Topological Data Structures for Surfaces

An Introduction to Geographical Information Science

Editor

SANJAY RANA

Centre for Advanced Spatial Analysis
University College London

John Wiley & Sons, Ltd

Topological Data Structures for Surfaces

Topological Data Structures for Surfaces

An Introduction to Geographical Information Science

Editor

SANJAY RANA

Centre for Advanced Spatial Analysis
University College London

John Wiley & Sons, Ltd

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester,
West Sussex PO19 8SQ, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wileyeurope.com or www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data

Topological data structures for surfaces : an introduction to geographical information science / Sanjay Rana, editor.
p. cm.
Includes bibliographical references and index.
ISBN 0-470-85151-1 (cloth : alk. paper)
1. Geographic information systems. 2. Information storage and retrieval systems–Geography. 3. Geography–Data processing. I. Rana, Sanjay.
G70.212.T675 2004
910′.285–dc22

2004005081

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-470-85151-1

Typeset in 10/12pt Times by Laserwords Private Limited, Chennai, India
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire
This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

Contents

List of Contributors

Chandrajit Bajaj Center for Computational Visualisation, CS & ICES, ACES 2.324A, University of Texas at Austin, Austin, TX 78712, USA.

Silvia Biasotti Istituto per la Matematica Applicata e le Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Via De Marini 6 – 16149, Genova, Italy.

Jason Dykes Department of Information Science, City University, Northampton Square, London, EC1V 0HB, UK.

Thomas Ertl Visualisation and Interactive Systems Group (VIS), University of Stuttgart, Universitaetsstrasse 38, 70569 Stuttgart, Germany.

Bianca Falcidieno Istituto per la Matematica Applicata e le Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Via De Marini 6 – 16149, Genova, Italy.

Martin Kraus School of Electrical and Computer Engineering, Purdue University, Box 220, 465 Northwestern Avenue, West Lafayette, IN 47907-2035, USA.

Marc van Kreveld Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands.

Atsushi Masuyama Department of Urban Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.

Jeremy Morley Department of Geomatic Engineering, University College London, Gower Street, London WC1E 6BT, UK.

Atsuyuki Okabe Center for Spatial Information Science, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.

René van Oostrum Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands.

Valerio Pascucci Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Box 808, L-560, Livermore, CA 94551, USA.

John Pfaltz Department of Computer Science, Thornton Hall, University of Virginia, Charlottesville, VA 22903, USA.

Sanjay Rana Centre for Advanced Spatial Analysis, University College London, 1–19 Torrington Place, London WC1E 6BT, UK.

Dan Schikore Computational Engineering International, 2166 N. Salem Street, Suite 101 Apex, North Carolina, NC 27523-6456, USA.

Bernhard Schneider Department of Geography and Department of Earth Sciences, University of Basel, Bernoullistrasse 32, H-4056 Basel, Switzerland.

Paul J. Scott Taylor Hobson Limited, 2 New Star Road, Leicester LE4 9JQ, UK.

Michela Spagnuolo Istituto per la Matematica Applicata e le Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Via De Marini 6-16149, Genova, Italy.

Shigeo Takahashi Department of Graphics and Computer Science, Graduate School of Arts and Sciences, University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo, 153-8902, Japan.

Gert W. Wolf Department of Geography, University of Klagenfurt, Universitaetsstrasse 65, A-9010 Klagenfurt, Austria.

Jo Wood Department of Information Science, City University, Northampton Square, London, EC1V 0HB, UK.

Foreword

Shape is fundamental to man and to mathematics. From the delineation of early Egyptian fields to Euclid and Pythagoras to current geometers and geographers, the notion of shape is fundamental. And as mathematicians, we prefer regular shapes; triangles, rectangles, and other convex polygons; circles, parabolas, and other conic sections, because we can compactly describe them in functional notation and reason about them.

But many shapes that are of practical importance to us are not regular. The topography of the earth is one. How do we describe the *shape* of the earth's surface? Abstractly we can say it is round or, more accurately, ovate. But that description is of little practical use to those of us living on the surface. Indeed for much of history, the global structure was regarded as flat, with no practical consequences to the lives of men. But, the presence of hills, ridges, and rivers is of great importance. The most common way to describe these, and other spatial phenomena, is with a map.

Ancient cartographers commonly *described* mountain ranges with inverted V's. These denoted the presence of mountains but did little to describe their shape. The use of contours as a descriptive tool did not emerge until the last 100 years or so. Yet even these are difficult to reason because they too are irregular shapes, but of one lower dimension.

Use of computer visualisation to describe the shape of surface terrain has made enormous strides in the last three decades. We can project triangulated surfaces; we can rotate them and zoom in for different perspectives. But in spite of the extraordinary intuitive comprehension that such visual descriptions provide, we still cannot reason about them.

Of course, we can create a dense array of elevation data. Now we can make assertions regarding minimal and maximal values. By assuming an interpolated, smooth, and differential surface, we can also define local properties such as slope and curvature. But neither of these seems to adequately describe spatial *shape* that tends to be global in nature.

