can vary between 0 and 1. A close value to 1 corresponds to an image with a high roughness. So, in our experiments, we use synthetic images with h equal to 0.8. The set is composed by 100 synthetic images and four of them are shown in Fig. 13.2.

13.3.2 Reference Brodatz Textures

As this paper is focused on textured images, we have naturally tested real textures of Brodatz. These images are often used for experiments in texture analysis like browsing and retrieval [23], classification [24] or feature extraction [25]. We have chosen the 110 first images available on the web site, four of them are presented in Fig. 13.3.

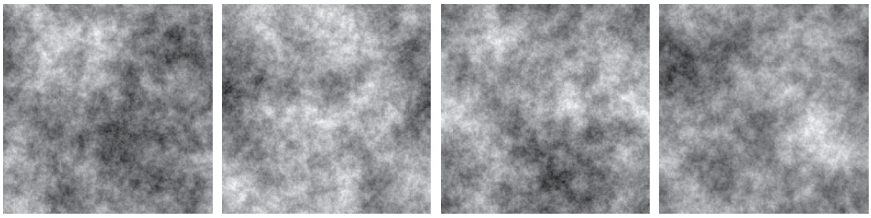


Fig. 13.2 Examples of synthetic textured images

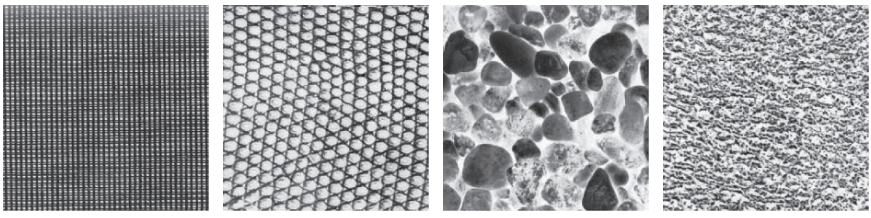
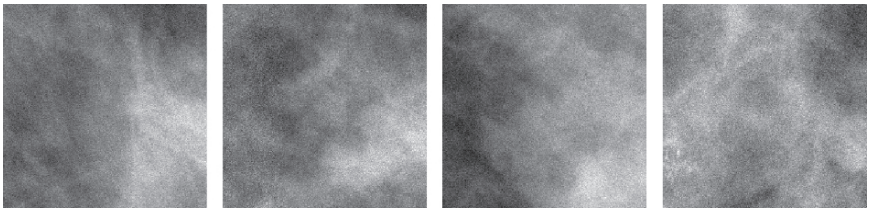


Fig. 13.3 Examples of Brodatz textures



(a) (b) (c) (d)

Fig. 13.4 Examples of digital mammography parts

13.3.3 Medical Images: Mammographies

As a last set of tested images, we have selected 100 mammography parts. So we have applied our defect detection methodology on real cases, which can be similar to detect microcalcifications. We present in Fig. 13.4 four mammography parts from the test set.

13.4 Designing Spherical Defects

Two shapes of spherical defects were investigated for the defect detection: the conic defect and the truncated conic defect. Their construction and the parameters used are explained in the following sections.

13.4.1 Cone-Shaped Defects

The conic defect have white pixels at its center and dark pixels at its rim. The pixels between the rim and the center are variations in grey levels. This kind of defect can be formulate as:

$$D(x,y) = \max\left\{0, 1 - \| \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} c_x \\ c_y \end{pmatrix} \| R^{-1}\right\}, \tag{13.5}$$

where R is the radius of the base of the cone and (c_x, c_y) the position of its center.

In our experiments, we have varied the radius R of the cone from 1 to 10 pixels in order to evaluate the influence of the defect size on the filter efficiency. An example of a such defect is shown in Fig. 13.5. In this case, R is equal to 50 pixels.

13.4.2 Truncated Cone-Shaped Defects

The truncated conic defects are created similarly to the conic defect, but the top of the cone is truncated at some height. An example of a such defect is presented in

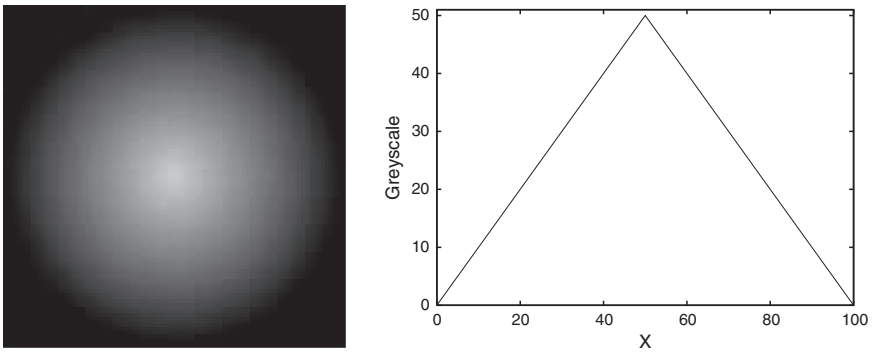


Fig. 13.5 Example of a conic defect with a radius of 50 pixels (**a**) and its profile (**b**)

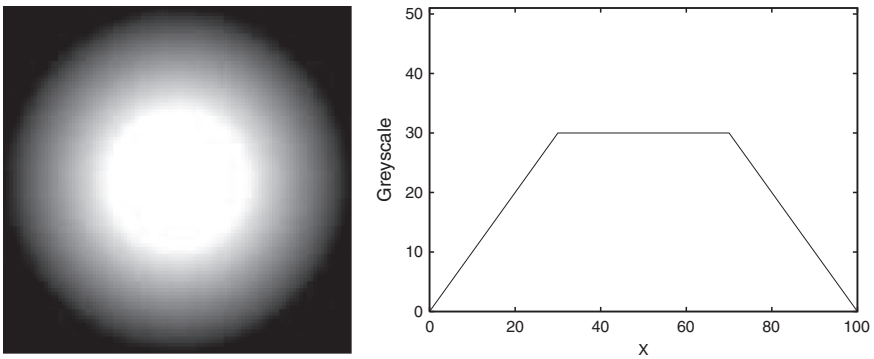


Fig. 13.6 Example of a conic defect with a radius of 50 pixels truncated at 60% (**a**) and its profile (**b**)

Fig. 13.6. This defect has a radius of 50 pixels and is truncated at 60% of its height. In our tests, the defects are truncated from 10% to 100% (the cone is thus complete) and the radius of the defects is 10 pixels to keep the shape. Indeed, a small defect truncated at 10% of its height do not look like a small defect anymore.

13.5 Defect Detection Methodology

The created defects in Section 13.4 are inserted in each image of the three sets described in Section 13.3. We insert ten defects per image. In fact, we add the image containing only the defects D to the textured image I and we obtain the image $I+D$ which is processed by the filters.

Our method determines the detection efficiency of the filters by decreasing the intensity of the defects until a given fraction of the defects are not detected anymore by the filters. The parameter Ω is thus introduced to vary the intensity of the defects, and the image to be filtered is $I + \Omega D$. Ω takes values between 0 and 1. At the beginning of the detection test, Ω is fixed to 1. Ω is then decreased until nine out of ten defects are correctly detected. In our experiments, we tolerate both a false-negative and a false-positive rate of 0.1%. The measure of the detection efficiency is the limiting value of Ω.

13.6 Filters and Wavelets Efficiency About Defect Detection

We present in this section the results obtained by the families of filters and wavelets for the detection of the conic defects and the truncated defects. For each type of defects, we analyze the results for each set of images. The results are presented in a graph a graph wherein the Mexican hat filter, the ψ wavelet and the Gabor filter are compared to the ϕ wavelet. In fact, for each image we consider the difference $\Delta\Omega$ between the best results, *i.e.* Ω, of the ϕ wavelet and the other filters, and we compute the mean and the standard deviation of the values for all images of the set. In the case of the conic defect, the mean and the standard deviation are computed for each radius (from 1 to 10 pixels), and in the case of the truncated defects, it is done for each fraction of truncation (from 10% to 100%). In these graphs, a negative value means that the reference wavelet ϕ gives better results for detection than the considered filter and inversely when positive.

13.6.1 Cone-Shaped Defects

We first present the results of the conic defect detection in the synthetic images. The corresponding graph is shown in Fig. 13.7. We can see that the Gabor filters perform less well than the nearly isotropic wavelets ϕ and ψ for all the considered reduce. For defects with a radius larger or equal to 2 pixels, the Mexican hat filters give better results than the ϕ and ψ wavelets. It is the contrary for the very small defects (radius of 1 pixel). This is probably due to the ability of the wavelets to eliminate the regular part of the signal with great efficiency.

We can observe the same trends for the second set of images, the Brodatz textures in Fig. 13.8. The efficiency of the Gabor filters is however closer to the efficiency of the ϕ wavelet than for the synthetic images. This can be explained by the fact that some Brodatz textures contain directional patterns. As the Gabor filters are oriented filters they would be more appropriate than non-oriented filters to process directional backgrounds.

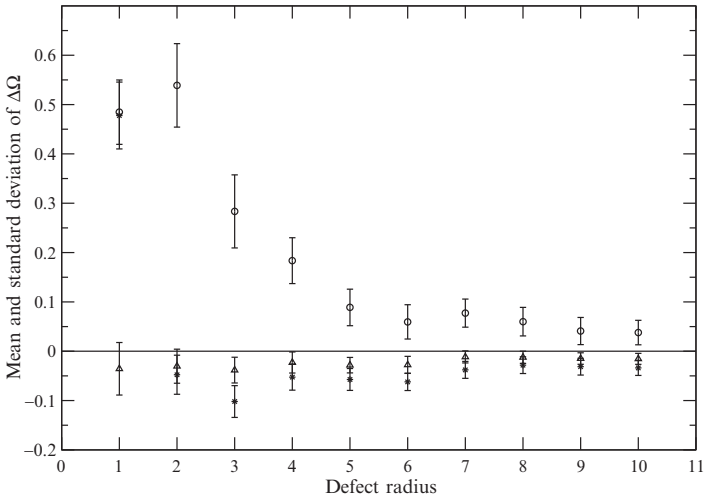


Fig. 13.7 Filters efficiency comparison about conic defect detection with ϕ on synthetic images: ($*$) Mexican hat; ($\triangle$) ψ; ($\circ$) Gabor

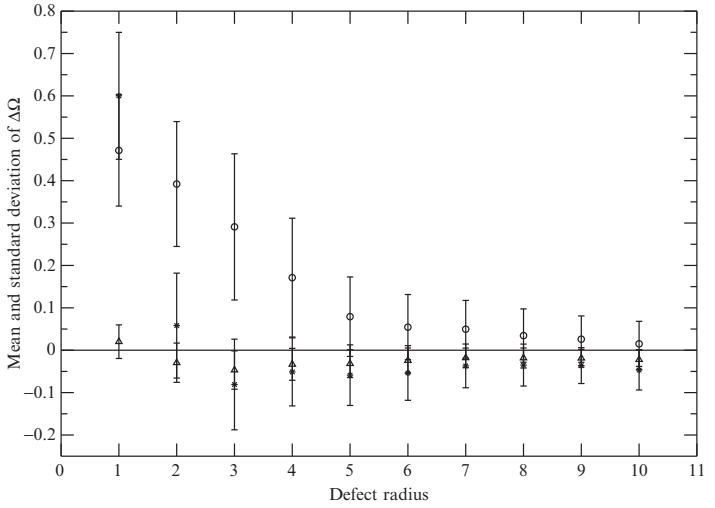


Fig. 13.8 Filters efficiency comparison about conic defect detection with ϕ on Brodatz textures: ($*$) Mexican hat; ($\triangle$) ψ; ($\circ$) Gabor

The observations for the mammography parts are less pronounced than the two other sets of images, the related results are presented in Fig. 13.9. All filters and wavelets seem to have the same efficiency to detect defects with a radius larger or equal than 5 pixels. For the detection of defects with a radius between 2 and 4 pixels, the Gabor filters give worse results than the nearly isotropic wavelets and than the Mexican hat filters, which provides equivalent results. As it was the case for the

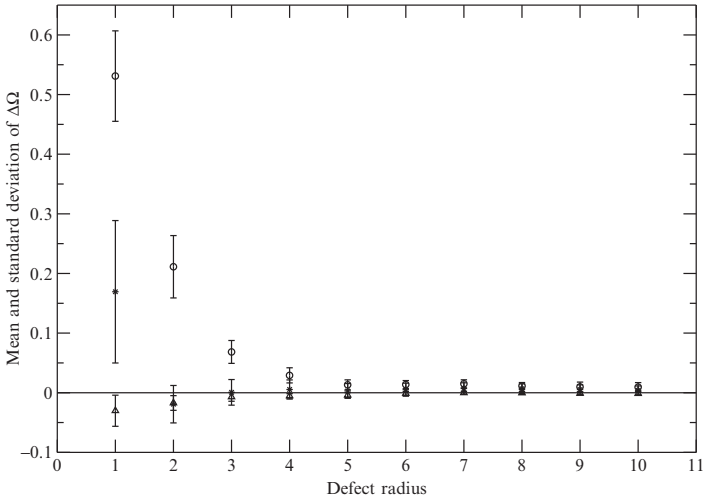


Fig. 13.9 Filters efficiency comparison about conic defect detection with ϕ on parts of mammographies: ($*$) Mexican hat; ($\triangle$) ψ; ($\circ$) Gabor

synthetic images and for the Brodatz textures, the ϕ and ψ wavelets are the most efficient and the Gabor filters are the less efficient to detect conic defects with a radius of 1 pixel.

From all the observations, we can see that the Mexican hat filters give the best results or equivalent results than the ϕ and ψ wavelets, except for very small defects (then it is the contrary). These two families are also the most spherical, as it was explained in Section 13.2. In addition, it has been shown that the Gabor filters detect with less efficiency conic defects than the ϕ and ψ wavelets and than the Mexican hat filters. The sphericity measure of this filter was the weakest, as evaluated in Section 13.2. A link can thus be established between the detection efficiency and the sphericity of the filters and wavelets, for the conic defects. In the following section, the shape of the defects is not modified but the top of the cone is truncated in order to verify if the filters and the wavelets locate the top of the cone for the detection of the defects.

13.6.2 Truncated Cone-Shaped Defects

In this section, the defects have still a conic shape but the top of the cone has been truncated at 10–100% of its maximal height. The radius of the defects studied is fixed to 10 pixels. In order to evaluate the effect of the top of the cone on the detection efficiency of the filters, we have to compare the results for the different fractions of truncation with the results for the truncation at 100%, i.e. when all the cone is considered.

In Fig. 13.10, the results and the order between the filters remain stable from 100% to 50%. Below this fraction, we observe a little detection improvement for the Mexican hat filters. The ϕ and ψ wavelets remain almost unchanged, and the Gabor filters present a slight lowering when the truncation is made at 10%.

Same observations can be seen in the Fig. 13.11 for the Brodatz textures and in the Fig. 13.12 for the mammography parts: for defects truncated at 50% or higher

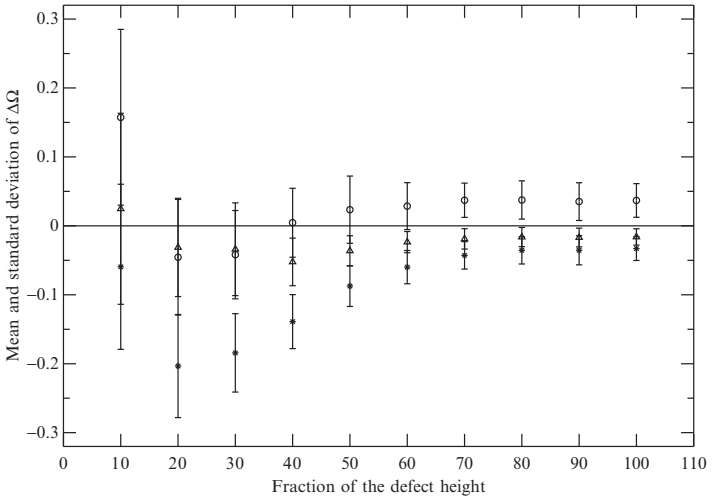


Fig. 13.10 Filters efficiency comparison about truncated defect detection with ϕ on synthetic images: (∗) Mexican hat; (△) ψ; (○) Gabor

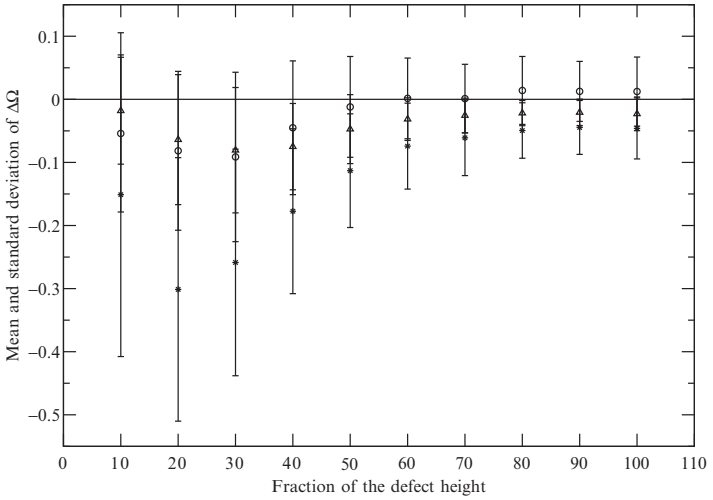


Fig. 13.11 Filters efficiency comparison about truncated defect detection with ϕ on Brodatz textures: (∗) Mexican hat; (△) ψ; (○) Gabor

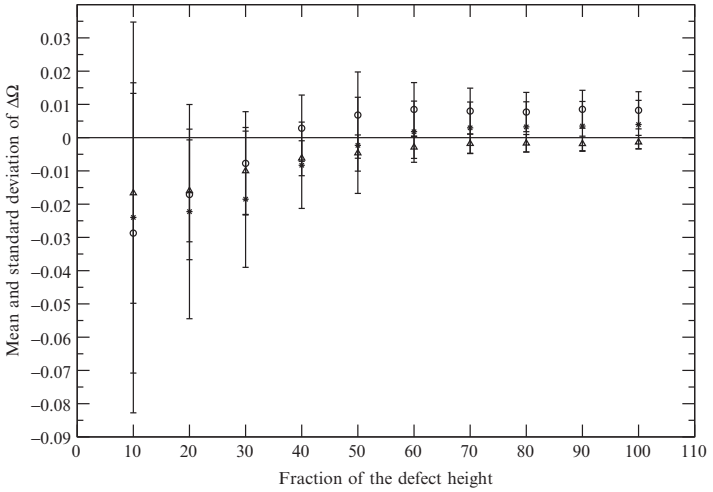


Fig. 13.12 Filters efficiency comparison about truncated defect detection with ϕ on parts of mammographies: ($*$) Mexican hat; ($\triangle$) ψ; ($\circ$) Gabor

than this fraction, the detection results of the filters remain unchanged. Below 50% of truncation, small variations can appear in the results.

In summary, the order of detection efficiency between the filters is slightly disturbed when the top of the cone is truncated. It means that the top of the cone do not play an important role in the detection procedure, it contributes to slightly improve the location of the defects.

13.7 Conclusions

This paper concerns the detection of isotropic defects in textured backgrounds with more or less spherical filters and wavelets. The Gabor filters, the least spherical filters, perform less well than the nearly isotropic wavelets ϕ and ψ for the detection of conic defects, and the Mexican hat filters, which are isotropic filters, provide the best results, almost equivalent to the results obtained with the ϕ and ψ wavelets.

In the second part of the work, the top of the cone has been truncated in order to determine its effects on the detection by the filters. When the truncation is performed at a height larger or equal than 50%, the results of detection remain unchanged. Below this fraction, the variations are very small but the Mexican hat filters are still the most efficient to detect this type of defects.

All observations remain correct whatever the textured background: synthetic textured images, Brodatz textures and parts of mammographies were tested in this work. We can conclude that the sphericity of the filters and the wavelets can be an important key when isotropic or nearly isotropic defects must be detected in various textured images.

Acknowledgements The authors would like to acknowledge the support provided by the *Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture* in Belgium. They also acknowledge the partial financial support of the *Fonds National de la Recherche Scientifique*.