The introduction of surface networks and Reeb graphs represents a more recent effort to describe the shape of a surface in a way that relates surface elements that may be rather distant. These structures are discrete, compact descriptions, and thus more amenable to logical analysis. Much of this will become evident as you peruse the following chapters. By collecting together in one place selections from the leading researchers in this exciting field, the editor, Sanjay Rana, has provided a significant service to all practitioners, not just geographers, who deal with spatial shape. Read carefully and you may find yourself seeing our familiar concepts of topological shape in a very different light.

This field probably began in 1870 with James Clark Maxwell's musing about *Hills and Dales*. But his approach was largely intuitive. The real formalisation was initiated by Marston Morse in a series of articles beginning in 1925 with the publication of *Relations Between the Critical Points of a Real Function of n Independent Variables*. If you become intrigued with the fascinating applications of Morse theory presented in this book, I urge you to read either this seminal paper or one of the more accessible books by Milnor, *Morse Theory* (1963) or Morse and Cairns (1969). Both indicate how mathematically rich this approach can be, particularly as we seek to describe shape in higher dimensions.

Morse theory was taken up by researchers in various fields in the 1970s. I believe that I, a computer scientist, coined the term *surface network*. But the concepts appeared in other disciplines as well. For example, C. K. Johnson (1977) wrote of *Peaks, Passes, Pales, and Pits* in crystallographic density maps. However, it was the quantitative geographers who really embraced these concepts. Already by the 1960s, William Warntz had introduced the essential ideas on the basis of Maxwell's and Cayley's papers. Warntz was modelling the *Topology of Socio-Economic Terrain and Spatial Flows* (1966). And the application of surface networks geographic terrain was a natural. I still recall the wonderful discussions at various conferences with early workers such as David Mark, Michael Woldenberg, Tom Peucker, Carrol Johnson, Waldo Tobler, and of course, Warntz himself regarding effective ways of modelling terrain. Not all felt surface networks were the answer; but all were conversant with its principles.

While we naturally associate two-dimensional surfaces with the surface topography around us because of its intimate involvement with our everyday life, it is better to approach the topic thinking in terms of general functions $f(x_1, \ldots, x_n)$ of n variables. After all, you will encounter applications in this book as different as the surface shape of a grinding wheel to the shape of three-dimensional volumes comprising body cavities. Neither has the shape characteristics of a water-eroded topography.

To ensure reasonable shape properties, most, but not all, of the following authors will require that the abstract surface $f(x_1, \ldots, x_n)$ be twice differentiable and that the Hessian matrix of partial derivatives be everywhere non-singular. Wow! These appear to be powerful constraints. But we find that in many applications, such conditions accurately reflect the true surface, or a close approximation of it. Thus, we ensure mathematical tractability. But as powerful as the calculus of partial differentiation is at capturing local behaviour, it is notoriously weak at capturing global behaviours. The identification of minima and maxima seem only to tell part of the story.

The genius, I think, of Maxwell and Morse was to focus on saddle points (or passes, or critical points, or points of equilibrium) where the gradient is also zero, rather than on just the extrema. These saddle points seem to be the key to describing topological shape in the large. Think of the role of passes in planning a transcontinental railroad. As you read on, you will see the pivotal role that saddle points, or whatever name the author uses, play in surface networks and Reeb graphs.

Surface networks and Reeb graphs provide a language for describing spatial shape. As with any descriptive tool or language, there are several universal themes that must be considered. First, what kinds of assertions can be made using the structures of the descriptive system? Do they help us to understand or manipulate the underlying phenomena? Second, how much compression does the descriptive system provide? Are

the important features of the underlying phenomena efficiently conveyed? Third, can we abstract within the system? That is, can we discern, within the system alone, more important features and create faithful descriptions that eliminate less important details? Clearly, this is related to the second theme, yet is still quite different. Finally, does the system support a description of phenomena change? Few systems, or languages, do. Almost all our formal systems of expression and thinking are concerned only with a description of state. They presume a static phenomenon. Perhaps only the calculus is really concerned with describing change.

As you read the individual contributions in this book, you should be forcibly struck, even as I was, by the creativity and imagination of the authors as they tackle these universal themes. They have not found all the answers to the thorny question of describing irregular shapes. However, they have laid a firm basis on which to build.

Finally, as one of those who made a small contribution at the beginning of this research area, I would like to make a prediction regarding its future. Describing the global shape of a functional surface is difficult. Describing how it changes, in the large, over time is orders of magnitude more difficult, but often more important. From the movement of offshore barrier islands to the spatial spread of an epidemic, it is the nature of the change that must be described if we are to understand the forces causing the change. It is here that surface networks, I believe, have their greatest potential.

The importance of graph-like descriptions such as surface networks have been largely overlooked, I believe, because so much of the rich detail found in a typical functional surface has been abstracted away. It is a bit like representing the human body with a stick figure. Only very small children do that. But stick figures have been fundamental in describing and analysing human movement. How does the knee joint move relative to the hip when a child is skipping? How does the saddle of a sand bar rotate relative to the peak in a north-east storm?

If my conjecture is correct, there is a wide-open field of inquiry that should be explored. The wealth of insight offered by the contributions of this book should be of great value in any such exploration. They will certainly be of interest to anyone concerned with the representation of irregular shapes.

John L. Pfaltz
Charlottesville, VA

Preface

The idea of the book was born out of my long search for a starting point of references on surface network that could put me in the right direction during the initial days of the PhD. During the MSc (GIS) dissertation on surface network, I found a number of interesting and novel challenges in the topic. My literature survey during the MSc (GIS) dissertation was very limited. Thus before embarking on a brainstorming on the unresolved issues I decided to do further literature review, somewhat hoping that I would not find much material, thus strengthening the novelty of my doctoral research. To my surprise, I started to find many research works mostly from computer science. The discovery of the literature in computer science was exciting and also revealed many duplicate researches and multiple terminologies for the same concept. In mid 2001, I realised that a book that presented all this diverse literature could be a good idea for future researchers.

I do realise that this book has a relatively narrow focus. But, I think the diversity of the applications of topological surface data structures and the related disciplines will justify the efforts in putting this book together. The book is primarily about the topological data structure for continuous surfaces called *surface network*. However, two other significant related data structures called *Reeb graphs* and *contour trees* are also included to broaden the scope of the book. The book is divided into two parts. The first part contains chapters that define the topological surface data structures and explain the various automated methods for generating these data structures. The second part demonstrates a number of applications of surface networks in diverse fields ranging from sub-atomic particle collision visualisation to the study of population density patterns. Most of the chapters in the book are based on previously established prominent research works on surface networks. The age of the research works vary from the 1980s to 2003. The authors of the chapters are mostly from geographic information science and computer science in which the most research on surface networks has taken place.

Despite the broad background of the authors, I was very fortunate to get positive responses from all the chapter authors. I would like to thank all chapter authors who despite their hectic schedule agreed to spend their time in preparing the manuscripts for the book within the prescribed timetable. I am also grateful to Paul Longley for his generosity in approaching Wiley on my behalf and suggesting the idea of the book to Wiley Publishers, John Pfaltz for writing the foreword for the book and supplying both materials and intellectual support during the write-up of the work, Shigeo Takahashi

for his excellent hospitality and Japanese pizza during my visit to his laboratory in Tokyo, and Mike Batty and Jo Wood for their comments on the book proposal. Finally, I want to thank Wiley Publishers Book Editors, Lyn Roberts and Keily Larkins, for providing me the opportunity to put the book together and for their patience despite a month's delay in the submission of the manuscript.

Sanjay Rana
London

Introduction

1

Introduction

Sanjay Rana

1.1 EVERY *THING* HAS A SURFACE

A surface is the most fundamental shape of matter to us. Surfaces surround us in various forms, ranging from the undulating ground we stand on to this flat page in the book, so much so that the recognition of surfaces and their structure is crucial to our daily life. In addition, a number of non-physical spatial phenomena such as temperature and population density are also modelled as surfaces to aid visualisation and analyses. Despite the wide variety, conceptually the description of even the most complex surface forms is rather simple. Basic surface descriptors such as circle, box, flat, convex, and others can be combined together to derive any arbitrary shape, thus enabling the computer graphics animators in Hollywood to reconstruct dinosaurs.

It is fascinating to appreciate how different disciplines describe surfaces. It is also worthwhile to highlight the issues in surface representation as it reveals the level of abstraction when the increasingly massive surface datasets are stored in computers. Mathematicians have modelled surfaces primarily with an aim to decompose the surface into the *basic descriptors* or elements even if it meant oversimplification (e.g. by using the primary surface elements mentioned above), leading to potential loss of the structure. Such descriptions are generic (i.e. universal to all types of surfaces), formal, and robust, such as that required in computer-aided designing. The aim of the mathematical description is to produce a constrained *global model* of the surface. The other large group of surface researchers from the field of physical geography use more *compound descriptors* (e.g. valleys, mounds, scarps, drainage network) with more emphasis on the preservation of the structural information of the surface. Although the *compound descriptors* are more *natural*, their relevance

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

is subjective to each individual; hence it is often difficult to derive an objective definition of surface features[1]. These researchers are more interested in the process that resulted in the surface; hence the descriptors are also symbolic of the factors in the process.

A relevant example of such fundamental dichotomy is the description of a terrain by these two disciplines. In order to achieve a simple and tractable model of terrain, a typical algebraic definition of terrain will be as follows:

> *A terrain is a smooth, doubly continuous function of the form* $z = f(x, y)$*, where* z *is the height associated with each point* (x, y)*.*

Further,

> *The local maxima or peak of the terrain is a point with a zero slope and a convex curvature.*

Other terrain features are defined similarly using morphometric measures. Most physical geographers will, however, find these definitions very restrictive because (i) they assume away the absence of some common terrain features such as lakes and overhangs crucial to certain applications, for example, runoff modelling; and (ii) natural terrain features cannot be localised to a point because a peak with zero local slope will really be the exception rather than the rule in nature. In physical geography, the description of the terrain surface and terrain features is more indicative than precise. Therefore, as long as the shape of the terrain around *an area* could be classified into terrain feature type, it is the responsibility of the geographer to locate the position of the peak on terrain based on his/her expertise. Figure 1.1 shows the difference between an algebraic topology (as a triangulated irregular network (TIN)) and physical geography (drainage network) description of a terrain surface.