References

1. Soo Chang Kim and Tae Jin Kang. Texture classification and segmentation using wavelet packet frame and gaussian mixture model. *Pattern Recognition*, 40(4):1207–1221, April 2007.
2. K. Muneeswaran, L. Ganesan, S. Arumugam, and K. R. Soundar. Texture classification with combined rotation and scale invariant wavelet features. *Pattern Recognition*, 38(10):1495–1506, October 2005.
3. O. Pichler, A. Teuner, and B.J. Hosticka. A comparison of texture feature extraction using adaptive gabor filtering, pyramidal and tree structured wavelet transforms. *Pattern Recognition*, 29(5):733–742, 1996.
4. Jacques Brochard, Majdi Khoudeir, and Bertrand Augereau. Invariant feature extraction for 3d texture analysis using the autocorrelation function. *Pattern Recognition Letters*, 22(6–7):759–768, May 2001.
5. D. Chetverikov and A. Hanbury. Finding defects in texture using regularity and local orientation. *Pattern Recognition*, 35:203–218, 2002.
6. D.-M. Tsai and C.-P. Lin. Fast defect detection in textured surfaces using 1d gabor filters. *Advanced Manufacturing Technology*, 20:664–675, 2002.
7. Chaoquan Chen and Guoping Qiu. Detection algorithm of particle contamination in reticle images with continuous wavelet transform. In *Proceedings of the British Machine Vision Conference*, 2001.
8. H. Q. Jiang, L. Ma, H. Y. Jiang, and A. Rinoshika. Application of wavelet-based singularity detection technique in automatic inspection system. *International Journal of Wavelets Multiresolution and Information Processing*, 4(2):285–295, June 2006.
9. H. W. Zhang, Y. L. Yin, and G. Z. Ren. An improved method for singularity detection of fingerprint images. *Advances in Biometric Person Authentification, Proceedings*, 3338:516–524, 2004.
10. J. M. Zhong and R. L. Ning. Image denoising based on wavelets and multifractals for singularity detection. *IEEE Transactions on Image Processing*, 14(10):1435–1447, October 2005.
11. C. Gouttière, G. Lemaur, and J. De Coninck. Influence of filter sphericity on the detection of singularities in synthetic images. *Signal Processing*, 87(3):552–561, March 2007.
12. C. Gouttière, G. Lemaur, and J. De Coninck. Influence of sphericity parameter on the detection of singularities in synthetic images. In Joao Manuel RS Tavares and Jorge R. M. Natal, editors, *Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications*, volume 1, pages 211–214, London, July 2007. Taylor & Francis.
13. C. Gouttière and J. De Coninck. Detection of synthetic singularities in digital mammographies using spherical filters. In Joao Manuel RS Tavares and Jorge R. M. Natal, editors, *Computational Vision and Medical Image Processing*, pages 97–100, London, 2008. Taylor & Francis.
14. G. Lemaur. *On the Choice of the Wavelet Basis Function for Image Processing*. PhD thesis, University of Mons-Hainaut, Belgium, 2003.
15. G. Lemaur and J. De Coninck. Sphericity of wavelets may improve the detection of singularities in images. In *Proceedings of Computing Engineering in Systems Applications*, Lille, France, July 2003.
16. L. Cayon, J. L. Sanz, E. Martinez-Gonzalez, A. J. Banday, F. Argueso, J. E. Gallegos, K. M. Gorski, and G. Hinshaw. Spherical mexican hat wavelet: an application to detect non-gaussianity in the cobe-dmr maps. *Monthly Notices of the Royal Astronomical Society*, 326(4):1243–1248, October 2001.

17. J. Gonzalez-Nuevo, F. Argueso, M. Lopez-Caniego, L. Toffolatti, J. L. Sanz, P. Vielva, and D. Herranz. The mexican hat wavelet family: application to point-source detection in cosmic microwave background maps. *Monthly Notices of the Royal Astronomical Society*, 369(4):1603–1610, July 2006.
18. S. Arivazhagan, L. Ganesan, and S. Bama. Fault segmentation in fabric images using gabor wavelet transform. *Machine Vision and Applications*, V16(6):356–363, February 2006.
19. A. Bodnarova, M. Bennamoun, and S. Latham. Optimal gabor filters for textile flaw detection. *Pattern Recognition*, 35(12):2973–2991, December 2002.
20. S.E. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on gabor filters. *IEEE Transactions on Image Processing*, 11(10):1160–1167, 2002.
21. D. M. Tsai, S. K. Wu, and M. C. Chen. Optimal gabor filter design for texture segmentation using stochastic optimization. *Image and Vision Computing*, 19(5):299–316, April 2001.
22. Alain Fournier, Don Fussell, and Loren Carpenter. Computer rendering of stochastic models. *Communication of the ACM*, 25(6):371–384, 1982.
23. B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996.
24. R. Manthalkar, P. K. Biswas, and B. N. Chatterji. Rotation invariant texture classification using even symmetric gabor filters. *Pattern Recognition Letters*, 24(12):2061–2068, August 2003.
25. D. R. Rohrmus. Invariant and adaptive geometrical texture features for defect detection and classification. *Pattern Recognition*, 38(10):1546–1559, October 2005.

Chapter 14
Spontaneous Intracerebral Hemorrhage Image Analysis Methods: A Survey

Noel Pérez, Jose Valdés, Miguel Guevara, and Augusto Silva

14.1 Introduction

Spontaneous intracerebral hemorrhages (ICH) account for 10–30% of all strokes and are a result of acute bleeding into the brain due to ruptures of small penetrating arteries. Despite major advancements in the management of ischemic strokes and other causes of hemorrhagic strokes, such as ruptured aneurysm, arteriovenous malformations (AVMs), or cavernous angioma, during the past several decades, limited progress has been made in the treatment of ICH, and the prognosis for patients who suffer them remains poor. The societal impact of these hemorrhagic strokes is magnified by the fact that affected patients typically are a decade younger than those afflicted with ischemic strokes. The ICH continues to kill or disable most of their victims. Some studies show that those who suffer ICH have a 30-day mortality rate of 35–44% and a 6-month mortality rate approaching 50%. Approximately 700,000 new strokes occur in the United States annually and approximately 15% are hemorrhagic strokes related to ICH. The poor outcome associated with ICH is related to the extent of brain damage. ICH produces direct destruction and compression of surrounding brain tissue. Direct compression causes poor perfusion and venous drainage to surrounding penumbra at risk, resulting in ischemia to the tissues that most need perfusion [16].

Diagnosis of ICH is based largely on clinical history and corroborative Computer Tomography (CT) scanning of the brain. The head CT scan has a sensitivity and specificity that approach 100% for acute ICH. The hemorrhage volume is the most important predictor of clinical outcome after ICH [20]. The volume of ICH can be estimated rapidly with a head CT. It is an important prognostic indicator

N. Pérez, J. Valdés, and M. Guevara
Centre for Advanced Computer Sciences Technologies. Computer Sciences Faculty, Ciego de Avila University, Cuba

M. Guevara and A. Silva
Electronic, Telecommunications and Informatics Department, Aveiro University, Portugal

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

and criterion for therapeutic intervention and its expansion can be associated with neurological deterioration. Contrast-enhanced CT scan and newer CT angiographic (CTA) acquisitions can now be performed quickly with the latest-generation scanners. These images can exclude most gross vascular and tumor causes of hemorrhage rapidly and can have an impact on the therapeutic plan [16].

This paper surveys ICH image analysis methods and computer-aided diagnosis (CAD) systems developed to assist medical personnel in diagnosing and planning the ICH therapeutic treatment. We briefly describe processing techniques such as: segmentation, shape representation, matching, and motion tracking that are often at the algorithmic core of ICH image analysis methods.

The paper is organized in two major sections. Section 14.2 outlines the basic details of selected methods and in Section 14.3 we conclude with a summary of the main contributions and ideas for future work.

14.2 Available Methods

14.2.1 CAD Systems

Several computer-based diagnosis systems (CAD) devoted to intracranial illness have been reported in the literature. These CAD systems have played an important role in quantitative analysis of medical images aiding medical personnel in selecting the appropriated treatment of ICH and others intracranial disorders [5].

Ratajewicz-Mokolajczak and Sikora [22] proposed an intraventricular hemorrhage identification technique based on the combination of electrical impedance tomography (EIT) and artificial neural network (ANN) techniques. This identification method uses a 16-electrode EIT system (see Fig. 14.1). Due to different conductivity values of human tissues, the measured potential on the scalp electrodes depends on the dimensions and the position of the hemorrhage. To simulate the electric potential on the electrodes a variant of the finite element method (FEM) proposed by Ratajewicz-Mikolajczak and Sikora [21] is used. The set of simulated

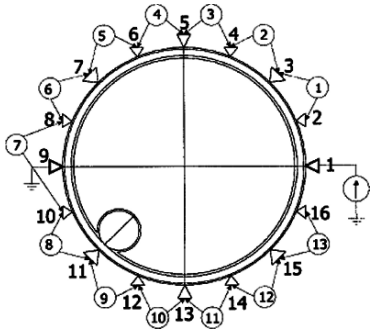


Fig. 14.1 System of electrodes and voltages between them

measurements is generated for different sizes of the identified object. This set is used as a training database for an ANN simulator named NETTEACH [14], then the trained ANN is used to recognize the size and even the position of the object inside the region.

These authors also tried other methods as the "Bell function method" [24], but results were not promising and the ANN approach was them considered more advantageous. It is fast enough, allowing the monitoring of the patients state in real time. Training ANN can be very time consuming and data preparation is not an easy task, but when the neural network is trained, the solution for object's radius identification and the identification of object position is almost instantaneous.

Akter et al. [1] described various Magnetic Resonance (MR) imaging techniques for depicting intracranial microhemorrhage, such as: the gradient-recalled echo imaging (GREI) to detect small hemorrhagic foci in the supratentorial white matter, the GRE-type single-shot echo-planar imaging (GRE-EPI), which is more sensitive than GREI, and the susceptibility-weighted imaging (SWI) method, which is considered by these authors a novel superior technique than the other methods (GREI and GREI – EPI), since it maximizes the lesion visibility and image quality in routine clinical practice. This technique was originally designed for magnetic resonance images MRI venography to depict cerebral venous deoxyhemoglobin, because it is extremely sensitive to susceptibility changes, and is now regarded as rather promising for detecting cerebral hemorrhages reflected by the presence of extravascular blood. The SWI technique consisted of a low-bandwidth (78 Hz/pixel) three-dimensional fast low-angle shot sequence (56/40, 25° flip angle) that was first-order flow compensated in all three orthogonal directions.

To test the methods performance an implementation of these was included in the software of the same MRI console. After applying the methods to the same MRI dataset a quantitative analysis was carried out to measure the signal intensity in hypointense lesions in a circular region of interest (ROI). Three lesions were randomly selected for the measurement of signal. Results showed that the SWI was the best for detecting small hemorrhagic hypointense foci, even in the near-skull-base and infratentorial regions (see Fig. 14.2).

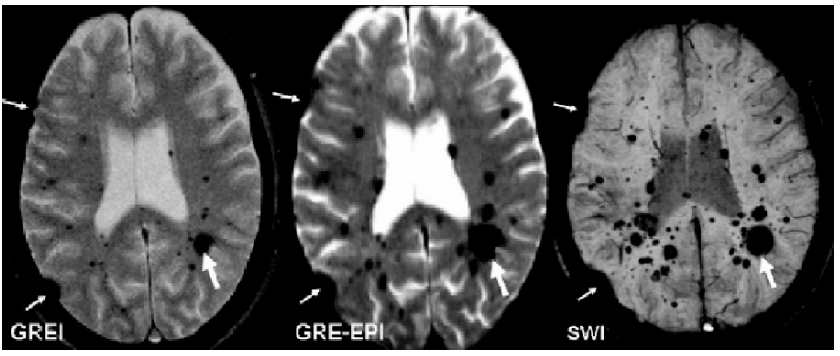


Fig. 14.2 Detection of small hemorrhagic hypointense foci by three techniques

14.2.2 Histogram and Statistical Approaches

Dhawan et al. [2] proposed a method for ICH volume quantification. In this method, three sets of scans were used: within 3 h after first symptoms, 1 h later, and within 20 h after first symptoms. These sets of scans provide the information about the ICH development course. The proposed technique is formed by the following steps: slice interpolation, k-means histogram-based clustering, ICH primary region segmentation and morphological ICH edema region segmentation.

Slice interpolation was carried out using the Catmull-Rom cubic interpolation method [3]. Segmentation of the image in two regions: background and foreground (to obtain the ICH primary region) was accomplished by the k-means histogram-based clustering technique (see Algorithm 1). This was based on the fact that ICH region is brighter than the surrounding background).

Algorithm 1. k-means clustering

1. *let* $v(x), x \in \Re^3$ be the 3D
2. $thr \leftarrow initial\_threshold\_guess$
3. *do*
   (a) $B = \{x \in \Re^3 | v(x) < thr\}$ (background)
   (b) $F = \{x \in \Re^3 | v(x) \geq thr\}$ (foreground)
   (c) $average\_background\_value = \frac{1}{|R|} \sum_{x \in B} v(x)$
   (d) $average\_foreground\_value = \frac{1}{|F|} \sum_{x \in F} v(x)$
   (e) $oldthr \leftarrow thr$
   (f) $thr \leftarrow (average\_barckground\_value + \ldots$
   $average\_foreground\_value)/2$
4. $while\,|thr - oldthr| < \varepsilon$

After this initial segmentation step many disconnected regions may appear, but only one of these binary regions corresponds to the primary region. The idea followed in this approach was to grow with thin layers on the top of ICH primary region and to examine pixel values to determine if they belong to the edema region. The layer growing is based on the 3D morphological binary dilation technique using a ball shaped structuring element. An example of resulting ICH primary and edema volumes is shown in Table 14.1 and the 3D visualization for the resulting ICH primary region is shown in Fig. 14.3.

Table 14.1 ICH primary and edema region volumes

	Baseline	1 h	24 h
ICH primary volume	8.76	9.37	10.05
ICH edema volume	10.03	11.17	15.61

Fig. 14.3 Three-dimensional visualization of ICH primary region

Majcenic and Loncaric [13] described a stochastic method for segmentation of CT head images based on simulated annealing (SA). In the proposed method the segmentation problem is defined a pixel labeling problem with labels for this particular application set to: background, skull, ICH, and brain tissue. The proposed method is based on the Maximum A-Posteriori (MAP) estimation of the unknown pixel labels. A Markov random field (MRF) model has been used for the posterior distribution. The process of segmenting individual pixels is viewed as the process of assigning labels to individual pixels. The label set $L = (l_1, l_2 \ldots, .l_G)$ where G is the number of labels represents the pattern classes in the image. The object set is defined as $P = (p_1,\ p_2 \ldots, .p_M)$ where M is the number of objects. The labeling problem is constrained by a number of rules describing the possible labeling of neighboring objects and/or possible labels for certain objects. The segmentation algorithm tries to estimate the true pixel labeling denoted by $x = (x_1, x_2 \ldots, .x_M)$ and the random field denoted as $Y = (Y_1, Y_2 \ldots, .Y_M)$ where Y_t is the feature vector associated with the t-th pixel. The pixel neighborhood influence is described using a MRF model of statistical dependence among the neighboring pixels. Given a set of observed feature vectors (current image), $Y = y$, and contextual information as an MRF, $P(\mathrm{X} = \mathrm{x})$, the problem lies in finding the optimal estimate of the true labeling x. The MAP method estimates $\widehat{x}$ that maximizes the posterior probability of $X = \widehat{x}$, given $Y = y$. The SA algorithm has been altered in a way to become faster (see Algorithm 2).

Algorithm 2. Simulated annealing

1. Choose an initial temperature T.
2. Initialize $\widehat{x}$ by choosing $\mathrm{x_t}$ as the color $\widehat{x}_t$ that maximizes $P(Y_t = y_t | X_t = x_t)$ for each pixel t.

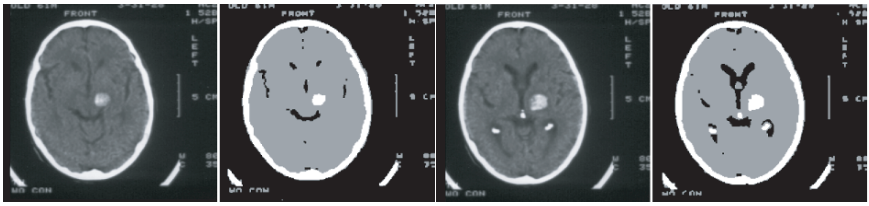
P is the conditional probability density function of the gray levels in the image for a specified label.

3. Perturb x into z by randomly choosing site t and setting its label to the random value in the interval $(0, 1, \ldots, G-1)$. Let

$$\Delta = U(\widehat{X}_t = \widehat{x}_t | Y_t = y_t) - U(\widehat{X}_t = z | Y_t = y_t)$$

   U is the energy function of a single pixel.

   If $\Delta > 0$ then replace $\widehat{X}_t$ by z else replace $\widehat{X}_t$ by z with probability $e^{\Delta/T}$.
4. Repeat step (3) N_{iter} times.
5. Replace T by $f(T)$ where f is the monotonically decreasing function.
6. Repeat steps (3)–(5) until frozen.



a **b** **c** **d**

Fig. 14.4 Two examples of original (**a–c**) and segmented (**b–d**) images

The authors assume distribution of gray values in the image to be normal and image regions of different brightness correspond to different modes of histogram (mean values μ_l and variance values σ_l).

The automatic procedure to determine histogram mean and variance values is the following:

- Smooth the histogram with a Gaussian function (extracts peaks and determine means μ_l)
- Calculate cross-correlation factor between histogram at site μ_l and a Gaussian function with increasing variance σ_l

The experimental results after applying this SA algorithm for two examples are shown in Fig. 14.4.

Imielinska et al. [9] proposed an algorithm for analyzing post-processed perfusion-weighted computed tomography (CTP) images. The CTP estimate a value for cerebral blood flow (CBF) using a series of axial head CT images tracking the time course of a signal from an intravenous contrast bolus [15]. This method converts CBF values, which must be viewed as meaningless outside of the context, to relative "difference," which represents side-to-side asymmetry and is a meaningful value. This conversion is performed by comparing a small region of the scan to the corresponding region in the contra-lateral hemisphere, quantifying the degree of relative difference, and representing this quantity of relative difference in 2D and 3D in a construct termed relative difference map (RDM). The quantitative analysis of the "relative difference" in both brain hemispheres and the six major vascular territories (see Fig. 14.5) contributes to the assessment of the degree of hypoperfusion in the regions. The authors considered two assumptions for the analysis: (1) in normal cases, the axial CT images of the left and right hemispheres are structurally symmetric and comparable, and there should be no more than minor side to side differences in relative blood flow between the two hemispheres; (2) in abnormal cases, the left and right hemispheres are still structurally symmetric and comparable, but there is a significant relative blood flow difference between the two hemispheres. The quantification of symmetry is performed by applying a 9×9 window in both brain hemispheres that visits the opposite regions pixel-by-pixel in a scan-line fashion.

The Kolmogorov-Smirnov test [4] is applied to find the greatest statistical discrepancy between the observed and expected cumulative frequencies between two

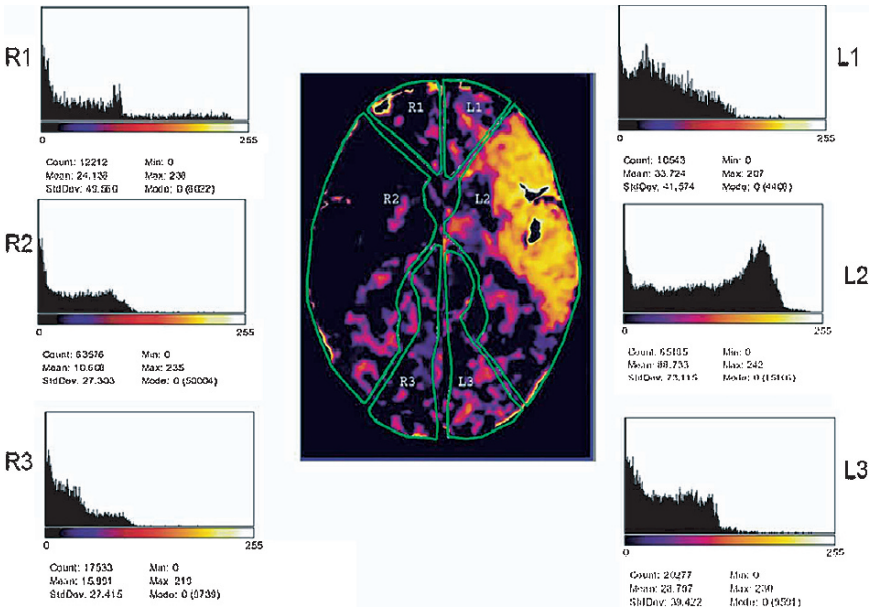


Fig. 14.5 Patient with ischemic stroke: histograms for left and right hemispheres, and six vascular territories

populations. The methodology to develop the proposed Automated System for Quantification of CTP can then be summarized in four key steps:

- Computation of the axis of symmetry of an input CBF image and re-orientation of the image in upright position, if necessary. (Computing the convex hull, centroids and the axis of symmetry by applying Fourier shape descriptors (FSD)).
- Computation of the unwrapped image. (Using FSD of the shape).
- Computation of RDM using the assigned axis of symmetry and 9×9 window difference calculation on the unwrapped image.
- Registration of six vascular territories using generic angles and computation of a histogram for each territory.