It therefore follows that a combination of these two ways of describing a surface should provide a complete and robust approach for describing surfaces. In other words, a data structure that could explicitly describe both the structure (e.g. hills and valleys) and the form (e.g. xyz coordinates) of a surface will be an ideal digital representation of the surface. A more general form of this requirement was stated by Wolf (1993). Wolf regarded an efficient surface data structure to be one that contains both the geometrical information (e.g. coordinates, line equations) and the topological information on the geometrical data (e.g. neighbourhood relationships, adjacency relationships) of the surface. But as you read the book, it will be clear that the construction of a topologically consistent surface data structure is a non-trivial task because real surfaces seldom obey the constraints required by topological rules. At this stage, I would like to propose a difference between the terms *surface data structure* and *surface data model*. I think the term surface data structure should merely imply a format for storing the geometric and topological information (e.g. point heights and adjacency relationships) in a single construction. On the other hand, surface data model should be an extended version of

[1] Wolf (1993, p24) highlights the importance for exact definitions quoting Werner (1988) and Frank et al. (1986).

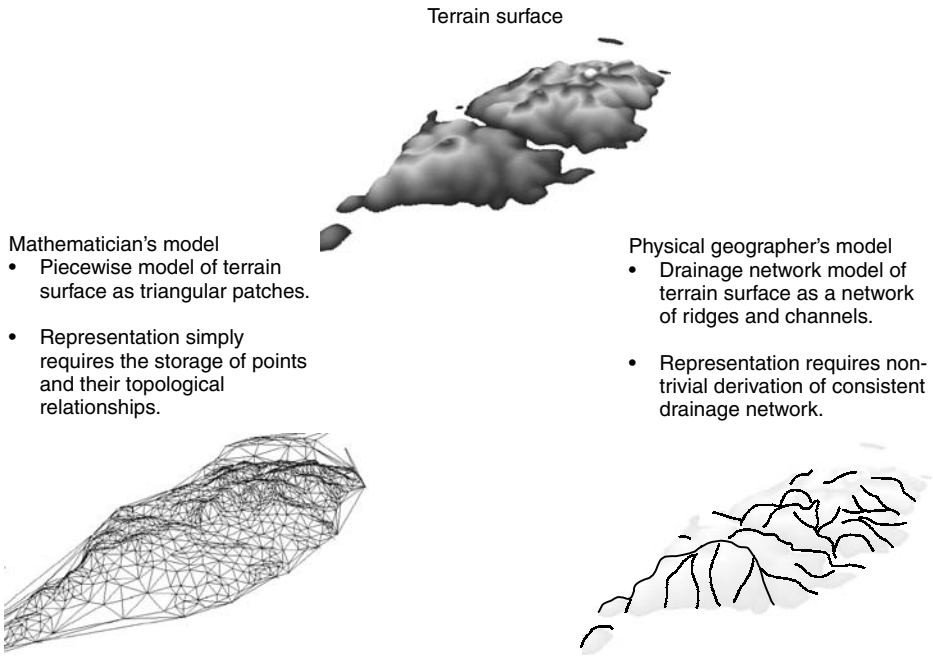

Figure 1.1 *Representation of the terrain surface into two different models depending upon the desired application*

the surface data structure in which additional metadata information characterising the surface (e.g. valleys, ridges, i.e. characteristic properties of surface) is also incorporated to produce a representation of the surface. In simple terms, a surface data model is a value-added product of the surface data structure and it explicitly represents the characteristics of the surface. Thus, all surface data models can be regarded as surface data structures but the opposite is not necessarily true.

1.2 TOPOLOGICAL DATA STRUCTURES FOR SURFACES

It is fairly straightforward to produce data structures that store the geometrical information about a surface. We simply need to collect certain points on the surface either on a regular lattice/grid or irregular locations. In fact, for many surface applications, only geometrical information is required for analyses. However, storing topological information has the following significant advantages:

- If we assume certain homogeneity in surface shape (e.g. smooth and continuous), using a topological data structure will reduce the number of points required to construct a surface. For example, by storing only certain morphologically important points (MIPs) (e.g. corners, inflexions) and their topological relationships, we could reconstruct the surface by means of interpolating between MIPs. Thus, the amount

of computer disk space required to store the surface will be reduced significantly. Helman and Hesselink (1991) reported 90% reduction in size on storing volumetric surface datasets in topological data structures.
- Topological relationships are a much more efficient way of accessing a spatial database, for example, sophisticated spatial queries such as clustering would be easily implemented on the basis of adjacency relationships.
- Topological data structures could provide a unified representation of the global structure of the surface. Thus, these data structures can be used for applications that require a uniform and controlled response from the entire surface such as morphing in computer graphics and erosion modelling in hydrology.
- Topological data structures will be useful for the visualisation of the structure of surfaces, particularly multi-dimensional surfaces. For example, Helman and Hesselink (1991) and Bajaj and Schikore (1996) reported that rendering of volumes surface datasets as skeleton-like topological data structures is more fast and comprehensible compared to traditional volume rendering.
- Bajaj and Schikore (1996) propose that topological data structures will be a simple mechanism for correlating and co-registering surfaces due to the embedded information on the structure of surfaces.

While the above benefits of topological data structures are applicable to all types of surfaces, it is uncertain which MIPs and topological relationships should constitute a universal surface topological data structure. Clearly, each surface should be characterised by MIPs suitable for a particular application. Many types of MIPs have been proposed by researchers in different disciplines and have been referred to by different names, for example, *landform elements* (Speight, 1976), *critical points and lines* (Pfaltz, 1976), *surface-specific features* (Fowler and Little, 1979), *symbolic surface features* (Palmer, 1984), *surface patches* (Feuchtwanger and Poiker, 1987) and *specific geomorphological elements* (Tang, 1992). The common aim of these classifications has been to provide a sufficient resemblance to the surface relevant to a particular application.

This book is primarily on the topological surface data structure called *surface network* (Pfaltz, 1976), which has been used in many disciplines because of its simple and universally applicable design. The book also discusses two other closely related data structures called the *Reeb graph* (Reeb, 1946) and the *contour tree* (Morse, 1968). I suppose some readers might be surprised to see the rather old lineage of the surface network. Surprisingly, however, these data structures have received little mention even in otherwise well-referenced texts (e.g. Koenderink and Van Doorn, 1998, Wilson and Gallant, 2000) despite a substantial, although I admit, irregular flow of research papers. This was indeed the main motivation behind this book. During the initial days when I was doing my Ph.D. on surface networks, I assumed that there was not much literature on surface networks, but, gradually, I started finding many works from all across the globe and from different disciplines, which was very encouraging. Hence, I decided to propose the book with the aim of putting together some key works on surface networks and the related data structures, so that future researchers could start from a single source.

Since each chapter in this book has a good introduction to the individual data structures, I will not define them here in detail. In this chapter, I will present instead the

interesting history related to the development of these data structures followed by an overview of the chapters. In simple and general terms, the Reeb graph, contour tree, and surface network are graph-based surface data structures whose vertices are the local peaks, local pits, and local passes. The edges of the graph are the channels, connecting pits to passes, and ridges, connecting passes to peaks. Peaks, passes, and pits are together called the *critical points* of a surface, and ridges and channels are together called the *critical lines* of a surface. In my opinion, all the above surface data structures also qualify as surface data models because their construction is very much based on surface elements. Theoretically, any n-dimensional, smooth and doubly continuous surface can be represented as a surface network (Pfaltz, 1976); however, the most common implementations are limited to two- and three-dimensional surfaces.

The primary origin of these data structures lies in the realisation of the critical points and critical lines of the surface. Critical points can be defined as characteristic local surface features that are common to all surfaces and contain sufficient information to construct the whole surface, thus taking away the need to store each point on the surface. Critical points have a local zero slope, that is, $\mathrm{d}z/\mathrm{d}x = \mathrm{d}z/\mathrm{d}y = 0$, and three such critical points of the surface are local maxima ($\partial^2 z/\partial x^2 > 0$, $\partial^2 z/\partial y^2 > 0$ (also called *peaks, summits*), local minima ($\partial^2 z/\partial x^2 < 0$, $\partial^2 z/\partial y^2 < 0$) (also called *pits, immits*), and passes ($\partial^2 z/\partial x^2 > 0$, $\partial^2 z/\partial y^2 < 0$ or vice versa) (also called *knots, saddles, bars*). These critical points have been referred to in physical geography as the *fundamental topographic features*. In physical geography, the derivation of these features has traditionally been based on the overall shape of contours (i.e. using a regional context) on a topographic map rather than local morphometric properties. For example, a peak is identified as the centre of a closed highest contour bounded by lower contours. At any point on the surface, a line following the steepest gradient is called a *slope line* (also called *topographic curves* and *gradient paths*). Critical lines are a special pair of slope lines that originate and terminate at critical points. There are two types of critical lines, namely ridge line and channel line. A ridge line originates from a peak and terminates at a pass, and a channel line starts from a pass and terminates at a pit.

The concept of *critical points* of a surface and critical lines was proposed as early as the mid-nineteenth century by the mathematicians (De Saint-Venant, 1852 reported by Koenderink and van Doorn, 1998; Reech, 1858 reported by Mark, 1977). In physical geography, Cayley (1859), on the basis of contour patterns, first proposed the subdivision of the topographic surface into a framework of peaks, pits, saddles, ridge lines, and channels. Maxwell (1870)[2] extended Cayley's model and proposed the following relation between the peaks, pits, and passes:

$$\text{peaks} + \text{pits} - \text{passes} = 2 \tag{1.1}$$

This relation was later proved by Morse (1925) using differential topology and is also known as the *mountaineer's equation* or the *Euler–Poincaré formula* (Griffiths, 1981 reported by Takahashi et al., 1995). Maxwell also described the partition of the

[2] The anxiety with which Maxwell presented his paper is quite amusing. His note to the editor of the journal reads "An exact knowledge of the first elements of physical geography, however, is so important, and loose notions on the subject are so prevalent, that I have no hesitation in sending you what you, I hope, will have no scruple in rejecting if you think it superfluous after what has been done by Professor Cayley".

topographic surface into hills (areas of terrain where all slope lines end at the same summit) and dales (areas of terrain where all slope lines end at the same immit) on the basis of the fundamental topographic features.

The earliest graph–theoretic representation of the topological relationships between the critical points of a terrain is the Reeb graph (Reeb, 1946; reported by Takahashi et al., 1995). Reeb graph basically represents the splitting and merging of equi-height contours (i.e. a cross section) of a surface as a graph. The vertices of the graph are the peaks, pits, and passes because the contours close at the pits and the peaks, and split at the passes. Consequently, the edges of the Reeb graph turn out to be the ridges and channels.