The RDMs are calculated based on a method proposed by Imielinska et al. [10]. This method corrects the inherent variability of the CTP methodology seen in the subarachnoid hemorrhage and it is potentially an aid in the diagnosis of cerebral vasospasm (see Fig. 14.6).

Phan et al. [19] describe the creation of a digital atlas (DA) of middle cerebral artery (MCA) infarction associated with MCA branch and trunk occlusion using MRI techniques. Hemispheric infarcts, with evidence of MCA trunk or branch occlusion, were manually segmented into binary images, linearly registered into a common stereotaxic coordinate space, and averaged to yield the probability of involvement by infarction at each voxel. Comparisons were made with existingmaps of the MCA territory. Twenty-eight patients (see Table 14.2) with median age of 74 years (range, 26 to 87 years) were studied. On the DA-MCA, the highest frequency

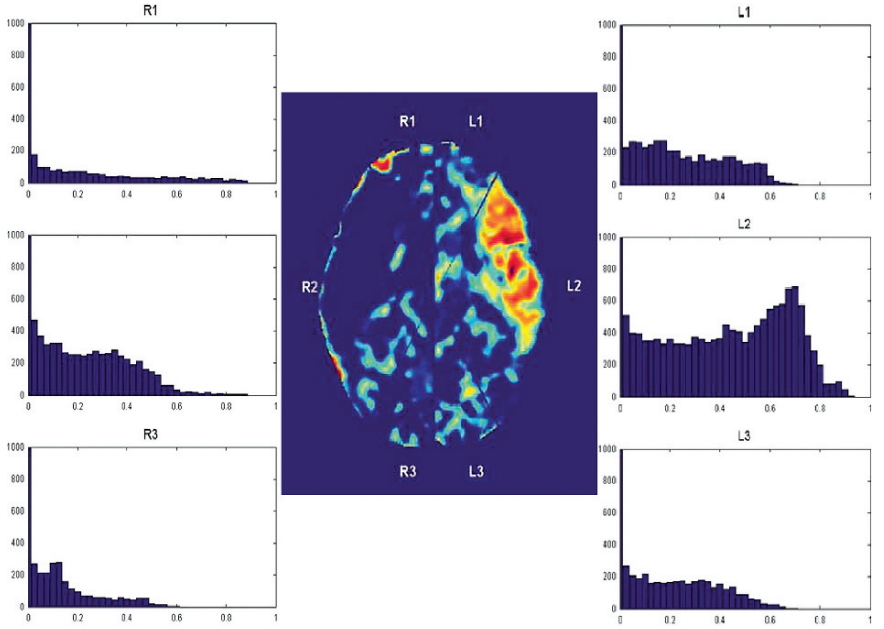


Fig. 14.6 Results with automated RDM/histogram generation: same patient as shown in Fig. 14.5. The automated system provides similar results as those generated by hand drawing

Table 14.2 Patient characteristics, ASPECTS ratings and infarct volume

Patient	Age (years)	ASPECTS				Infarct volume (ml)	Stroke mechanisms
		Rater 1	Rater 2	Rater 3	Rater 4		
1	74	2	3	2	1	147.6	Cardioembolic
2	88	6	7	5	7	96.6	Cardioembolic
3	79	8	8	7	9	7.4	Large artery
4	68	8	9	7	9	15.9	Unknown mechanisms
5	76	4	6	5	5	88.2	Large artery
6	63	4	3	3	3	99.7	Unknown mechanisms
7	77	6	7	9	5	1	Unknown mechanisms
8	70	5	6	4	5	72.8	Large artery
9	56	5	5	4	5	65.9	Large artery
10	77	6	4	6	4	94.4	Unknown mechanisms
11	78	3	5	5	5	87.8	Cardioembolic
12	78	2	3	2	3	139.9	Unknown mechanisms
13	76	1	1	0	2	195.6	Cardioembolic
14	79	5	5	5	5	64.6	Large artery
15	61	5	7	5	5	80.1	Cardioembolic
16	85	3	4	2	3	112.4	Large artery
17	73	6	8	7	7	10.7	Cardioembolic
18	88	0	0	0	0	204.1	Cardioembolic
19	65	9	9	8	9	4.7	Cardioembolic

of infarction was within the striatocapsular region, centrum semiovale, and the insula. The mean and maximal MCA infarct volumes were 195.5 and 366.3 cm^3, respectively. Comparison with published maps showed that the most common difference from the DA-MCA was in the supermedial extent of the MCA territory. Some maps showed the MCA territory reaching the interhemispheric fissure, whereas in the DA-MCA it did not. There was a lower variability in the anterior boundary of the MCA territory compared with its posterior counterpart. This technique demonstrated its usefulness to establish the distribution of the MCA and other arterial territories and the border zones between them with greater certainty.

14.2.3 Knowledge-Based Methods

Cosic [5] proposed an automatic computer-based system to observe and quantitatively analyze 3-D changes in ICH volume and structure. The segmentation of CT images with ICH is performed using an automatic hybrid method, which has an hierarchical structure and it is based on unsupervised fuzzy clustering and expert system-based labeling techniques. The algorithm steps followed were:

1. Digitization of CT films.
2. Alignment of digitized images.
3. Image segmentation (two levels: global and local).
4. Interpolation of segmented images (shape-based interpolation algorithm) [23].
5. Quantitative analysis of interpolated images.

The alignment process of digitized images was performed by means of a cross-correlation between a template (a pattern of the image that is present in all images) from a reference image and the others images.

The Segmentation method has an hierarchical structure and is based on unsupervised fuzzy clustering algorithm, specifically a combination of the fuzzy c-means algorithm and the fuzzy maximum likelihood estimation (FMLE) [8] to break the original image into three number of spatially localized image regions having uniform brightness (global level: bright, dark, and gray regions) and expert system-based labeling to assign one of the following labels to each region: background, skull, brain, calcifications, and ICH local level (see Fig. 14.7) [5].

An expert system was employed for the edema segmentation and the ICH refinement. Every pixel is a single object represented by a fact, which becomes a part of the expert system knowledge. The facts contain the features and the properties of the object. The expert system rules use the pixel distance to ICH and the neighborhood relationships between pixels to decide how to classify a particular pixel. The features used to characterize a pixel are the pixel's coordinates, the intensity, and the ratio between the pixel intensity and the average pixel intensity in a neighborhood of the pixel. The dimensions of the segmented images are 128×128 with 256 gray levels and the results of the algorithm for one slice are presented in Fig. 14.8.

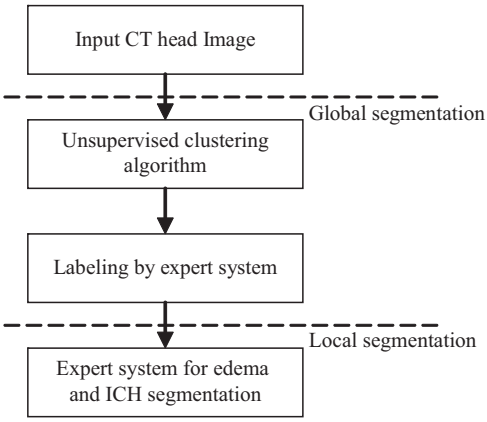


Fig. 14.7 Hierarchical structure of the method for automatic segmentation of CT head images

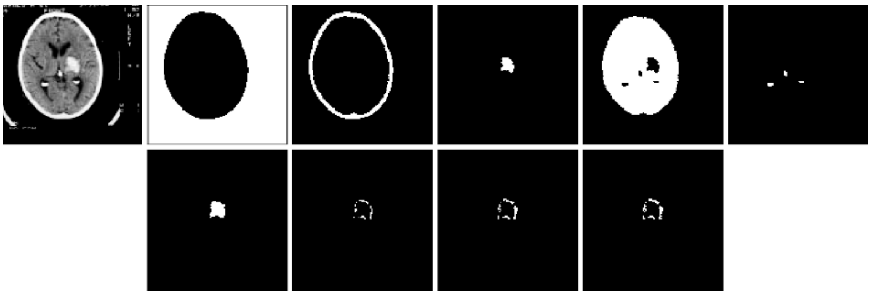


Fig. 14.8 Segmentation results: first row: original image, skull, hemorrhage, brain, background, and calcifications; second row: hemorrhage after refinement, edema segmentation results

14.2.4 Mathematical Morphology Techniques

Perez et al. [17] proposed a 3D mathematical morphology based semiautomatic method. This method relies on the application of 3D mathematical morphology, histogram analysis, optimal thresholding, and prior geometric information techniques. Each patient image study is considered as an anisotropic volume restricted manually to be in the interval formed by the first ICH occurrence and the most outer neighbor ICH slices. The authors recognize that CT head brain images have common intensity histograms, in that way, it is possible to perform some important correlations between image objects of interest and the characteristic intensity levels of skull, grey matter and white matter. The ICH regions are related to higher intensities values. This approach has two main steps: preprocessing and ICH auto detection.

Preprocessing is carried out to remove narrow impulsive spots and lightly smooth the image edges by applying a median filter slice by slice, in axial form, with a windows of 3×3 pixels.

The ICH auto detection consists in computing the global maximum, which is used as the threshold value for the ICH segmentation (to produce a binary mask). Hereafter 3D mathematical morphology operations are applied to auto detect the ICH (see Algorithm 3).

Algorithm 3. 3D mathematical morphology operations

1. Define ICH neighbor slices interval.
2. Create an anisotropic volume of interest (VOI) containing the ICH slices.

$$\mathrm{St} = \{(x,y,z) \in N^3 : 0 < x < x_f, 0 < y < y_f, z_0 < z < z_f\};$$
$$\mathrm{i} = \{\mathrm{i}_1, \mathrm{i}_2, \mathrm{i}_3 \ldots \mathrm{i}_\mathrm{n}\} \qquad \mathrm{n} = 255$$

   x_f and y_f means row and column image size, z_0 and z_f are the most outer neighbor slices. The studies intensities are defined as one-dimensional array i with n intensity levels.
3. Compute $gm = \max(St)$; gm is the maximum intensity value of St.
4. Obtain ICH binary mask using gm as threshold value.

$$ICH_{BM(x,y,z)} = \begin{cases} 0 & \forall\ \mathrm{St}_{(x,y,z)} < \mathrm{gm}^* 0.90 \\ 1 & \forall\ \mathrm{St}_{(x,y,z)} \geq \mathrm{gm}^* 0.90 \end{cases}$$

5. $ICH_{erode} = ICH_{BM(x,y,z)} \ominus se$: where $\ominus$ represent the morphological erosion operation and se is an spherical structuring element (se diameter is set to 5 voxels and it was heuristically determined).
6. Select the ICH_{fa} (biggest foreground connected object in ICH_{erode}), which is a first approach of the ICH segmentation.
7. $ICH_{mask} = ICH_{fa} \oplus se$: 3D ICH binary mask resultant.
8. Apply a flood fill operation to avoid holes inside the ICH_{mask}.
9. Multiply the new ICH_{mask} by the original VOI and ICH segmentation is carried out.

This method was used experimentally by qualified medical personnel to evaluate the behavior and changes of spontaneous ICH (shape, size, etc.) during the disease course. The performance achieved in a representative ICH data set was 83.3% (see Fig. 14.9).

14.2.5 Similarity-Based Algorithms

Perez et al. [18] proposed a semiautomatic technique for ICH CT head images segmentation and tracking. The segmentation and tracking processes involve two main steps: selection of a query object and object retrieval throughout the selected patient study.

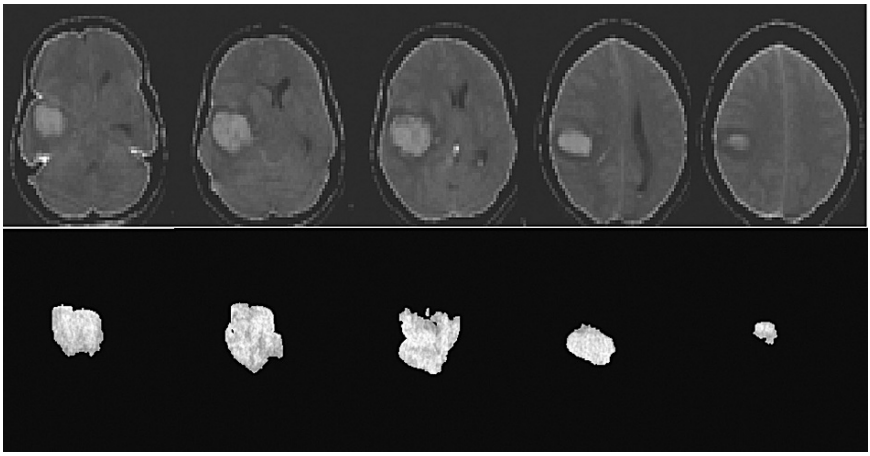


Fig. 14.9 Semiautomatic 3D mathematical morphology results. First row: original images. Second row: segmented ICH

The proposed method includes the following steps:

(a) Image preprocessing
(b) Query object selection
(c) Object retrieval

In order to reduce the computational cost and to improve ICH contrast with respect to background a simple contrast enhancement procedure was introduced in order to emphasize the object's (ICH) edge intensities. A linear contrast stretching function was used to map the gray scale values to new values such that 1% of data is saturated at both intensity extremes.

Query object elements are formed by the reference points and their associated similarity vectors. The quantity of reference points and their associated similarity vectors is variable and depend on the specific problem domain knowledge (e.g. for the ICH problem solution successful results were obtained using only eight reference points).

The selection of query object (I_n) involved several different tasks:

- Extract the object (ICH) contour (CT_n), using livewire techniques [6, 11]
- Extract the reference points (set of points that belong to the query object contour)
- Create similarity vectors from reference points

The reference points are formed by a set of points that are located close or in the object's contour (resultant curve after life wire application) with direction $i\theta$, where $i : 0..\alpha - 1,\ \alpha = \text{int}(360/\theta),\ 0 < \theta \leq 180^\circ$ where θ is the user selected displacement angle, taking the object center of mass as the center of the coordinates system (see Fig. 14.10). Initial reference points are determined by the following algorithm:

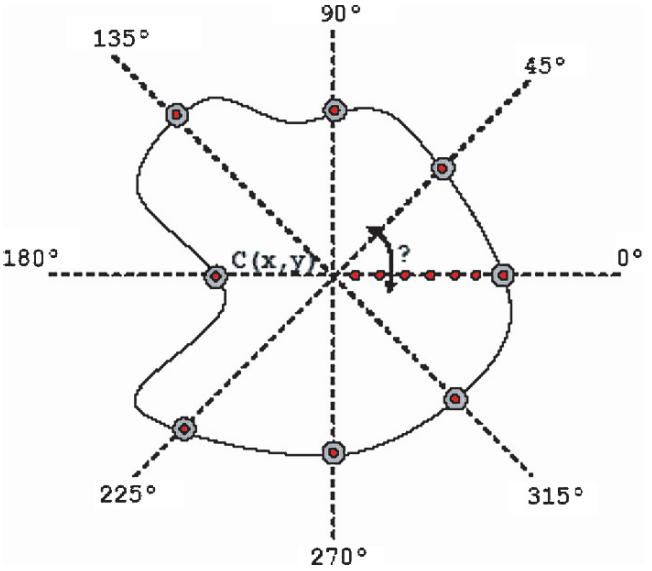


Fig. 14.10 Reference points obtained with $\theta = 45°, \varepsilon = 0.5$

Algorithm 4. Generate initial reference points

1. Select initial slice $I_{(n)}$
2. Select $\theta,\ 0 < \theta \leq 180°$
3. Select: admissible (small) distance between $P_i(x,y)$ and CTn
4. Compute $P_i(X,Y)$: reference points

$$P_i(X,Y) = C_x + r^* \cos(i), C_y + r^* \sin(i)$$

   Coordinates (X,Y) of P_i are obtained by the expression:
   where $r \in Z^+, 1 \leq r \leq R$ *and r increase in* 1 on each iteration until $r = R$ and R is the value in which $d_{Pi}(P_i(X,Y), CT_n) \leq \varepsilon$, d is the euclidian distance between $P_i(X,Y)$ and CT_n
   C_x and Cy are the *(X,Y)* centroid coordinates
5. Store new calculated reference points in the $P_i(X,Y)$ array

Similarity vectors (S_i) are the basic elements that allow to track the ICH regions on the rest of slices (images) belonging to a selected patient study. The reference points represent the centroid $(C_{x,y})$ of the similarity matrices(Q_i). Similarity matrices are *MxM* windows, where $M = 2^*\delta + 1,\ \delta$ is a positive integer scalar provided by the user. The similarity vectors are obtained from the similarity matrices using the following mathematic formulation:

$$Q_i = \begin{matrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3m} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mm} \end{matrix}$$

$S_i = \{a_{1,1} \ldots a_{m,1},\ a_{1,2} \ldots a_{m,2},\ a_{1,m} \ldots a_{m,m},\ a_{1,m+1} \ldots a_{m,m+1}\}$ where S_i represent the set of similarity vectors corresponding to P_i and Q_i.

The object retrieval process in this case aims to track the ICH regions on a set of selected slices. To carry out this process the authors evaluated several similarity descriptors, but the best results were achieved with the similarity descriptor proposed by Fuertes [7,12] (see Eq. (14.1)), which was used to select the new reference points belonging to the slice under analysis:

$$d(P_{i(n)}, P_{i(n+1)}) = \sqrt{(S_{i(n)} - S_{i(n+1)})^t * (S_{i(n)} - S_{i(n+1)})} \tag{14.1}$$

$S_{i(n)}$ is a similarity vector associated to reference point $P_{i(n)}$ in the I_n image, $S_{i(n+1)}$ is a vector associated to the point $P_{i(n+1)}$ in the I_{n+1} slice, t denote a transpose matrix. The new reference points will be those $P_{i(n+1)}$ where $d(P_{i(n)},\ P_{i(n+1)})$ is minimum.

The object retrieval algorithm developed is outline below:

Algorithm 5. Retrieval

1. Select new slice $(I_{(n+1)})$
2. Compute P_{ik} points
   for each i angle
   where $k : 1..D + \sigma$, σ is a value selected by the user, D is the Euclidean distance between C_n *and* $P_{i(n)}$
   Build vector S_{ik} from P_{ik}
   Compute R_{ik} $R_{ik} = s(S_{ik}, S_{in}) :$ s similarity function
3. Select $m = k$ where R_{ik} is minimum

$$P_{ik}(X,Y) = C_{nx} + k^* \cos(i),\ C_{ny} + k^* \sin(i)$$

4. Created and store $P_{i(n+1)}$ and $S_{i(n+1)}$

$$P_{i(n+1)} = P_{im}(X,Y),\ S_{i(n+1)} = S_{im}$$

5. Build the contour $CT_{(n+1)}$ based on the $P_{i(n+1)}$ array
6. Compute the centroid from of $CT_{(n+1)}$
7. If (slice set selected is empty) then end else go to 1

The proposed method was tested on three different set of parameter settings, for θ, the displacement angle, δ, the window size for the similarity matrix and ε, the admissible euclidean distance from CT_n and selected reference points):

Case 1: $\theta = 45°,\ \delta = 20$ *and* $\varepsilon = 0.5$
Case 2: $\theta = 72°,\ \delta = 20$ *and* $\varepsilon = 0.5$
Case 3: $\theta = 90°,\ \delta = 20$ *and* $\varepsilon = 0.5$

The system answered successfully in 30 cases of 36 patient studies that represent the 83.3% of the total analyzed patient studies and the better segmentation results were obtained with the parameter set 1 (see Fig 14.11).

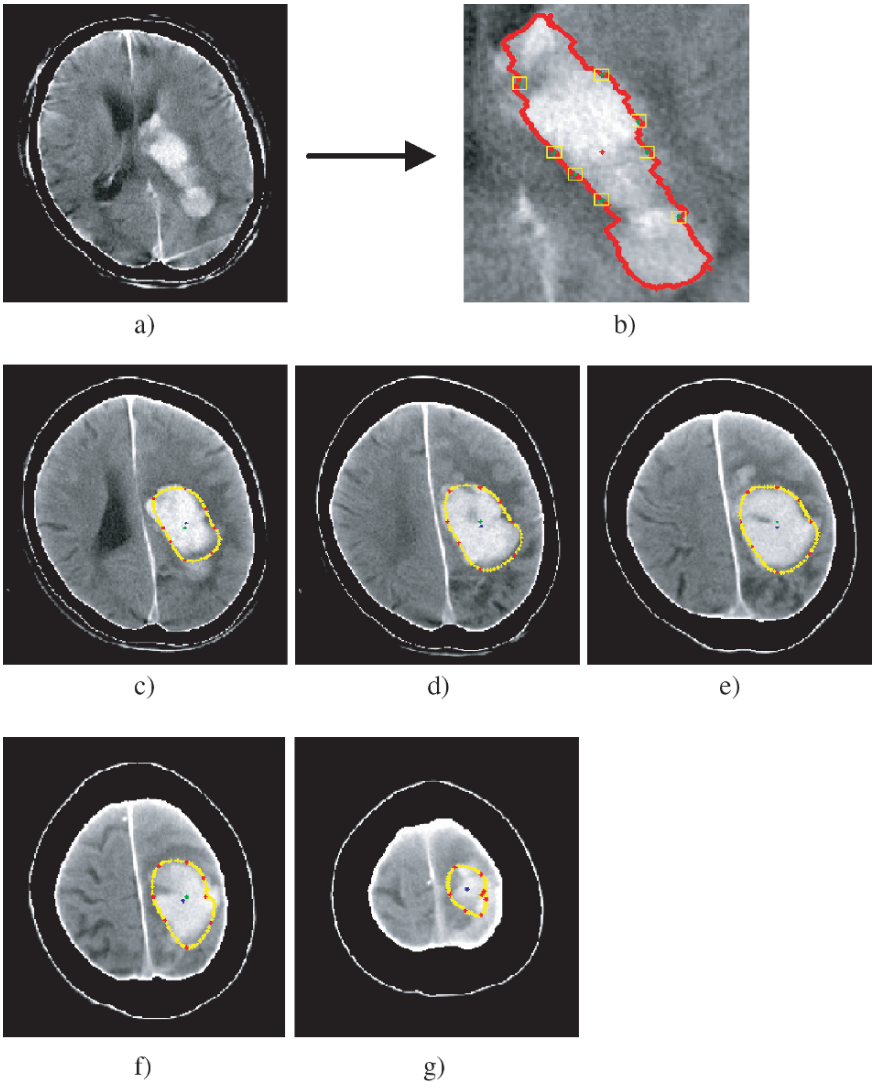


Fig. 14.11 Patient number 138-06 (**a**) selected image, (**b**) query object, (**c–g**) retrieval images

14.3 Summary

Spontaneous intracerebral hemorrhages (ICH) account for 10–30% all strokes and are a result of acute bleeding into the brain by rupturing of small penetrating arteries. The societal impact of hemorrhage strokes are magnified by the fact that affected patients typically are a decade younger than those afflicted with ischemic strokes. The ICH continue to kill or disable most of their victims some studies show that those who suffer ICH have a 30-day mortality rate of 35–44% and a 6-month mortality rate approaching 50%. Diagnosis of ICH is based largely on clinical history and corroborative Computer Tomography (CT) scanning of the brain. The heat CT scan has a sensitivity and specificity that approach 100% for acute ICH. The hemorrhage volume is the most important predictor of clinical outcome after ICH and it can be approximated rapidly with a head CT. Contrast-enhanced CT scan that may now be readily accomplished on the latest-generation scanners. These images can exclude most gross vascular and tumor causes of hemorrhage rapidly and can have an impact on the therapeutic plan.

We survey several available medical image analysis methods, which have been used in CAD systems for segmentation and tracking of ICH.

These methods including diverse algorithms and techniques such as:

- MRI based techniques: susceptibility-weighted imaging (SWI), gradient-recalled echo imaging (GREI) and GRE-type single-shot echo-planar imaging (GRE-EPI)
- Artificial neural networks training based on the electrical impedance tomography signals
- Statistical techniques as frequency histograms and k-means clustering
- Labeling approaches based on the combination of maximum a-posteriori (MAP) estimation and Markov random fields (MRF) models
- Quantitative measure of side to side of cerebral blood flow (CBF) asymmetry algorithm
- Volume region extraction based on digital atlas
- Hybrid approaches including the suitable combination of two or more methods such as:
  - Unsupervised fuzzy clustering and expert system-based labeling
  - Mathematical morphology and histogram based intensity analysis
  - Deformable models and similarity measures

Spontaneous ICH segmentation, at present, is not a solved problem. Future work will be focused on the development of better automatic segmentation and tracking methods to gain in accuracy and precision in the ICH volume determination.

References

1. Akter, M., T. Hirai, et al. (2007). Detection of Hemorrhagic Hypointense Foci in the Brain on Susceptibility-Weighted Imaging: Clinical and Phantom Studies. *Academic Radiologic* **14**: 1011–1019.

2. Atam, P. Dhawan, S. Loncaric, K. Hitt, J. Broderick, T. Brott (1993). *Image Analysis and 3-D Visualization of Intracerebral Brain Hemorrhage*. Proceedings of the Sixth Annual IEEE Symposium on Computer-Based Medical Systems, pp. 140–145.
3. Catmull, E., R. Rom (1974). A class of local interpolating splines. *Computer Aided Geometric Design*_(R.E. Barnhill and R.F. Riesenfeld (eds.)): 317–326.
4. Chakravarti, L., J. Roy (1967). *Handbook of Methods of Applied Statistics*. New York, Wiley.
5. Cosic, D., S. Loncaric (1997). *Computer System for Quantitative Analysis of ICH from CT Head Images*. Proceedings of the 19th Annual International Conference of the IEEE.
6. Falcão, A. X., J. K. Udupa, F. k. Miyazawa (2000). An Ultra-Fast User Steered Segmentation Paradigm: Live-Wire-on-the-Fly. *IEEE Transactions on Medical Imaging* **19**(1): 55–62.
7. Fuertes, J. M. (1999). Recuperación de Imágenes en bases de datos a partir del color y la forma. *E. T. S. de Ingeniería Informática*. Granada, Universidad de Granada. PhD.
8. Gath, I., A. Geva (1989). Unsupervised Optimal Fuzzy Clustering. *IEEE Transactions on PAMI* **11**: 773–781.
9. Imielinska, C., X. Liu, et al. (2005). Toward Objective Quantification of Perfusionweighted Computed Tomography in Subarachnoid Hemorrhage: Quantification of Symmetry and Automated Delineation of Vascular Territories. *Academic Radiologic* **12**: 874–887.
10. Imielinska, C., L. Xin, et al. (2004). *Computer Assisted Radiology and Surgery*. CARS 2004, Amsterdam, Elsevier-Verlag.
11. Liang, J., T. McInerney, D. Terzopoulos (2006). United Snakes. *IEEE Transactions on Medical Image Analysis* **10**: 215–233.
12. Llerena, Y. (2004). Técnicas para el almacenamiento y recuperación de información gráfica en bases de datos de imágenes. *CEIS*. La Habana, Centro Universitario José Antonio Echevarria. Master Sciences.
13. Majcenic, Z., S. Loncaric (1998). *CT Image Labeling Using Simulated Annealing Algorithm*. IX European Signal Processing Conference, Island of Rhodos, Greece.
14. Osowski, S. (1994). Neural Network (in Polish). Warsaw, Warsaw University.
15. Paterakis, K., A. Karantanas, et al. (2000). Outcome of Patients with Diffuse Axonal Injury: The Significance and Prognostic Value of MRI in the Acute Phase. *Journal of Trauma* **49**: 1071–1075.
16. Patrick, C. Hsieh, I. A. Awad, C. C. Getch, B. R. Bendok, S. S. Rosenblatt, H. H. Batjer (2006). Current Updates in Perioperative Management of Intracerebral Hemorrhage. *Neurologic Clinics* **24**: 745–764.
17. Perez, N., J. Valdes, et al. (2007a). *Set of Methods for Spontaneous ICH Segmentation and Tracking from CT Head Images*. 12th Iberoamerican Congress on Pattern Recognition, Viña del Mar/Valparaiso, Chile.
18. Perez, N., J. A. Valdes, et al. (2007b). *A New Semiautomatic Method for ICH Segmentation and Tracking from CT Head Images*. VIP Image, Porto. Portugal, Tylor & Francis.
19. Phan, T. G., G. A. Donnan, et al. (2006). The ASPECTS Template Is Weighted in Favor of the Striatocapsular Region. *ELSEVIER. NeuroImage* **31**: 477–481.
20. Qureshi, A. I., S. Tuhrim, J. P. Broderick, et al. (2001). Spontaneous Intracerebral Hemorrhage. *New England Journal of Medicine* **344**: 1450–1460.
21. Ratajewicz-Mikolajczak, E., J. Sikora (1999). *Neural Network Approach to Electrical Impedance Tomography*. International Symposium on Electromagnetic Fields in Electrical Engineering, ISEF'99, Pravia, Italy.
22. Ratajewicz-Mokolajczak, E., J. Sikora (2002). Neural Networks Method for Identification of the Objects Behind the Screen. *IEEE Transaction on Medical Imaging* **21**(6): 613–619.
23. Raya, S. P., J. K. Udupa (1990). Shape-Based Interpolation of Multidimensional Objects. *IEEE Transactions on Medical Imaging* **9**: 32–40.
24. Sikora, R., Z. Giza, et al. (2000). The Bell Function Approximation of Material Coefficients Distribution in the Electrical Impedance Tomography. *IEEE Transactions on Magnetics* **1036**: 1023–1026.

Chapter 15
Fluid-Structure Interaction Applied to Blood Flow Simulations

Eduardo Soudah, Eugenio Oñate, José García, Jorge S. Pérez, Andrés Mena, Elvio Heindenreich, José Félix Rodríguez, Miguel Angel Martínez, and Manuel Doblaré

Abstract A coupled fluid-structure interaction model has been developed in order to study the vessel deformation and blood flow. This paper presents a methodology from which a smooth surface is obtained directly form segmented data obtained from DICOM images. An integrated solution for segmentation-meshing-analysis is also implemented based on the GiD platform.

15.1 Introduction

Integration of different disciplines is an important aspects in the current development of computer applications in biomedical engineering in order to go from imaging to computer simulations of tissue, organs, or biological systems. Computed Tomography (CT), and Magnetic Resonance Imaging (MRI for short) use special equipment to obtain image data from different angles around the body. Using this information, radiologists can diagnose problems such as cancers, cardiovascular disease, infectious disease, trauma and musculoskeletal disorders more easily [1, 2]. Due to their detailed information, these tools have become an essential tool in preventive medicine.

These images, on the other hand, can also be used to extract the geometry of the organs and tissues for computer analysis via segmentation of the DICOM images (Digital Imaging and Communications in Medicine). After performing the segmentation, a discretization of the domain is required for computer simulation.

E. Soudah, E. Oñate, J. García, and J.S. Pérez
International Center for Numerical Methods in Engineering, UPC, Barcelona, Spain
e-mail: cimne@cimne.upc.edu, esoudah@cimne.upc.edu

A. Mena, E. Heidenreich, J.F. Rodríguez, M.A. Martínez, and M. Doblaré
Group of Structural Mechanics and Materials Modelling. Aragon Institute of Engineering Research. University of Zaragoza. Zaragoza, Spain
e-mail: i3a@unizar.es, jfrodrig@unizar.es

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

Generating a mesh for Finite Element simulations from a segmented image can be cumbersome due to the usually complicated geometry. To overcome this problem, methodologies which make direct use of the segmented data (voxel geometry) have been proposed [3]. Even though they result useful for electrophysiologic simulations, the non-smooth nature of the surface pose serious problems in solid mechanics and fluid-structure interaction simulations. Therefore, methodologies which provide smooth surfaces of the organs and tissues from biomedical images are desirable for computer simulations.

This paper presents a methodology for performing patient-specific computer simulations of cardiovascular systems, in particular fluid-structure interaction in an arterial bifurcation implemented within the Decision Support System DSS-DISHEART. The system incorporates a database for managing patient specific data (i.e., images, cardiovascular data, velocity and pressure profiles, etc.), as well as a number of tools for performing image analysis and segmentation, meshing and finite element tools, with fluid, structure, and fluid-structure interaction capabilities. It also incorporates a Neural-Network for medical decision support. In the example presented in the paper, the image of a femoral bifurcation is initially segmented and voxelized to defined the geometry. The voxel data is then used to produce meshes for the fluid and solid domains. Biomechanical data from each specific patient are then used for the simulations, providing more realistic information regarding the performance of the particular cardiovascular system. The remaining of the paper is organized as follows. Section 15.2 describes the DSS-DISHEART environment. Section 15.3 describes the Methods used for image processing, segmentation, image voxelization and meshing. Section 15.4, Section 15.5 and Section 15.6 detail the fluid and solid solvers and their interaction in FSI simulations. Section 15.7 presents an application to a femoral artery of a patient and Section 15.8 includes some conclusions.

15.2 DSS-Disheart Environment

Five separate programs were required to create patient-specific geometric models from medical imaging data and perform fluid-structure interactions in blood flow simulations. A modular software architecture was then developed (see Fig. 15.1) to allow the use of best-inclass component technology and create a single application capable of making a patient-specific simulation. Figure 15.1 shows the integration environment, a data repository for an abstract data exchange between modules and user, and those modules roughly corresponding to the major tasks in the process. An integrated system was developed utilizing the architecture shown in Fig. 15.1 that enables a professional to go from medical imaging data to analysis results.

In Fig. 15.2, the flowchart of the DSS-DISHEART shows the process to obtain the results of a specific case and how the DSS is created. Figure 15.2 shows five distinctive parts: definition of the cardiovascular problem, definition of the data analysis, results, creation and training of the ANN and Decision Support System. Each of these parts will be explain in the following lines. About the creation and

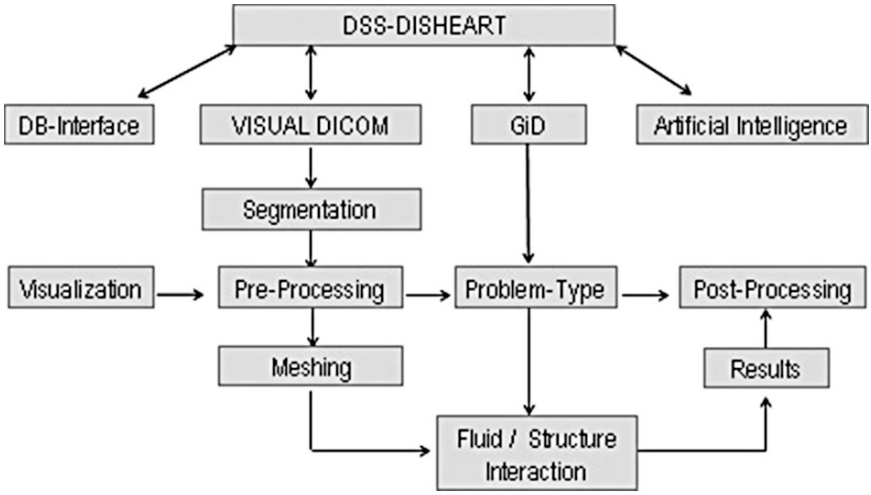


Fig. 15.1 Block diagram depicting the whole process for a fluid-structure interaction problem of a specific patient. The DICOM file of the patient is read to the DSS-DISHEART Data Base. To extract the geometry of the analysis VISUAL DICOM is used, after performing the segmentation of the specific domain for the computer simulation. Subsequently, a new Gid Problem-Type able to visualize, manipulate, generate the mesh and impose the specific boundary conditions for the fluid and the structure problem over 3D medical-geometries is developed. After the simulation, the new problem-type will read the results so the user can analyze them for this specific case

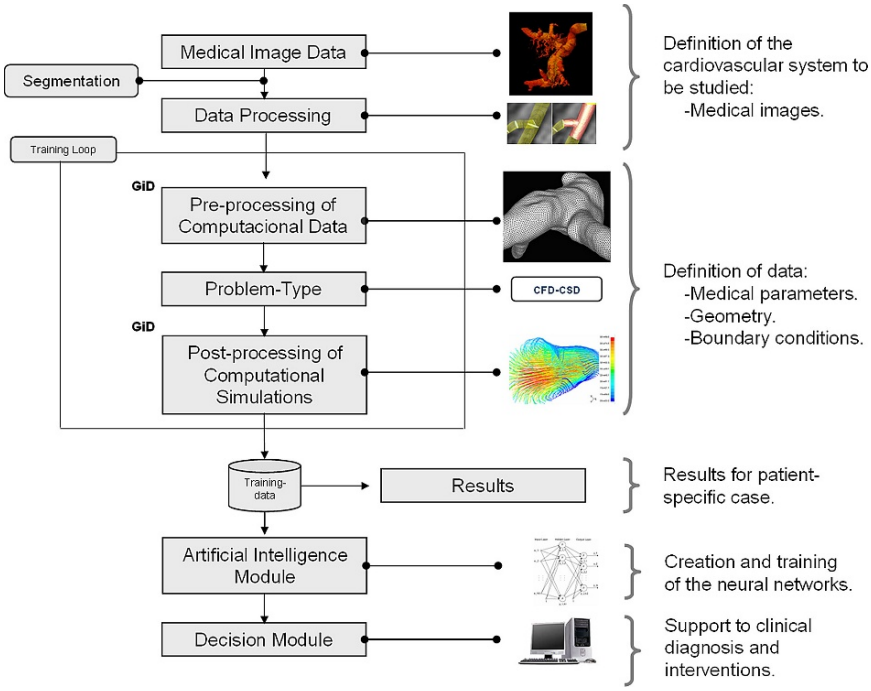


Fig. 15.2 Modular software architecture

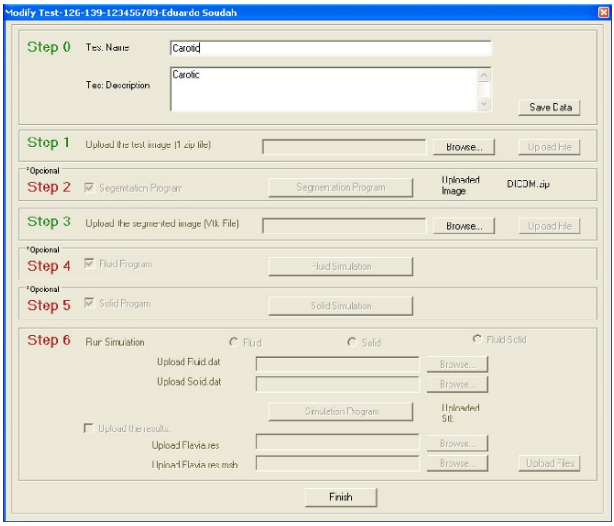


Fig. 15.3 DSS-DISHEART Wizard

training of the ANN and the DSS, more information may be found in [4]. The main function of the Database is to store a great amount of medical and material images, to later analyze them and train the neural network. The Wizard (Fig. 15.3) guides the user through the whole process of introducing and analyzing a specific case of the database. The different stages are thus followed in an ordered and synchronized way. The stages that have been successfully finished are depicted in green, and the still pending steps are depicted in red.

Step 0: Test name and description. The name of the test to be analized is introduced, together with a brief description.

Step 1: Upload the test image. At this stage the user selects the image in DICOM format corresponding to the patient. To assign this DICOM, a browser is automatically opened.