In a significant related development in mathematics, Morse (1925) derived the relationship between the numbers of critical points of sufficiently smooth functions (called *Morse functions* under certain constraints), which is known as the *Critical Point Theory* or *Morse Theory*. The generic nature and wide applicability of Morse Theory led to an expansion in the interest in the critical points of surfaces amongst various disciplines.

Warntz (1966) revived the interest of geographers and social science researchers in critical points and lines when he applied the Maxwell's "hills and dales" concept for socio-economic surfaces, referred to as the *Warntz network* (the term apparently coined by Mark, 1977).

A data structure identical to Reeb graph is the contour tree (Morse, 1968), also called the *surface tree,* by Wolf (1993). The contour tree represents the adjacency relations of contour loops. The treelike hierarchical structure develops because of the fact that each contour loop can enclose many other contour loops but it can itself be enclosed by only one contour loop. As is evident, the contour tree is the same as the Reeb graph except that it is separated by two decades. Kweon and Kanade (1994) proposed another similar idea called the *topographic change tree*. As in the case of the Reeb graph, the vertices of such a contour tree are the peaks, pits, and passes.

Pfaltz (1976) proposed a graph–theoretic representation of the Warntz network called *surface network* (Mark, 1977 used the term Pfaltz's graph). While the topology of the Pfaltz's graph was based on the Warntz network, Pfaltz added the constraint that the surface will have to be a Morse function. Since Pfaltz was in the computer science field, his work attracted the attention of researchers in three-dimensional surfaces such as in medical imaging, crystallography (e.g. Johnson et al., 1999, Shinagawa et al., 1991), and computer vision (e.g. Koenderink and Doorn, 1979). He also proposed a homomorphic contraction of the surface network graph to reduce the number of redundant and insignificant vertices. Along similar lines, Mark (1977) proposed a pruning of the contour tree to remove the nodes (representing contour loops) that do not form the critical points, i.e. the vertices of the contour tree, and called the resultant structure the *surface tree*. This essentially reduces the contour tree to the purely topological state of a Pfaltz's graph. It is easy to realise that the Reeb graph, Pfaltz's graph, and surface tree are fundamentally similar and are actually inter-convertible (Takahashi et al., 1995).

Wolf (1984) extended Pfaltz's graph by introducing more topological constraints in order that it be a consistent representation of the surface. He proposed assigning weights to the critical points and lines to indicate their importance in the surface and thus he proposed the name *weighted surface network* (WSN) for the Pfaltz's graph. He

demonstrated new weights-based criteria and methods for the contraction of the surface networks. Later, Wolf suggested that in order to visualise the WSN for cartographic purposes and to make it useful for spatial analyses, the vertices could be assigned metric coordinates (Wolf, 1990). The resultant representation is termed *metric surface network* (MSN).

Recent works have mostly focused on the automated extraction of surface networks from raster (Wood and Rana, 2000, Schneider, 2003) and TIN (Takahashi et al., 1995), which will be discussed in detail in the following chapters.

1.3 OBJECTIVES OF THE BOOK

As mentioned earlier, while there is an extensive literature on other topological surface data structures (e.g. TIN and quadtrees by Samet, 1990a, b, van Kreveld et al., 1997), the topics of surface network, contour tree, and Reeb graph, proposed more than three decades ago, have only had irregular and scattered reports of the research on them. This gap is the main motivation of this book. Despite the unique inter-disciplinary scope of these data structures, there is generally a lack of awareness about their complete potential amongst modern researchers. The book is also timely because publications demonstrating all the promised potential of these data structures for practical applications such as visualisation of large datasets (e.g. Takahashi et al., 1995, Bajaj and Schikore, 1996), fast contour extraction (e.g. van Kreveld et al., 1997), generalisation and compression of surfaces (e.g. Rana, 2000a,b, Kraus and Ertl, 2001), and spatial optimisation (e.g. Rana, 2003a, Kim et al., 2003) have finally started coming out.

The objective of this book is to bring together some key earlier and modern researches on these data structures to rejuvenate these topics and fuel ideas for future research. Some of the important features of this book are as follows:

- Popular morphometric feature extraction algorithms, useful in drainage analysis, computer vision, and information organisation, are described with practical examples with links to the directions for future research.
- A comprehensive and condensed treatment of these data structures, unpublished elsewhere, has been made available to the reader.
- This is a multidisciplinary area of research and this volume provides accessible content to practitioners in a range of fields.

1.4 OVERVIEW OF THE BOOK

The book is divided into two main parts. The first part deals with concepts, automated extraction, and issues related to the Reeb graph, contour tree, and surface network. The second part of the book presents a number of applications of these topological surface data structures.

The primary audience for this book are postgraduates and professionals. As can be clearly seen from the diverse background of the authors, this book will appeal to members of a number of disciplines such as Geographic Information Science, Computer Science, and other sciences involved in the morphometric analysis of surfaces.

Owing to the very practical nature of this book and its deliverables, I am tempted to believe that the commercial organisations, particularly those involved in research and development of solutions, would be keen to explore the ideas presented in this book. I feel that many of the ideas are still in their early phase of development with promising outputs, which could also provide topics for postgraduate and higher-level research.

All chapters contain a basic to intermediate discussion on the data structures; hence each chapter is quite self-contained and could be read independently. The following brief descriptions should give a general idea about each chapter so that the reader can decide to follow the book at his/her own pace and order.