Step 2: Perform the segmentation of the image. At this stage, the image is segmented by using the software VISUAL DICOM. By pressing the button "*Segmentation Program*", a window is automatically opened with the loaded image, and it is segmented using several tools available.

Step 3: Upload the segmentation image. The new file, in format STL or VTK, is stored in the Database.

Step 4: Fluid Simulation. The new 3D geometry file is read and meshed in GiD. In this stage the specific boundary conditions for the fluid are imposed and the Fluid calculation file is created. Additional information may be found in [4].

Step 5: Solid Simulation. The new 3D geometry file is read and meshed using GiD. In this stage the specific boundary conditions for the solid are imposed and the Solid calculation file is created. Additional information may be found in [4].

Step 6: Three different simulations(fluid problem, solid problem and fluid-structure interaction problem) have been implemented. This option gives the user the possibility of studying three different problems, fluid simulation, solid simulation and fluid-structure interaction simulation depending of the specific-case.

In this paper the Interaction simulation based on the FEM is explained in detail. When the calculation has finished, the results are stored in the Database and can be consulted and post-processed.

The system is able to integrate other input files, so the Steps 2, 4 and 5 are optional. This gives more flexibility to the system, since, for example, the user can upload a VTK file without the necessity of using VisualDicom. The segmentation file will be automatically saved in the database. The same happens with the fluid and solid calculation files in Step 4 and Step 5.

15.3 Methods

15.3.1 Medical Images: CT Image Segmentation

The objective of the image segmentation is to find and to identify the boundaries of objects with certain characteristics from the rest of the image. This segmentation allows visualizing and extracting the volume of interest. One of most widely used techniques is the grey thresholding segmentation. It is possibly the simplest and most direct method. The selection of grey thresholds able to identify the object of interest is usually interactive, even though some alternative techniques have been proposed to determine it in a more automatic way. These threshold can be defined in either a local or global data set, and sometimes, over a three-dimensional data set.

In most applications, threshold segmentation is accompanied by manual segmentation which requires the physician expertise. Figure 15.4 shows the resulting segmentation of a human heart.

15.3.2 Meshing Algorithm

The development of finite element simulations in medicine, molecular biology and engineering has increased the need for quality finite element meshes. After segmenting the medical image, a file with the image data and the value of the iso-surface value defining the boundary of the volumen of interest is available. The imaging data V is given in the form of sampled function values on rectilinear grids, $V = F(x_i, y_j, z_k)|0 \leq i \leq n_x, 0 \leq j \leq n_y, 0 \leq k \leq n_z$. We assume a continuous function F is constructed through the trilinear interpolation of sampled values for each cubic cell in the volume. The format used to read the medical data is VTK structured point [7]. The description of this format can be found in [7]. The image in

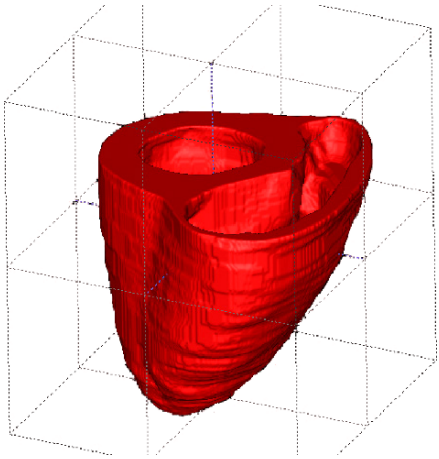


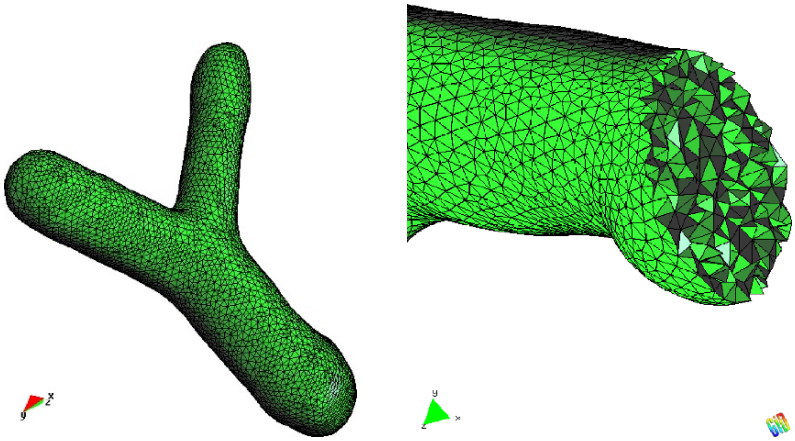
Fig. 15.4 Segmented image of a human heart

this format can also be rendered as a volume and manipulated with the Itk library. Given an isosurface value defining the boundary of the volume of interest we can extract a geometric model of it. We are interested in creating a distcretization of the volume suitable for finite element computation. In this work we have implemented the following methods to generate the finite element mesh to be used in the analysis stage: (i) dual contouring, (ii) marching cubes, (iii) advancing front, (iv) volume preserving Laplacian smooth. All these methods have been integrated into the general Pre/Post-processor GiD [6].

15.3.2.1 Tetrahedral Mesh Generation

In order to generate a tetrahedral mesh from voxels we combine the Marching Cubes method to generate the boundary mesh first and then, after smoothing, an Advancing Front [5] method to fill the interior with tetrahedras. The Marching Cubes [10] algorithm visits each cell in the volume and performs local triangulation based on the sign configuration of the eight vertices. If one or more vertex of a cube have values less than the user-specified isovalue, and one or more have values greater than this value, the voxel must contribute some component of the isosurface. By determining which edges of the cube are intersected by the isosurface, we can create triangular patches which divide the cube between regions within the isosurface. By connecting the patches from all cubes on the isosurface boundary, we get a surface representation.

Some of the triangles generated by the Marching Cubes method do not exhibit good quality to be used in a finite element computation, so in order to improve the quality of those elements we apply a laplacian smoothing with volume preservation. The smoothing algorithm implemented is simple: it preserves the volume after each application of the laplace operator by doing an offset of the vertices along the



(**a**) Mesh generated by marching cubes (**b**) Advancing front

Fig. 15.5 Detail of the mesh process

normals. Figure 15.5a shows the boundary mesh generated by Marching Cubes and this smoothing algorithm.

The Advancing Front [5] is an unstructured grid generation method. Grids are generated by marching from the boundaries (front) towards the interior. Tetrahedral elements are generated based on the initial front. As tetrahedral elements are generated, the "initial front" is updated until the entire domain is covered with tetrahedral elements, and the front is emptied. Figure 15.5b shows a cut of the tetrahedral mesh generated by the Advancing Front method.

15.3.2.2 Hexahedral Mesh Generation

The dual countour method [8] generates a quadrilateral mesh aproximating the boundary of the body. Here we implement a variation [9] of the original method to generate hexahedral mesh for which the boundary is a quadrilateral mesh similar to the one generated by the dual contour method. Dual contouring analyzes those edges that have endpoints lying on different sides of the isosurface, called sign change edge. Each sign change edge is shared by four cells, and one minimizer is calculated for each of them by minimizing a predefined Quadratic Error Function (QEF). The QEF is defined as follows:

$$QEF[x] = \sum (n_i(x - p_i))^2$$

where p_i, n_i represent the position and unit normal vector of the intersection point respectively. For each sign change edge, a quad is constructed by connecting the minimizers. These quads provide an approximation of the isosurface. Figure 15.6 shows an example of the dual contour method in $2D$.

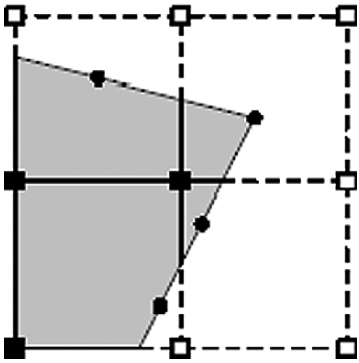
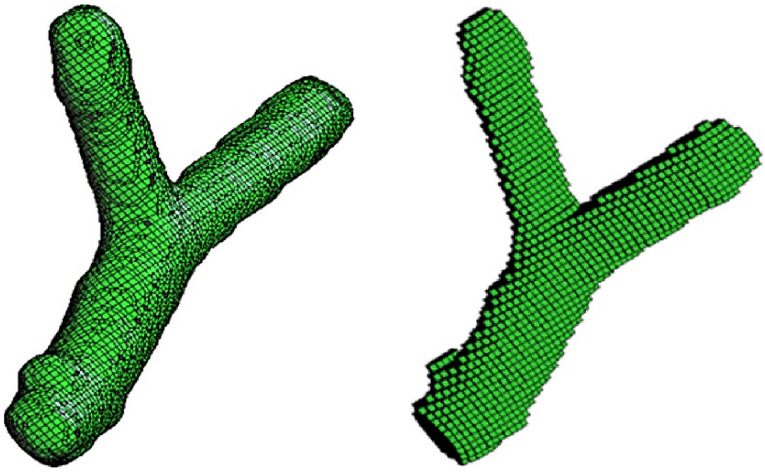


Fig. 15.6 2D example of a dual countour



(**a**) Hexahedral mesh generated by dual contour method

(**b**) Finite difference mesh

Fig. 15.7 Finite element and finite difference meshes

The uniform hexahedral mesh extraction algorithm is simple. Each interior vertex (a grid point inside the volume), which is shared by eight cells, is analyzed. One minimizer is calculated for each of the shared cells, and those eight minimizers construct a hexahedron. An example of a mesh generated by dual contour is shown in Fig. 15.7a.

Also a finite difference mesh can also be obtained. This consist of aligned orthogonal hexahedra. Each pair of neighbor voxels on the boundary of the body generated shares a common face. Figure 15.7b shows and example of such a mesh.

15.4 Computational Fluid Mechanics Solver

Blood is a suspension of red and white cells, platelets, proteins and other elements in plasma and exhibits an anomalous non-Newtonian viscous behavior when exposed to low shear rates or flows in tubes of less than 1mm in diameter. However, in large arteries, vases of medium calibre as well as capillaries, blood may be considered a homogeneous fluid, with "standard" behaviour (Newtonian fluid) [26].

The governing equations for blood flow used in this work are the Navier-Stokes equations, with the assumptions of incompressible and Newtonian flow (90% of the blood is water). For the representation of the Navier-Stokes equations of a deforming fluid domain based on the arbitrary Lagrangian-Eulerian (ALE) method [25], we adopt the following notation: Ω is a three-dimensional region denoting the portion of the space on which we focus our attention, and $x = (x_1, x_2, x_3)$ is an arbitrary point of Ω ; $\mathbf{u} = \mathbf{u}(x,t)$ denotes the blood velocity. For $x\ \varepsilon\ \Omega$ and $t > 0$ the conservation of momentum and continuity in the compact form are described by the following Eq. (15.1):

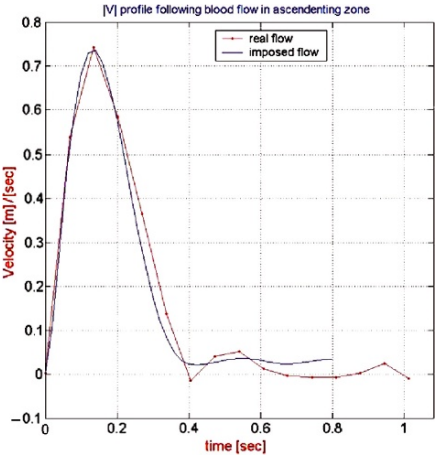
$$\begin{aligned} \rho \cdot \left(\frac{\partial \mathbf{u}}{\partial \mathbf{t}} + (\mathbf{u} \cdot \bigtriangledown \mathbf{u}) \right) + \nabla\, p - \nabla \cdot (\mu \bigtriangleup \mathbf{u}) = \rho \cdot \mathbf{f} \qquad & \text{in } \Omega(0,t) \\ \nabla \mathbf{u} = 0 \qquad & \text{in } \Omega(0,t) \end{aligned} \tag{15.1}$$

where **u** = **u** (x, t) denotes the velocity vector, p = p (x, t) the pressure field, ρ density, μ the dynamic viscosity of the fluid and **f** the volumetric acceleration. Blood flow is simulated for average blood properties: molecular viscosity μ =0.0035 Pa.s and density $\rho = 1050$ kg/m^3. The volumetric forces($\rho \cdot \mathbf{f}$) are not taken in to account in the present analysis.

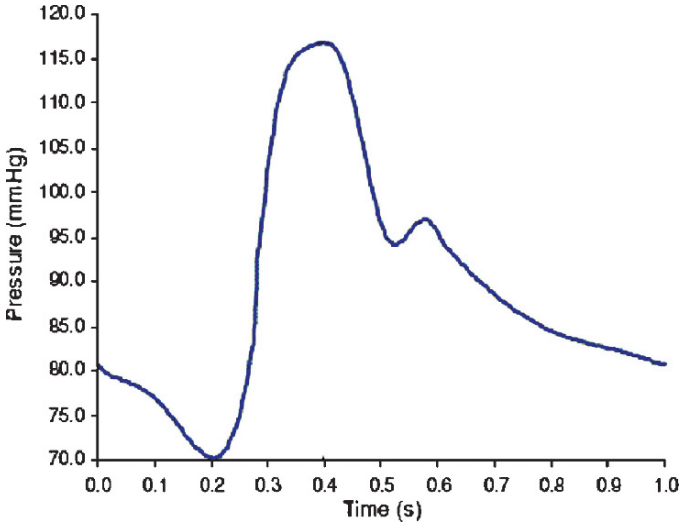
The boundary conditions for the pulsatile flow, the inflow mean velocity is time-dependent and the volumetric flow rate is oscillatory, as shown in Fig. 15.8a. The pulsatile velocity waveform is represented by a polynomial equation based on the in-vivo measurement by magnetic resonance imaging. This pulse is appropriate for normal hemodynamic conditions at the end of an abdominal segment of the human aorta (inflow of study case).

For the outflow the pressure is also time-dependent and oscillatory, as shown in Fig. 15.8b. The pulsatile pressure waveform was calculated using a 1D model [27] and validated con with different analyse.

For solving the fluid problem, Tdyn [21, 22] has been integrated within DSS-DISHEART. Tdyn is a fluid dynamics and multi-physics simulation environment based on the stabilized Finite Element Method. Tdyn's Navier-Stokes solver is based on an iterative monolithic predictor-corrector algorithm, where the correction step is based on imposing the divergence free condition in the velocity field by means of the solution of a scalar equation for the pressure. It uses an innovative stabilization method based on the Finite Increment Calculus (FIC) concept [20, 23, 24]. In this methodology, and that by considering the balance of flux over a finite sized domain, higher order terms naturally appear in the governing equations, which supply the necessary stability conditions for a classical Galerkin finite element discretisation with velocity and pressure interpolations of equal order.



(**a**) Velocity Waveform



(**b**) Pressure wafeform

Fig. 15.8 Velocity and pressure profile

15.5 Solid Finite Element Solver

Biological soft tissues sustain large deformations, rotations and displacements, have a highly non-linear behaviour and anisotropic mechanical properties and show a clear time and strain-rate dependency [11]. Their typical anisotropic behaviour is caused by several collagen fibre families (usually one or two fibres coincide at each point) that are arranged in a matrix of soft material named ground substance [11]. Typical examples of fibred soft biological tissues are blood vessels,

tendons, ligaments, cornea and cartilage. Therefore, to capture the nonlinear anisotropic hyperelastic behavior, it is necessary to consider the formulation of finite strain hyperelasticity in terms of invariants with uncoupled volumetric-deviatoric responses, first suggested in [12, 13], generalized in [14], and employed for anisotropic soft biological tissues in [15, 16].

Let $\mathbf{F} = \partial \mathbf{x}/\partial \mathbf{X}$ be the deformation gradient mapping a point $\mathbf{X}$ in the reference configuration to a point $\mathbf{x}$ in the current configuration. Further, let $J = \det(\mathbf{F})$ be the jacobian of the motion. Proper volumetric and deviatoric uncoupled responses can be defined following the kinematic decomposition

$$\bar{\mathbf{F}} = J^{1/3}\mathbf{F}, \quad \bar{\mathbf{C}} = \bar{\mathbf{F}}^T\bar{\mathbf{F}}, \tag{15.2}$$

where $\bar{\mathbf{F}}$ is deviatoric deformation gradient, and $\bar{\mathbf{C}}$ is the right Cauchy-Green tensor associated to $\bar{\mathbf{F}}$. Let $\mathbf{a}_0$ and $\mathbf{b}_0$ be the directions of collagen fibers within the tissue defining the transverse anisotropic behavior of the tissue [16]. For isothermal processes, we can postulate the existence of a unique decoupled representation of a strain-energy density function Φ [17]. Based on the kinematic decomposition (15.2), and following [12, 16] the free energy for an anisotropic hyperelastic soft tissue can be written in a decoupled form as

$$\Phi(\mathbf{C}, \mathbf{a}_0, \mathbf{b}_0) = U(J) + \bar{\Phi}(\bar{I}_1, \bar{I}_2, \bar{I}_4, \bar{I}_6), \tag{15.3}$$

with

$$\begin{array}{cc} \bar{I}_1 = \mathrm{tr}\bar{\mathbf{C}}, & \bar{I}_2 = 1/2(\mathrm{tr}(\bar{\mathbf{C}})^2 - \mathrm{tr}\bar{\mathbf{C}}^2), \\ \bar{I}_4 = \mathbf{a}_0 \cdot \bar{\mathbf{C}} \cdot \mathbf{a}_0, & \bar{I}_6 = \mathbf{b}_0 \cdot \bar{\mathbf{C}} \cdot \mathbf{b}_0, \end{array} \tag{15.4}$$

$\bar{I}_1$ and $\bar{I}_2$ are invariants of $\bar{\mathbf{C}}$, and $\bar{I}_4$ and $\bar{I}_6$ are the square of the stretches along $\mathbf{a}_0$ and $\mathbf{b}_0$ respectively. For a hyperelastic material, the 2nd Piola-Kirchoff stress is derived from the strain energy as $\mathbf{S} = 2\partial\Phi/\partial\mathbf{C}$. For the form of the strain energy given in Eq. (15.3), the 2nd Piola-Kirchoff and the Cauchy stress, σ, tensors, with the later obtained as the weighted push forward of $\mathbf{S}$, can be written as

$$\mathbf{S} = U' J\mathbf{C}^{-1} + 2J^{-2/3}\mathrm{DEV}\left[\frac{\partial\bar{\Phi}}{\partial\bar{\mathbf{C}}}\right], \quad \sigma = U'\mathbf{1} + \tfrac{2}{J}\mathrm{dev}\left[\bar{\mathbf{F}}\frac{\partial\bar{\Phi}}{\partial\bar{\mathbf{C}}}\bar{\mathbf{F}}^T\right], \tag{15.5}$$

where $\mathrm{DEV}[\cdot] = [\cdot] - \frac{1}{3}([\cdot] : \bar{\mathbf{C}})\bar{\mathbf{C}}^{-1}$, and $\mathrm{dev}[\cdot] = [\cdot] - \frac{1}{3}([\cdot] : \mathbf{1})\mathbf{1}$.

The total energy of the system is then given by the functional

$$\Pi(\mathbf{u}) = \int_\Omega \Phi(\mathbf{X}, \mathbf{C}\mathbf{a}_0, \mathbf{b}_0)dV + \Pi_{\mathrm{ext}}(\mathbf{u}), \tag{15.6}$$

where the explicit dependence on $\mathbf{X}$ accounts for the heterogeneity, and Π_{ext} is the potential energy associated to the external load. The finite element formulation is based in the minimization of Eq. (15.6), which first variation with respect to $\mathbf{u}$ along the direction η is given by

$$D_u\Pi(\mathbf{u}) \cdot \eta = \int_\Omega [\sigma : \nabla\eta - g_{\mathrm{ext}}(\eta)]dV, \tag{15.7}$$

where σ is de Cauchy stress and g_{ext} is the virtual work of the external loading.

Introducing the standard finite element approximation, $\mathbf{u} = \sum_{k=1}^{N_{nod}} = N_k \mathbf{u}_k$, with $\mathbf{u}_k \in \mathbb{R}^3$, and $\mathbf{N}$ the isoparametric interpolation functions, into Eq. (15.7) leads to a nonlinear system of equations. Restricting Eq. (15.7) to a single element

$$G_e(\mathbf{u}, \eta)|_{\omega_e} = \int_{\square} [\sigma_e : \nabla^s \eta] J_e j_\xi d\xi] dV - g_{\text{ext}}(\eta)|_{\omega_e} = 0, \tag{15.8}$$

where j_ξ is the jacobian of the isoparametric mapping, and the integral is carried on the unit cube [15]. The solution of this system of equations for a given increment $\Delta\mathbf{u}$ is performed iteratively by means of Newtons method, by consistent linearization of Eq. (15.8) about the displacement in the current iteration. The linearization of Eq. (15.8) at iteration k is given by

$$L_{\mathbf{u}^k} G_e(\mathbf{u}) = G_e(\mathbf{u}^k) + DG_e(\mathbf{u}^k) \cdot \Delta(\mathbf{u}). \tag{15.9}$$

This linearization leads to a linear system of equations at the element level of the form

$$\mathbf{K}^e(u^k) \Delta\mathbf{u}^{k+1} = \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{int}} \tag{15.10}$$

where $\mathbf{K}^e$ stands for the element stiffness matrix. After imposing proper boundary conditions to the global system of equations, solving for $\Delta\mathbf{u}$, allows computing an approximation to the displacement field at iteration $k+1$ as $\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta\mathbf{u}^k$.

15.6 Numerical Study of the Coupled Fluid-Structure Problem

A distinctive feature of the fluid-structure problem is the coupling of two different sub-problems, the first referring to the fluid (whose solution is characterized by the pressure and turbulence fields of the blood) and the second to the structure (whose unknown variable is the displacement field of the vascular wall). To match the two solvers, we can proceed in many different ways. Similarly, different strategies can be considered for the computation of the grid velocity in the ALE perspective. In this section, we illustrate, in particular, an explicit algorithm for the coupling of fluid and structure.

To consider the problem arising when coupling fluid and structure models, let us restrict our analysis to a domain Ω. The boundary Γ is composed of a portion Γ_C, which is assumed to be compliant, and a part Γ_F, which is assumed to be fixed.

Let us consider the interface conditions between the fluid and the structure. The first condition ensures the continuity of the velocity field, and reads:

$$\mathbf{v} = \dot{\mathbf{u}} \qquad \mathbf{x} \in \Gamma_C \tag{15.11}$$

where $\dot{\mathbf{u}}$ is the velocity field of the vessel and $\mathbf{v}$ the one of the fluid.

The fluid exerts a surface force field over the vessel (we will neglect the possible stresses related to the surrounding organs in our analysis). These forces must be treated as a (Neumann) boundary data for the structure problem:

$$\Phi = -P\mathbf{n} + \mathbf{nS} \qquad \mathbf{x} \in \Gamma_C \tag{15.12}$$

where Φ is the force field vector applied in the vessel due to the blood flow, $\mathbf{n}$ is the normal to the surface, P is the pressure and $\mathbf{S}$ is the extra-stress tensor.

The fluid-structure interaction problem is therefore specified by (15.1), (15.11), (15.12) and the governing equations of the vessel model. In view of its numerical solution, the coupled problem ought to be split at each time step into two sub-problems, one in Ω, the other on Ω_s (the vessel domain), communicating to one another through the matching conditions (15.11) and (15.12). In particular, the structural problem provides the boundary data for the fluid problem while the fluid problem provides the forcing term for the structure.

15.6.1 Numerical Solution

The coupled problem is split at each time into a structure and a fluid problem, communicating each other through boundary terms: a forces field boundary term in the vessel due to the fluid and a velocity restriction of the boundary Γ_C of the fluid. Figure 15.9 shows the basic steps of the algorithm to illustrate the evolution from time level n to time level $n+1$.

The algorithm iteration process for each time step can be resumed as follows:

(a) Solving the structure problem (vessel wall) with the boundary terms due to the blood flow. At the first time level, the scheme is suitable modified, taking into account the initial data on the position and the velocity at time $t = 0$.

(b) Updating domain configuration (Fig. 15.10) and boundary conditions for the fluid solver: Once they are known, we can compute the domain deformations and the movement of the nodes of the grid for the fluid. The new position of the boundary Γ_C is computed through the relation:

$$\boldsymbol{x}_i^{n+1} = \boldsymbol{x}_i^0 + \mathbf{u}_i^{n+1} \tag{15.13}$$

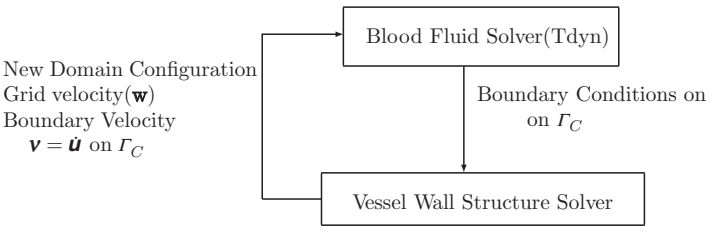


Fig. 15.9 Representation of the splitting in two sub-problem for our approach (coupled solver)

The displacement of the nodes of the grid for the fluid is obtained as a diffusion problem of the boundary displacement into the fluid domain. Diffusion process is based on an arrangement by levels of the mesh nodes, where level 0 corresponds to the mesh nodes on the surface, level 1 to the nodes connected to level 0 nodes, and so on.

The velocity mesh, **w**, is then computed by the equation:

$$\boldsymbol{w}^{n+1} = \frac{1}{\Delta t} \cdot (\boldsymbol{x}^{n+1} - \boldsymbol{x}^{n}) \tag{15.14}$$

The idea underlying this approach is to take advantage of the regularization due to the inversion of the Laplace operator in order to have an acceptable mesh. From time to time, however, it could be necessary to remesh the whole domain, if the grid is too distorted after a certain number of steps.

Another strategy consists of computing the velocity mesh as the solution of the problem:

$$\begin{aligned} -\Delta \boldsymbol{w}^{n+1} &= 0 && \text{in } \Omega(0,t) \\ \boldsymbol{w}^{n+1} &= \dot{\mathbf{u}}^{n+1} && \text{on } \Gamma_C(0,t) \\ \boldsymbol{w}^{n+1} &= 0 && \text{on } \Gamma_F(0,t) \end{aligned} \tag{15.15}$$

Finally, the mesh update is obtained by:

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^{n} + \Delta t \cdot \boldsymbol{w}^{n+1} \tag{15.16}$$

For a comparison of the two strategies, see [19].

(c) Solving the blood flow problem. The ALE formulation of the Navier-Stockes equations (15.1) is solved by using an implicit 2nd order accurate projection scheme. The choice of the time-advancing method satisfies the Geometric Conservation Laws.

(d) Computing the force field applied as a boundary condition on the structural problem due to the fluid. When the boundary nodes of the structure and the fluid are not coincident it is necessary to make an interpolation of the nodal quantities during the interaction algorithm. The methodology used for this interpolation is based on an octtree search algorithm of elements and standard finite element techniques.

This algorithm performs a staggered coupling between the fluid and the structure problems; therefore, it should generally undergo stability limitations on the time step. These limitations could turn out to be restrictive in practical computations.

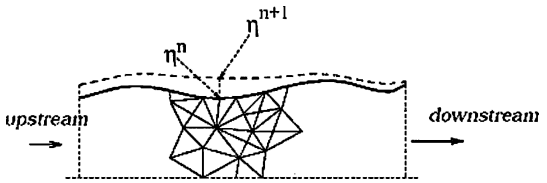


Fig. 15.10 Updating of the mesh

15.7 Results

15.7.1 Femoral Artery

An example of a human femoral bifurcation is considered in this section to demonstrate de methodology. The patient was injected with XXX mL of contrast agent into a peripheral vein. Images of the femoral bifurcation were captured in a 16 Detector/16 Slice Toshiba Multidetector CT Scanner using a slice thickness of 3.2 mm with slices reconstructed every 1.6 mm to maximize the longitudinal resolution. Images were reconstructed using Maximum-Intensity-Projection (MIP) algorithm in the frontal and sagittal views.

Arterial segmentation was automatically performed by means of threshold segmentation using the Visual DICOM software within DSS-DISHEART. Figure 15.7b shows the voxel representation of the geometry of the bifurcation. From this voxel representation, the mesh was generated by the Marching Cubes method and then smoothed as illustrated in Fig. 5.11b.

From the voxel image given in Fig. 5.11a, surface and solid meshes were created for the fluid and solid domains using the software GiD within the DSS-DISHEART. The resulting meshes were composed of 3,003 hexahedral finite elements for the solid wall and 74,125 tetrahedral elements for the fluid. Figure 15.5a shows the meshes for the solid (arterial wall) and the fluid respectively. The solid mesh for the arterial wall was generated by extruding the surface mesh a uniform thickness of 1.5 mm.

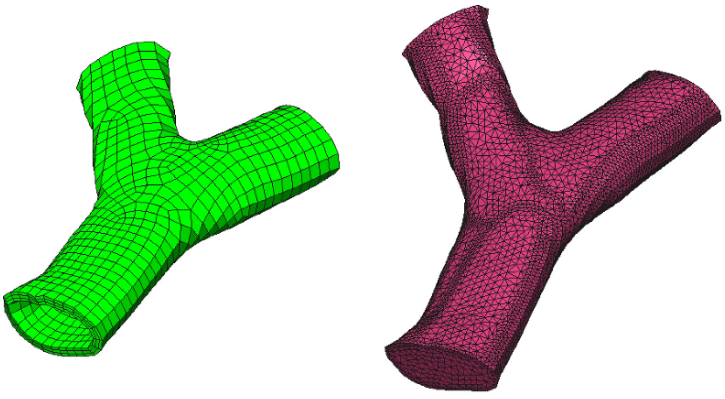


Fig. 15.11 Finite element meshes obtained from the VTK file for the femoral artery. (**a**) Volume mesh for the arterial wall, (**b**) Volume mesh for the fluid

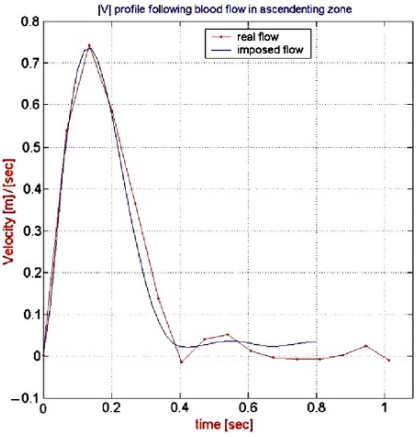
For the solid simulations, the femoral artery was considered as an isotropic material with an strain energy function given by

$$\Phi(\mathbf{C}) = c_{10}(\bar{I}_1 - 3.0) + c_{20}(\bar{I}_1 - 3.0)^2 + U(J) \tag{15.17}$$

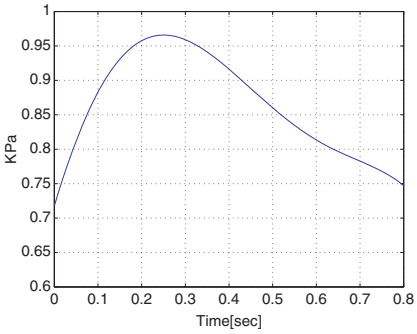
with $c_{10} = 174.0$ kPa, and $c_{20} = 1,880$ kPa [18]. The material was treated as incompressible with the incompressibility constrain treated by means of the three variational principle introduced in [14, 15]. Solid finite element calculations were carried with the software MYDAS and the fluid element calculations with the software TDYN, both integrated within the DSS-DISHEART program. All displacements were restricted at both ends of the femoral artery in the solid domain. For the fluid domain, velocity boundary conditions were imposed at the main branch of the artery, while pressure boundary conditions were applied at both bifurcations ends. Figure 15.12 shows the wave form of both boundary conditions for the fluid used in the calculations.

Figure 15.13 shows the displacement filed of the solid wall and of the ALE-mesh at the instant of maximum pressure. This demonstrates the effective coupling achieved between the solid and the fluid solvers.

Figure 15.14 shows the stress field in the arterial wall and the velocity field in the artery mid-plane. As expected with incompressible fluids, the maximum stress in the arterial wall occurs at the section were fluid velocity reduces since it implies an increment in the local pressure.

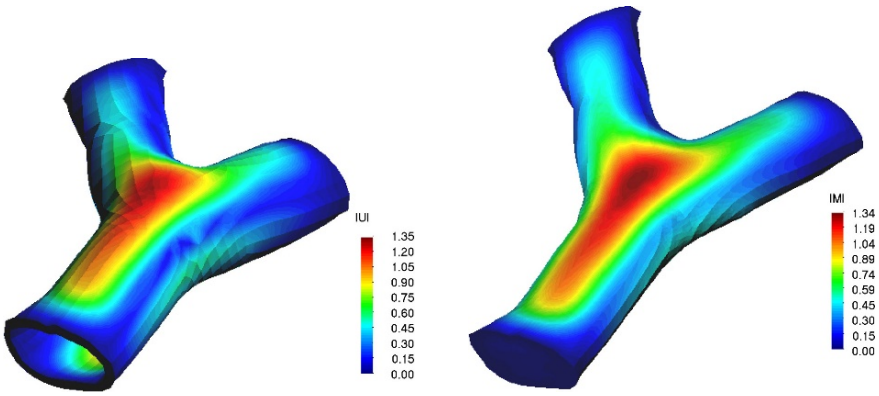


(**a**) Velocity waveform at the inlet (main branch of the bifurcation)



(**b**) Pressure waveform at the outlet of the bifurcation

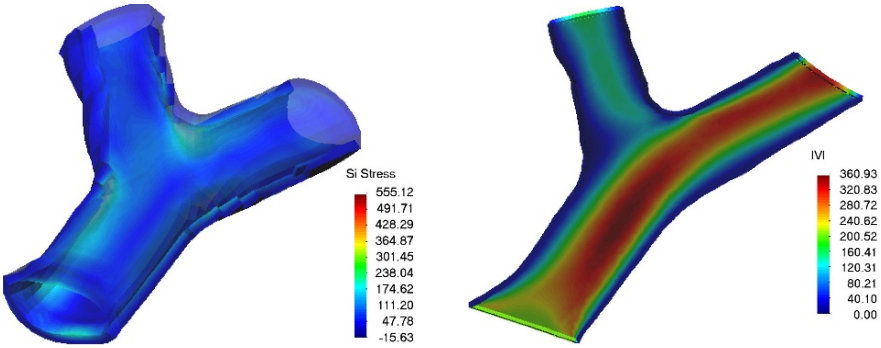
Fig. 15.12 Boundary conditions used in the simulations



(**a**) Magnitude of the displacement of the solid wall in (mm)

(**b**) Displacement of the ALE-mesh in (mm)

Fig. 15.13 Displacement field for the coupled problem



(**a**) Stress field in (kPa)

(**b**) Velocity field in (mm/sec)

Fig. 15.14 Stress and velocity fields in the artery at the time of maximum pressure

15.8 Conclusions

The results show the viability of applying the presented methodology to generate computational finite element meshes from segmentation files obtained from medical images. This tool opens new possibilities for patient specific biomechanical applications.

Acknowledgements The authors would like to thank Dr. Fracesc Carreras and the Cardiology Unit at Sant Pau University Hospital Barcelona for his work on generating of the medical images and their technical support. Financial support for this research was provided by the European Research Area under the Sixth Framework Program CRAFT-Project.

References

1. Fasel, J.H.D., Selle D., Evertsz, C.J.G., Terrier, F., Peitgen, H.O., Gailloud, P. Segmental anatomy of the liver: poor correlation with CT, Radiology. 206(1998):151–156.
2. Goldin, J.G., Ratib, O., Aberle, D.R. Contemporary cardiac imaging: an overview. J Thorac Imag. 15(4)(2000):218–29.
3. Heidenreich, E., Mena A, Rodríguez J.F., Olmos, S., Doblaré M. Simulación de electrofisología cardiaca de imagenes médicas. Modelos numéricos específicos a pacientes. Congreso Annual de la Sociedad Española de Ingeniería Biomédica.,Pamplona, 6–8 November 2006. Pamplona, Spain.
4. Soudah, E., Rodríguez, J.F., Bordone, M., Heidenreich, E., Doblaré, M., Oñate, E. "Grid based decision support system for assisting clinical diagnosis and interventions in cardiovascular problems." M-IS88,ISBN:88-95999-87-1, Vol 1, Vol 2. CIMNE, 2007.
5. Lohner, R., Parikh, P. Three dimensional grid generation by the advancing-front method. Int. J. Numeri. Meth. Fl., 8(1988):1135–1149, 1988.
6. GiD - The personal pre and postprocessor, http://www.gidhome.com/,CIMNE (2006).
7. VTK File Formats,Kitware Inc, http://www.vtk.org/pdf/file-formats.pdf (2006).
8. Ju, T., Losasso, F., Schaefer, S., Warren, J. Dual contouring of hermite data. In Proceedings of SIGGRAPH (2002), pp. 339–346.
9. Yongjie Zhang, Chandrajit Bajaj, Bong-Soo Sohn. "SM'03: Proceedings of the eighth ACM symposium on Solid modeling and applications" (2003), 286–291, ACM Press, New York, NY, USA.
10. Lorensen, W.E., Cline, H.E. Marching cubes: A high reso-lution 3d surface construction algorithm. In Proceedings of SIGGRAPH, pages 163–169 (1987).
11. Humphrey, J.D. Cardiovascular solid mechanics, Springer, New York (2002).
12. Spencer, A.J.M. Theory of Invariants, in Continuum Physics, 239–253, Academic. New York (1954).
13. Flory, P.J. Thermodynamic relations for high elastic materials. Transactions of the Faraday Society, 57:829–838, (1961).
14. Simo, J.C., Taylor, R.L. Quasi-Incompresible Finite Elasticity in Principal Stretches. Continuum Basis and Numerical Algorithms, Comput Methods Appl Mech Engrg, 85:273–310, (1991).
15. Weiss, J.A., Maker, B.N., Govindjee, S. Finite element implementation of incompressible, transversely isotropic hyperelasticity, Comput Methods Appl Mech Engrg, 135:107–128, (1996).
16. Holzapfel, G.A., Gasser, C.T. and Ogden, R.W. A new constitutive framework for arterial wall mechanics and a comparative study of material models. J. Elasticity, 61:1–48, (2000).
17. Ciarlet, PG. Mathematical Elasticity. Vol I: Three dimensional elasticity. Elsevier science publishers, (1991).
18. Ballyk, P.D., Walsh, C., Butany, J., Ojha, M. Compliance mismatch my promote graft-artery intimal hyperplasia by altering suture-line stresses. Journal of Biomechanics, 31: 229–237.
19. Nobile, F.: Numerical approximation of Fluid-Structure interaction problems with application to hemodynamics. PhD thesis-2001. École Polytechnique Fédérale de Lausanne (EPFL) Thesis N 2458.
20. Oñate, E., Valls, A., García, J. "Computational Mechanics", "FIC/FEM formulation with matrix stabilizing terms for incompressible flows at low and high Reynolds numbers", Springer Berlin / Heidelberg, 2006 ISSN: 0178-7675 (Paper) 1432–0924 (Online).
21. COMPASS Ingeniera y Sistemas SA.Tdyn. Environment for Fluid Dynamics (Navier Stokes equations), Turbulence, Heat Transfer, Advection of Species and Free surface simulation. Theoretical background and Tdyn 3D tutorial. March (2002).
22. Oñate, E., García, J., Idelsohn, S.R. and del Pin, F. "Computer Methods in Applied Mechanics and Engineering", "Finite calculus formulation for finite element analysis of incompressible flows. Eulerian, ALE and Lagrangian approaches" Elsevier,Laussane (Switzerland),2006,ISSN 0045–7825.

23. Oñate, E., Valls, A., García, J. “Journal of Computational Physics”, “Modeling Incompressible Flow at Low and High Reynolds Numbers via a Finite Calculus-Finite Element Approach”, Elsevier, New York (USA) 2007, ISSN 0021–9991.
24. Oñate, E., Valls, A., García, J. “International Journal for Numerical Methods in Fluids”, “Computation of turbulent flows using a finite calculus-finite element formulation”,John Wiley and Sons, London (GB) 2007, ISSN 0271-2091.
25. Formaggia, L., Nobile, F., Quarteroni, A., Veneziani, A. (1999). Multiscale modelling of the circulatory system: a preliminary analysis. Comput. Visual. Sci. 2, 75–83
26. Perktold, et al., “Pulsatile non-Newtonian Blood Flow in Three-Dimensional Carotid Bifurcation Models: A Numerical Study of Flow Phenomena Under Different Bifurcation Angles”, Nov. 1991, J. Biomed. Eng., Vol. 13, pp. 507–515.
27. Soudah, E., Mussi, F. and Oñate, E. “Validation Of The One-Dimensional Numerical Model In The Ascending-Descending Aorta With Real Flow Profile”.III International Congress on Computational Bioengineering (ICCB 2007) Venezuela.