1.4.1 Part I – Concepts and implementation

Part I starts with a chapter by Gert Wolf (Chapter 2), who extended the work of Pfaltz (1976). Wolf describes the construction of the surface network graph and how weights could be assigned with edges and nodes to indicate their importance for both the macro- and the micro-structure of the underlying surface in great and lucid detail. The graphs thus obtained – termed *weighted surface networks* (WSNs) – represent a powerful tool for characterising and generalising topographic surfaces. The technique of generalising surface networks using two graph–theoretic contractions is explained. The chapter then proposes an improvement of the purely topological WSNs, by attaching (xy)-coordinates to the nodes resulting in an MSN. Finally, the generalisation process of a real landscape taken from the Latschur region in Austria is shown.

Shigeo Takahashi (Chapter 3) describes algorithms for extracting surface network and Reeb graphs. The fully automated algorithms extract these data structures from an input TIN by simply generating a linear interpolation over the elevation samples instead of computationally expensive higher-order interpolations. Furthermore, the extracted features are correct in the sense that they maintain topological integrity (e.g. the Euler–Poincaré formula for critical points) inherited from the properties of smooth surfaces. The present algorithms are robust enough to handle troublesome datasets such as noisy or stepwise discrete samples, and such robustness is demonstrated with several experiments on real terrain datasets. The resultant configuration of critical points allows us to tackle other geographical information systems (GIS) related issues, which is also discussed in this chapter.

Bernhard Schneider and Jo Wood (Chapter 4) describe two methods for the identification of MSNs from raster terrain data. This chapter includes a discussion of some of the computational issues that have previously prevented automated construction of MSNs, and presents two new automated methods that may be applied to raster terrain data. The chapter first describes a simple and robust bilinear polynomial approximation method followed by an advanced bi-quadratic polynomial method.

Marc Kreveld and others (Chapter 5) describe an efficient method to construct the contour tree and to obtain seed sets that are provably small in size. The contour algorithm can be used for regular and irregular meshes.

Silvia Biasotti and others (Chapter 6) introduce and describe the concept of extended Reeb graph (ERG). First of all, a useful overview of the definition of critical points and Morse complexes for smooth manifolds is given. It is followed by a description

of some existing methods that extend these concepts to piecewise linear 2-manifolds, focusing, in particular, on topological structures as surface networks and quasi-Morse complexes, available for analysis and simplification of triangular meshes. To avoid the dependency of such structures on the locality of the critical point definition, they propose to consider critical areas and influence zones instead of usual ones, and give their formal definition. They base the ERG representation on this characterization and compare it with the surface network structure. Finally, they describe the ERG structure and its construction process, also introducing several examples from a real terrain from New Zealand.

1.4.2 Part II – Applications

This is perhaps the more interesting part of the book and has taken the most effort to put together.

The analysis of urban population distributions has been one of the central subjects of human geography since Clark's pioneering study in the 1950s (Clark 1951, 1958). Following Clark, most initial studies dealt with population distributions in terms of a population density with respect to the distance from the centre of a city (i.e. one-dimensional distributions). These studies were extended to two-dimensional distributions by use of the trend surface analysis. This analysis, however, has difficulty in interpreting the estimated coefficients. To overcome this difficulty, Okabe and Masuda (1984) proposed a method for analysing population surfaces in terms of surface networks. In the first chapter of this part, Atsuyuki Okabe and Atsushi Masuyama (Chapter 7) propose a new method for measuring topological similarity between activity surfaces in terms of modified counter trees. This method has an advantage in measuring not only local similarity but also global similarity. They develop an algorithm for implementing the proposed method and apply the method to urban population surfaces in Japan and show topological similarity among Japanese urban population surfaces.

In Chapter 8, Valerio Pascucci discusses a technique that reduces the user responsibility to infer implicit information present in the data, by computing topological features like maxima, minima, or saddle points and determining their relationships. He first introduces the formal framework, based on Morse theory and homology groups, necessary to analyse the critical points of a scalar field and to classify the shape of its level sets. He discusses a set of algorithms that map this mathematical formalism into an efficient pre-processing of the data. The chapter concludes with a discussion on user interfaces that present intuitively the computed information and a demonstration of examples from real scientific datasets such as subatomic particle collision.

Martin Kraus and Thomas Ertl (Chapter 9) show how to apply the concepts of scalar topology to the volume visualisation of structured meshes. This chapter discusses the role of topology-guided downsampling in direct and indirect volume visualisation. While most algorithms related to scalar topology work on unstructured meshes, topology-guided downsampling is a recently published downsampling algorithm for structured meshes, for example, Cartesian grids. The main goal of this technique is to preserve as many critical points as possible, that is, to preserve as much as possible of the topological structure of the original scalar field in the downsampled scalar field.

After presenting this downsampling algorithm, they discuss its application in indirect volume visualisation with isosurfaces and in direct volume rendering. Particular emphasis is made on the interplay with recent developments in direct volume rendering, namely the use of programmable per-pixel shading for pre-integrated volume rendering. They also show how to employ topology-guided downsampling in the generation of hierarchical volume data that can be rendered with the help of programmable per-pixel shading.