Chapter 16
Validity of Paranasal CT Image Reconstruction for Finite Element Models in Otorhinolaryngology

Maria Elizete Kunkel, Analia I. Moral, Kathrin Tingelhoff, Friedrich Bootz, and Friedrich Wahl

Abstract The purpose was to evaluate an approach for use of segmented computed tomography images in volumetric estimation of the paranasal sinuses cavities. Four hundred and fifty-two CT images were processed with the software Amira™ 4.1. The images were obtained from a dummy human head, which is used to rehearse the movements of the surgeon during endoscope nasal surgery. The volumes of the frontal, maxillar, sphenoidal and ethmoidal sinuses were examined both by material injection and by 3D-reconstruction of CT images. The volumes of the paranasal cavities were all in the respective ranges compared with previous reports. The precise knowledge of the geometric configuration of the paranasal regions is necessary because reconstruction on the paranasal sinuses will be used for the creation of finite element models for Endonasal surgery simulations.

16.1 Introduction

16.1.1 Paranasal Sinuses Anatomy

The nasal cavity is a large air-filled space above and behind the nose in the middle of the face. It is divided into right and left halves by the nasal septum. The paranasal sinuses are four pairs of hollow structures within the bones surrounding the nasal cavity (Fig. 16.1). The sinuses are divided into subgroups named according to the bones they lie under. The frontal sinus is located over the eyes, in the forehead bone; the maxillary sinuses, under the eyes, in the upper jawbone; the ethmoidal sinuses are comprised of a variable number of air cells, ranging from 3 to 18 on each side, and they are between the nose and the eyes, backwards into the skull; and finally, the

M.E. Kunkel, A.I. Moral, K. Tingelhoff, F. Bootz, and F. Wahl
Institute for Robotics and Process Control, Technical University of Braunschweig, Germany

J.M.R.S. Tavares, R.M.N. Jorge (eds.), *Advances in Computational Vision and Medical Image Processing,* Computational Methods in Applied Sciences 13,

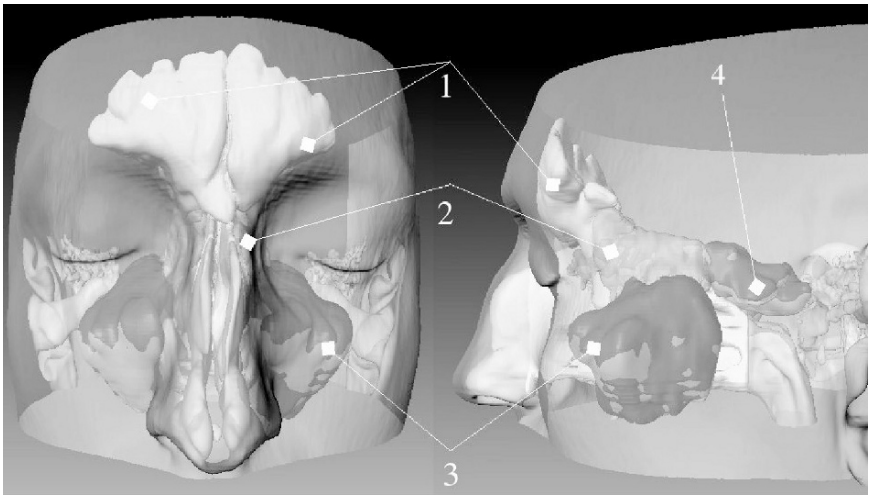


Fig. 16.1 Frontal and lateral view of the human paranasal sinuses: frontal (1), ethmoidal (2) maxillary (3), and sphenoidal (4) sinuses

sphenoidal sinuses, are located in the centre of the skull base [11]. The shape, size and position of each sinus differ between individuals and variation may even occur on either side of the head [17].

All paranasal sinuses have their origin in the ethmoidal portion of the nasal cavity and communicate with this above the inferior nasal concha via small orifices called ostia [15].

16.1.2 Functional Endoscopic Sinus Surgery

Endonasal surgery has become standard for the treatment of pathologies that can be reached via the nasal cavity. Functional endoscopic sinus surgery (FESS) is a minimal invasive approach adopted in case of chronic sinusitis (inflammation of the paranasal sinuses) and is aimed to restore normal physiology by reestablishing normal mucociliary drainage and ventilation of the sinuses [6, 19]. During FESS the surgeon needs to remove the localized mucosal disease obstructing the nasal pathways, for that it is necessary to move the endoscope and other surgical instruments within the nasal cavity and through the ostia to reach the paranasal sinuses [27]. Although most of the otolaryngologists accept FESS as the best treatment for chronic sinusitis [3, 8], the technique has its clear limitations as well as its specific problems. The major drawback in this surgery is regarding the surgeons to be subject to fatigue due to handling of surgical instruments with one hand for long time periods while the other hand is guiding the endoscope.

Surgical workflows demonstrate the need and the feasibility of automatic assistance in guiding the endoscope in FESS [20].

16.1.3 Purpose

A project for robotic endoscope guidance in endonasal surgery is being developed for the Robotic Surgery Group at the Institute for Robotics and Process Control (Technical University of Braunschweig, Germany) in cooperation with the Clinic and Policlinic for Ear, Nose and Throat/Surgery (University of Bonn, Germany) [4, 9, 10, 13, 21, 22, 25]. In that context, models of the nasal cavity and paranasal sinus, based on Finite Element Method (FEM), are being created to perform simulation of the mechanical behaviour of the inner nasal structures under endoscope loading during endonasal surgery. Finite Element Analysis (FEA) will provide important information for the safe control strategy of the robotic endoscope guidance system.

Some characteristics of the paranasal structures as inhomogeneity, anisotropy, viscoelasticity and nonlinearity make difficult the task of producing its biomechanical model. Therefore, to produce an initial model, it is necessary to use simplified concepts, to gain more insight into the mechanical behaviour of this structure during endoscope interaction. Despite of the structures that form the human nasal structures present inhomogeneities due to the diversity of the biological tissues (bone, cartilage, mucosa and others), it is possible consider the hypothesis that a homogeneous model represents the deformation of the nasal structures under endoscopic loading. This is acceptable just if it is assumed that the inner nasal structures function as a linear system submitted to small deformation due to endoscopic mechanical contacts during FESS.

In this paper, we present results concerning to obtaining of the geometric configuration of the human paranasal sinuses for FE modelling. Our purpose was to perform segmentation of the paranasal sinuses regions for three-dimensional (3D-) reconstruction, visualization and volumetry. To evaluate the approach, the volume of the paranasal sinuses was examined both by material injection and by 3D reconstruction of CT images.

16.2 Materials and Methods

16.2.1 Dummy Human Head

The anatomical model of the nose and paranasal sinuses is a dummy human head from silicone (Axel LANG, Zurich, Switzerland), based on a cadaver specimen with normal anatomy that was designed to reproduces the human nasal structures and to provide a realistic training environment for endoscopy for endonasal sinus surgery, allowing the implementation of the robotic endoscope guidance, reproducing de movements of the endoscope as a real sinus surgery (Fig. 16.2) [12]. The dummy human head is for didactic purposes divided into five parts showing different level

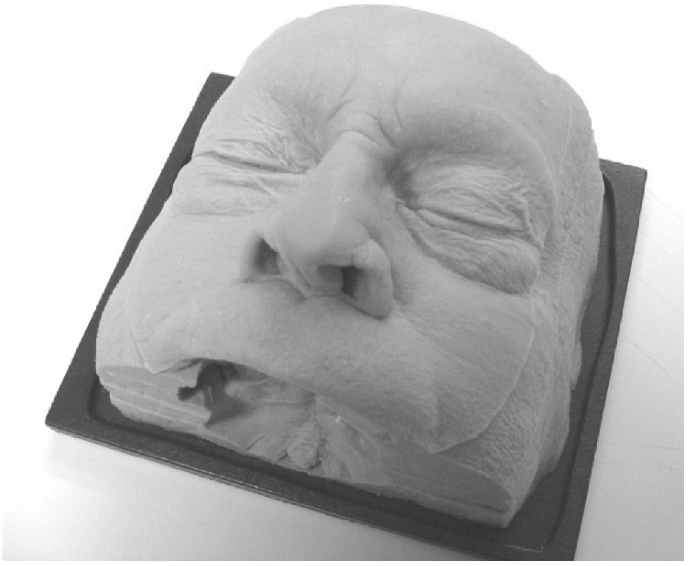


Fig. 16.2 Three-dimensional-model of the nose and paranasal sinuses (Axel LANG, Zurich, Switzerland) to be used for the homogeneous model of the nasal cavity and paranasal sinuses

of the nasal structures. A total of 452 CT images from the dummy human head were acquired by a spiral CT from Philips. The images have high resolution $(0.0390625 \times 0.0390625 \times 0.04)$ and a slice thickness of 0.625 mm.

16.2.2 Segmentation and Reconstructions

A semi-automatic segmentation of the CT images of the nasal cavity and paranasal sinuses was performed using Amira™ 4.1 software for medical images (Mercury Computer System Inc., USA). The same software was also used to perform the 3D-reconstruction and volumetry. Since the material of the anatomical head is homogeneous only one type of material was used. The semi-automatic segmentation was performed, using different tools that segment the image following an image gradient or a growing region based on the grey level of the object (head), the background or the nasal cavities. These regions could be clearly differentiated in all CT images (Fig. 16.3). The regions identified in the segmentation were the model of the dummy human head and the nasal cavity and paranasal sinuses that were separately segmented (Fig. 16.4).

To verify the correspondence between the model and the segmented head, a 3D-reconstruction of surface of the nasal cavity and paranasal sinuses was performed (Fig. 16.5). The 3D volume was generated after the segmentation results were post processed.

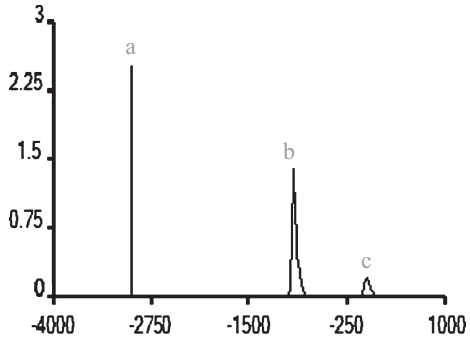


Fig. 16.3 Histogram of CT images from the dummy human head showing the grey level of the black background, of the cavities, and the structures of the head

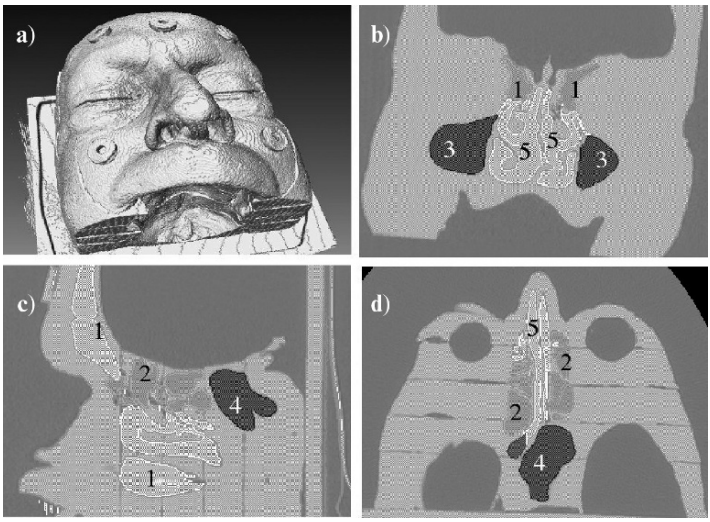


Fig. 16.4 Dummy head (**a**), segmentation in frontal (**b**), lateral (**c**) and transversal view (**d**). Frontal (1), ethmoidal (2), maxillary (3) and sphenoidal sinuses (4) and nasal cavity (5)

16.2.3 Paranasal Sinuses Volume Measurements

The measurement of paranasal sinuses volume using the 3D reconstructions of CT images was performed using the tool tissue statistics from Amira™ 4.1. For each sinus the number of voxels which belongs to this sinus is counted and the numbers of voxels are multiplied by the volume of one voxel in cubic millimeters.

To perform the direct measurement of paranasal sinuses volume we have used a method considered to be reliable and accurate described for Uchida et al. [24]. The dummy human head, that is divided in five layers, was maintained closed in a box, than the nasal cavities and the sinuses were filled with molding plaster

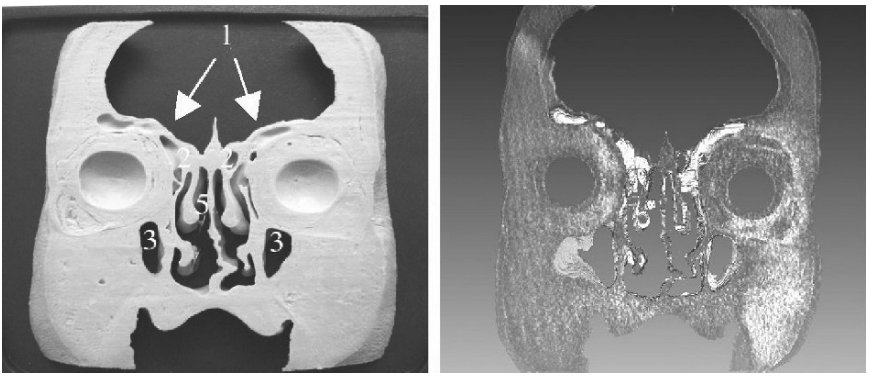


Fig. 16.5 Frontal view of one layer of the dummy human head (left) and the partial 3D surface reconstructions of the nasal cavity and paranasal sinuses (right). The numbered structures are: frontal (1), ethmoidal (2), maxillary (3) and sphenoidal sinuses (4) and nasal cavity (5)

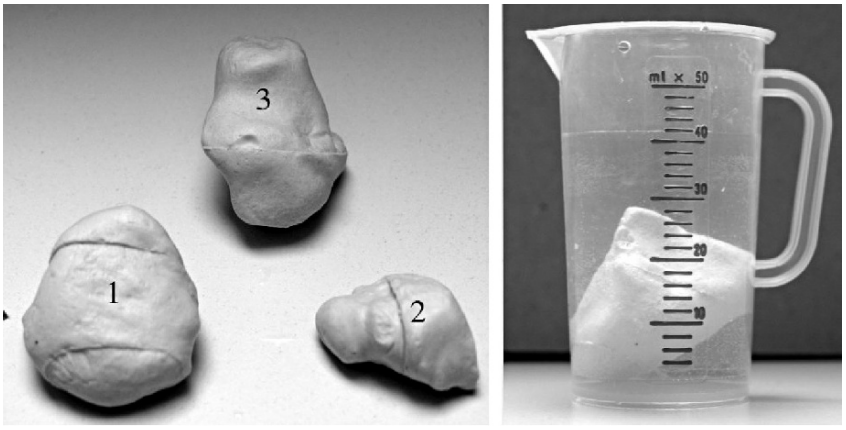


Fig. 16.6 Left side: solid casts of right (1) and left (2) maxillary and right (3) sphenoid sinuses of a human dummy head. Right side: graduated cup filled with water used to perform volume measurement of the casts of the sinuses through immersion in water

(Model Gips Krone, Osterode, Germany) using a syringe positioned in the left nostril. After 1 h, the impression material had hardened, the dummy human head was opened, and the solid cast of the sinuses were easily removed (Fig. 16.6). To measure the volume, the casts were immersed in a graduated cup filled with water (Fig. 16.6). The reproducibility of this method was assessed by repeating measurements. The cast of the ethmoidal sinuses as well as, the left sphenoid were despised.

16.3 Results

16.3.1 3D Paranasal Sinuses Reconstruction

The 3D-reconstruction of paranasal sinus CT images of the dummy human head is shown in Fig. 16.7 in frontal (a), lateral (b) and transversal (c) view.

Ostia are small orifices that connect the sinuses to the nasal cavity. If the sinus ostium is blocked, the entire sinus thus becomes the pathologic cavity (sinusitis). Figure 16.8 shows the maxillary and sphenoidal ostium.

16.3.2 Estimated Paranasal Sinuses Volume

The volume of paranasal sinuses measured by both 3D-reconstruction from CT images and direct measurements are given in Table 16.1 and Fig. 16.9.

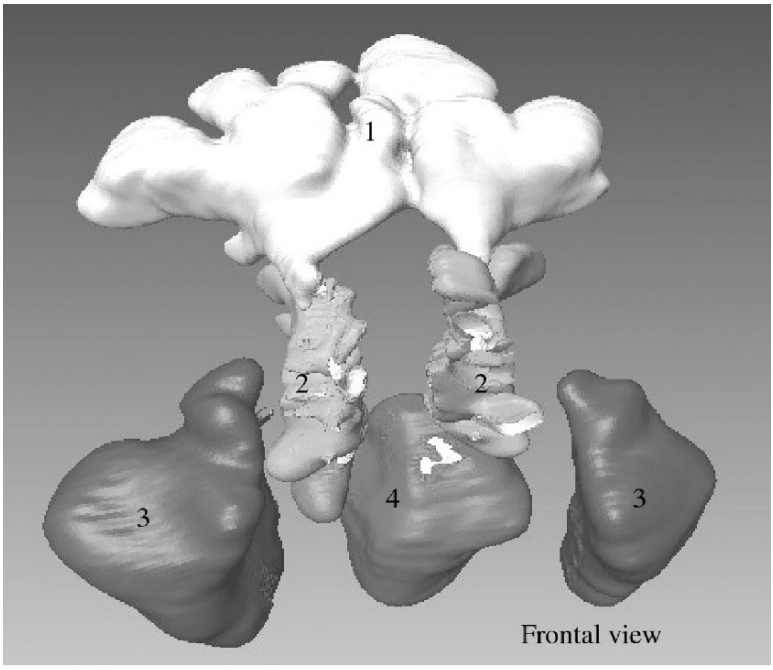


Fig. 16.7a Three-dimensional-reconstructions of paranasal sinus from CT images using Amira software. Frontal (1), ethmoidal (2), maxillary (3) and sphenoidal sinuses (4)

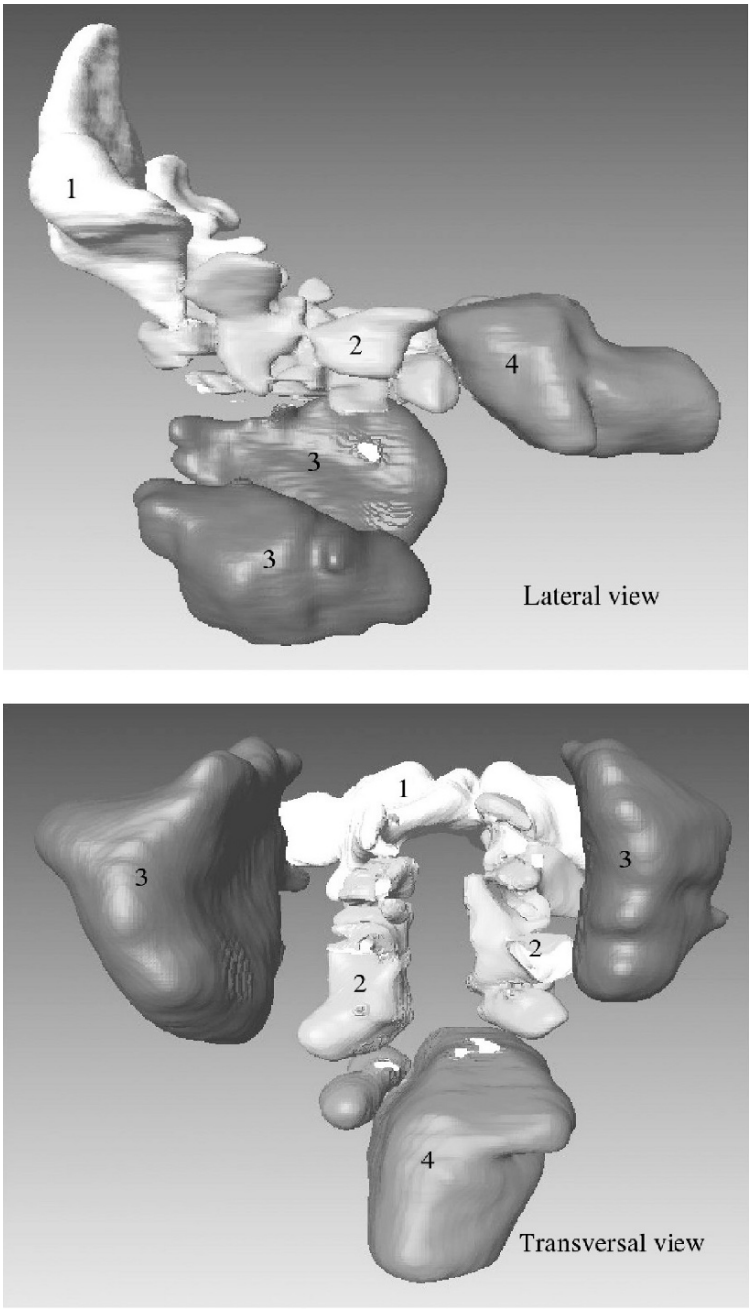


Fig. 16.7b Three-dimensional-reconstructions of paranasal sinus from CT images using Amira software. Frontal (1), ethmoidal (2), maxillary (3) and sphenoidal sinuses (4)

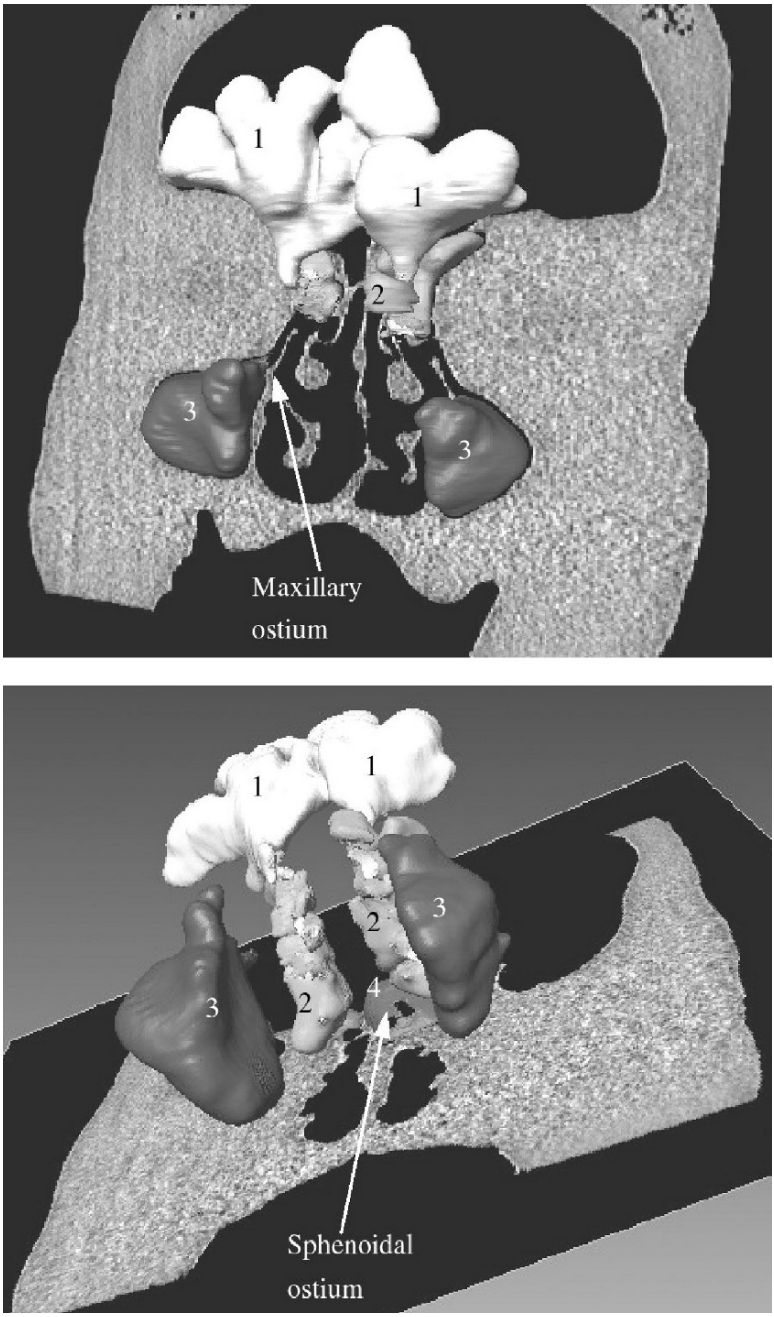


Fig. 16.8 The numbered sinuses are: frontal (1), ethmoidal (2), maxillary (3) and sphenoidal sinuses (4)

Table 16.1 Estimated volumes of the paranasal sinuses by both 3D-reconstruction from CT images and direct measurements

Paranasal sinuses		Estimated volume (ml)	
		3D-reconstruction	Direct measurement
Maxillary	Right	12.85	12.2
	Left	5.46	5.8
	Total	*18.32*	*18*
Sphenoidal	Right	10.96	11.6
	Left	0.80	N/A
	Total	*11.76*	*11.6*
Ethmoidal	Right	3.02	N/A
	Left	3.00	N/A
	Total	*6.02*	*N/A*
Frontal	*Total*	*12.04*	*11.8*

N/A: not available

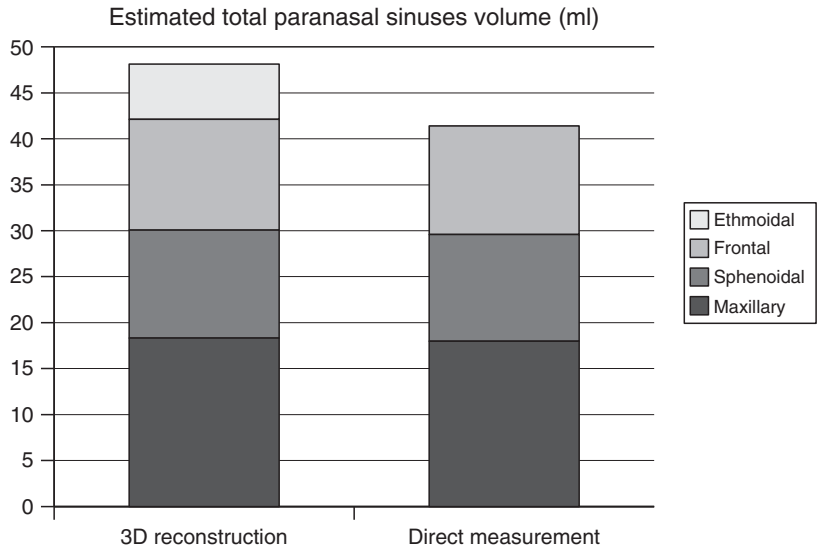


Fig. 16.9 Estimated total paranasal sinuses volume

16.3.3 Paranasal Sinuses Cavities Mesh Generation

Figure 16.10 shows the 3D mesh generated from the dummy human head without the paranasal sinuses volume.

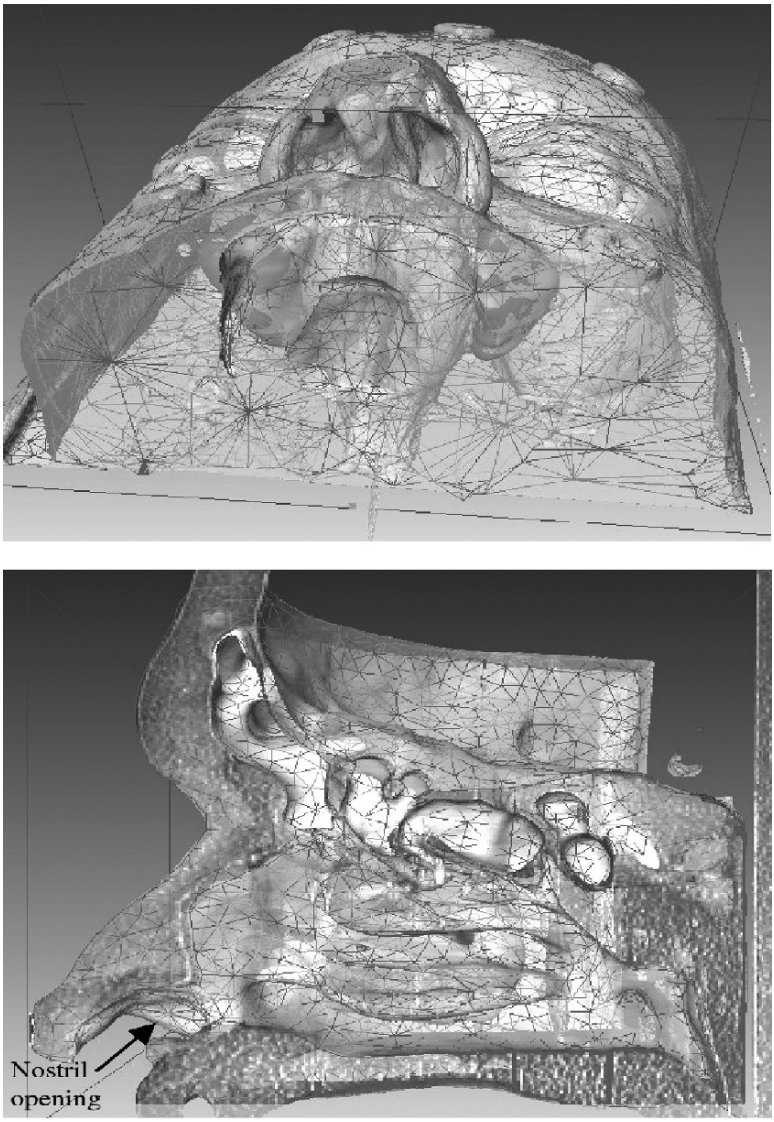


Fig. 16.10 Transversal (top) and lateral (bottom) view of the 3D mesh from the dummy human head without the paranasal sinuses volume

16.4 Discussion

Through this work we have evaluated an approach to obtain the geometric anatomy of the paranasal sinuses from CT images. Three-dimensional-representation of the head without these structures was transformed into tetrahedral meshes and will be used for the FE model that will allow robotic assisted endonasal surgery simulations.

Our modelling hypothesis was to consider the paranasal sinuses as a homogeneous, linear, isotropic and elastostatic material. For FE modeling it is very important the accurate estimation of the volume sinuses regions, because of the anatomical variation among individuals. Our results show the great asymmetry between left and right sides. For example, for the sphenoid and maxillary sinuses, a huge variation was seen between the right and left cavities (Table 16.1 and Fig. 16.9).

The results obtained from segmentation and 3D-reconstruction from paranasal sinuses CT images allow us to show the 3D morphologies of the sinuses from any viewpoint (Fig. 16.7). Moreover, we could also identify the ostia that communicate nasal cavity and paranasal sinuses, as the maxillary and sphenoidal ostiums (Fig. 16.8).

The cavity volume of paranasal sinuses is an important index for the paranasal sinuses evaluation [5]. Paranasal sinuses volumes have conventionally been measured using cadavers by directly injecting a variety of material [1, 24], while others employed CT images [2, 5, 18]. In the current study paranasal sinuses volumes were estimated using commercial software (Amira). The accuracy of the 3D-recosntruction is an important prerequisite for a precise FE model of the nasal structures and it depends of the segmentation approach when the inner boundary of the paranasal cavity is semi-automatically traced. To validate the accuracy of the 3D reconstructions it was performed a comparison of these measurements with the real size of dummy human head sinuses (Table 16.1).

It was very difficult to guarantee full injection of the impression material in the ethmoidal sinuses, due to their anatomical complexity (Fig. 16.7). Moreover, the frontal sinus was damaged when attempting to remove it. For this reason, these casts were not used for the comparison. The morphology and the values of the calculated sinuses volumes did not significantly differ from those obtained by other studies. The volumes were all in the respective ranges compared with previous reports (Table 16.2 and Fig. 16.9).

The generation of a FE mesh is critical since it affects both the accuracy and cost of subsequent numerical simulations. As shown in Fig. 16.9, the paranasal sinuses geometry created with Amira software is simple enough to allow conversion into a mesh of finite elements that could be then later loaded by external forces. However, a refinement of the FE mesh to more accurately capture original anatomical detail is desirable. Currently, we have applied with success this approach for segmentation and 3D-reconstruction of sinuses cavities of *ex vivo* [21] and *in vivo* [13] CT images.

16.5 Conclusion

In this study, the approach proposed to obtain a precise geometric reconstruction of the human paranasal sinuses from CT images was evaluated. The results show accurate comparisons between volume estimation of the paranasal sinuses from an anatomical model by material injection and by 3D CT images. From the 3D-reconstruction of the sinuses regions, was obtained a simplified homogeneous

Table 16.2 Previous reports about the human paranasal sinuses volume

Paranasal sinuses	Reference	Average* (ml)
Maxillary	[7]	10.1
	[24]	11.3 ± 4.60
	[16]	12.5
	[2]	14.7 ± 6.33
	[26]	15.1
	[23]	15.6
	[1]	23.5
	[18]	24.76
	[5]	44.2 ± 6.6
Sphenoidal	[26]	4.1
	[23]	4.1
	[7]	4.4
	[16]	4.5
	[5]	15.4 ± 6.9
Ethmoidal	[14]	4.9
	[7]	5.3
	[5]	12.5 ± 1.6
Frontal	[7]	2.8
	[16]	3.5
	[5]	8.1 ± 5.1

* Right + left sinuses

model that resembled the general complex geometry of the paranasal sinuses cavities for FE models. It will to be used as a background for the creation of further FEM models that should incorporate more structural features, such as inhomogeneity, different anatomical variations or pathological conditions.

References

1. Anagnostopoulous S, Verieratos D, Spyropoulos N (1991) Classification of human maxillar sinuses according to their geometric features. Anat. Anz. 173(3): 121–130.
2. Ariji Y, Kuroki T, Moriguchi S, et al. (1994) Age changes in the volume of the human maxillary sinus: a study using computerized tomography. Dentomaxillofac. Radiol. 23: 163.
3. Becker, DG (2003) The minimally invasive, endoscopic approach to sinus surgery. J. Long-Term Eff. Med. Implants 13(3): 207–221.
4. Eichhorn KW, Tingelhoff T, Wagner I, Westphal R, Rilk M, Kunkel ME, Wahl FM, Bootz F (2008) Evaluation of force data with a force/torque sensor during FESS. A step towards robot-assisted surgery. (in German) HNO Journal 56, n. 8, 789–794. ISSN 0017-6192 (Print) 1433–1458 (Online).
5. Kawarai Y, Kkunihiro F, Teruhiro O, Kazunori N, Mehmet G, Masaaki F, Yu M (1999) Volume quantification of healthy paranasal cavity by three-dimensional CT imaging. Acta Otolaryngol. 119(1), Supplement 540: 45–49.
6. Kennedy DW, Zinreich S, Rosenbaum AE, Johns M (1985) Functional endoscopic sinus surgery: theory and diagnostic evaluation. Arch. Otolaryngol. 11: 576–582.

7. Kikkawa Y (1942) Imaging for ethmoid sinusitis. Practica Otologica 37: 701–762.
8. King JM, Caladarelli DD, Pigato JB (1994) A review of revision functional sinus surgery. Laryngoscope 104: 4004–4008.
9. Kunkel ME, Moral AI, Rode D, Rilk ME, Wahl FM (2007) Towards a FEM model of the nasal cavity and paranasal sinuses for robot assisted endoscopic sinus surgery. The Finite Element Method in Biomedical Engineering, Biomechanics and Related Fields, 14th FEM Workshop, Ulm, Germany, 2007. ISBN: 978-3-9806183-9-7. pp. 8–19.
10. Kunkel ME, Tingelhoff K, Moral AI, Rilk M, Wagner I, Eichhorn KWG, Wahl FM, Bootz F (2007) Relationships between the internal carotid arteries and the sphenoidal sinuses assessed by 3D reconstruction and biomechanical tests: an analysis for Robot Assisted Endonasal Surgery. 6. Annual Meeting of the Deutsch Society of Computer and Robot Aided Surgery CURAC 2007. Karlsruhe, Germany, pp. 7–10.
11. Lang J (1989) Clinical anatomy of nose, nasal cavity and paranasal sinuses. New York: Thieme Medical Publishers.
12. Manestar D, Manestar M, Groscurth P (2006) 3D-model of the nose and paranasal sinuses. Anatomy in coronal sections and corresponding CT-images. Tuttlingen, Endo-Press.
13. Moral AI, Kunkel ME, Tingelhoff K, Rilk M, Wagner I, KWG Eichhorn, Bootz F, Wahl FM (2007) 3D endoscopic approach for endonasal sinus surgery 29th Annual International Conference of the IEEE, Engineering in Medicine and Biology Society EMBS 2007, Lyon, France, pp.4683–4686. ISBN: 978-1-4244-0788-3 ISSN:1557-170X.
14. Murai Y (1937) Anatomical structure of sphenoid sinus in Japanese. J. Med. (Tokyo): 56: 2267–2388.
15. Navarro JAC (2001) The nasal cavity and paranasal sinuses. Berlin, Springer.
16. Oosugi K (1923) Anatomical study of nose in Japanese. II. Nose. Practica Otologica 15: 1–300.
17. Pérez-Pinas I, Sabaté J, Carmona A, et al. (2000) Anatomical variations in the human paranasal sinus region studied by CT. J. Anat. 197: 221–227.
18. Shi H, Scarfe WC, Farman AG (2006) Maxilary sinus 3D segmentation and reconstruction from cone beam CT data sets. Int. J. CARS. 1: 83–89.
19. Stammberger H, Posawetz W (1990) Functional endoscopic sinus surgery. Concept, indications and results of the Messerklinger technique. Eur. Arch. Otorhinolaryngol. 247: 63–76.
20. Strauß G, Fischer M, Meixensberger J, Falk V, Trantakis C, Winkler D, Bootz F, Burgert O, Dietz A, Lemke HU (2005) Bestimmung der Effizienz von intraoperativer technologie: Workflow-Analyse am Beispiel der endoskopischen Nasennebenhöhlenchirurgie. HNO Springer Medizin Verlag 54: 528–535.
21. Tingelhoff K, Moral AI, Kunkel ME, Rilk M, Wagner I, Eichhorn KWG, Wahl FM, Bootz F (2007) Comparison between manual and semi-automatic segmentation of nasal cavity and paranasal sinuses from CT images 29th Annual International Conference of the IEEE, Engineering in Medicine and Biology Society EMBS 2007, Lyon, France, 2007, pp. 5505–5508. ISBN: 978-1-4244-0787-3 ISSN:1557-170X.
22. Tingelhoff K, Eichhorn KW, Wagner I, Kunkel ME, Moral AI, Wahl FE, Bootz F (2008) Analysis of manual segmentation in paranasal CT images. European Archives of Oto-Rhino-Laryngology. 265, n.9, 1061–1070. ISSN 0937–4477 (Print) 1434–4726 (Online).
23. Toida N (1937) Study of nasal cavity in Chinese. J. Med. Mantura 27: 743–752.
24. Uchida Y, Goto M, Katsuki T, Akiyoshi T (1998) A cadaveric study of maxillary sinus size as an aid in bone grafting of the maxillary sinus floor. J. Oral Maxillofac. Surg. 56: 1158–1163.
25. Wagner I, Tingelhoff K, Westphal R, Kunkel ME, Wahl FM, Bootz F, Eichhorn K (2008) Ex vivo evaluation of force data and tissue elasticity for robot-assisted FESS. European Archives of Oto-Rhino-Laryngology. 18 Mar 2008. [Epub ahead of print]. ISSN 0937-4477 (Print) 1434–4726 (Online).
26. Yoshinaga T (1915) Anatomical study of sphenoid sinus in Japanese. J. Med. Tokyo 29: 397–437.
27. Zinreich SJ, Kennedy DW, Rosenbaum AE, et al. (1987) Paranasal sinuses: CT imaging requirements or endoscopic surgery. Radiology 163: 769–775.

Index