Jason Dykes and I (Chapter 10) present a surface networks–based features modelling approach for the visualisation of dynamic maps. This chapter extends the proposals of Valerio Pascucci (Chapter 8). Despite their aesthetic appeal and condensed nature, dynamic maps are often criticised for the lack of an effective information delivery and interactivity. We argue that the reasons for these observations could be due to their information-laden quality, lack of spatial and temporal continuity in the original map data, and a limited scope for a real-time interactivity. We demonstrate, with the examples of a temporal and an attribute series of a terrain and a socio-economic surface, respectively, how the re-expression of the maps as the surface network, spatial generalisation, morphing, graphic lag, and the brushing technique can augment the visualisation of dynamic maps.

Traditional surface texture parameters take a statistical approach to characterisation using only the height and location information of the individual measured points. Many applications, that is, lubrication, paintability of a surface, anodized extruded aluminium, and so on, require the characterization of features such as peaks, pits, saddle points, ridge lines, course lines, and so on, and the relationships between these features. That is a pattern recognition approach. In Chapter 11, Paul Scott proposes a topological characterisation of surface texture. This is based on the topological relationships between critical points and critical lines on the metrological surface and incorporated into a WSN. He then removes the predominant insignificant critical points, caused by measurement noise, and Gert Wolf's graph contractions (see Chapter 2), and then assesses the relationships between the significant surface features.

In Chapter 12, Jeremy Morley and I propose the advantages of using the fundamental topographic features forming the surface network of a terrain, namely the peaks, pits, passes, ridges, and channels, as the observers or the targets in visibility computation. We demonstrate that considerable time can be saved without any significant information loss by using the fundamental topographic features as observers and targets in the terrain. The optimisation is achieved because of a reduced number of observer–target pair comparisons, which we call the *Reduced Observers Strategy* and *Reduced Targets Strategy*. The method has been demonstrated for a gridded digital elevation model. Owing to this selected sampling of observers in the terrain, there is an underestimation of the viewshed of each point. Two simple methods for assessing this uncertainty have been proposed.

In the conclusion of the book, I raise a number of unresolved issues related to the data structure model, their automated generation, and generalisation.

Part I

Concepts and Implementations

2

Topographic Surfaces and Surface Networks

Gert W. Wolf

2.1 INTRODUCTION

Nowadays many GIS applications require the mapping and analysis of spatial data at widely differing scales. On the one hand, the accumulation of such a great body of information evidently demands sophisticated data structures in terms of spatial resolution and topology; on the other hand, it obviously requires intelligent, rule-based software packages for applying generalisation procedures in order to satisfy user queries and to provide adequate graphical output. Ideally, a single large-scale representation of the spatial data could be stored, and, as a result, smaller-scale versions could easily be derived from it. However, for the time being the automation of the necessary generalisation processes is not sufficiently well advanced and, as a consequence, multiple representations at different scales have to be maintained (Jones, 1991).

This chapter concentrates on two minute but nevertheless important aspects referring to the previously addressed and hitherto inadequately solved problems, namely the feature-based modelling and generalisation of topographic surfaces. While growing attention concerning this area of research has been aroused in geography and cartography by the rapid evolution of GIS, it is obvious that topographic surfaces are also of special interest for a variety of other disciplines, such as mathematics, computer graphics, meteorology, hydrology, and also for social sciences and economic sciences, to mention only a few. For this reason, this chapter will at first focus on the clarification of the term *topographic surface* in order to proceed with a formal discussion of some of its fundamental properties.

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

2.2 TOPOGRAPHIC SURFACES AND CRITICAL POINT THEORY

As Kweon and Kanade (1991, 1994) have pointed out, traditional natural language definitions of topographic features are ambiguous as they suffer from the substantial drawback that they either use terms that are not exactly defined but are assumed to be generally understood or end up in circular definitions. In order to overcome this deficiency and to give their work a sound mathematical justification, many researchers have meanwhile decided to apply concepts of multi-dimensional calculus, differential geometry, algebraic topology, and so on. Following this approach, the formal characterisation of topographic surfaces centres on the question which types of functions may be regarded as abstract models of real surfaces. From a geographical point of view, the importance of giving a satisfactory answer to the previous question is derived above all from the following four facts: Firstly, theoretical results obtained for functions describing topography also hold for functions describing such phenomena as population density, accessibility, pollution, temperature, precipitation, and so on; secondly, topographic surfaces represent the underlying continuous model of DTMs whereby *DTM* may stand as an abbreviation for *digital terrain model* or *discrete terrain model* respectively; thirdly, a great many of the results derived for mappings from $\mathbb{R}^2 \rightarrow \mathbb{R}$ are also true for real-valued mappings defined on curved surfaces – so-called *differentiable manifolds*[1]. This point, however, especially deserves our special attention "(because) geographical data (are) distributed over the curved surface of the earth, a fact which is often forgotten ... (and) we have few methods for analyzing data on the sphere or spheroid, and know little about how to model processes on its curved surface" (Goodchild, 1990, pp. 5). The final and perhaps the most important fact, however, why topographic surfaces should be characterised in a formal way is that a precise description clearly reveals those concepts that are commonly used in practice, but which are seldom or never explicitly stated.

As the formal characterisation of topographic surfaces requires some basic definitions and theorems from multi-dimensional calculus, we will proceed by summarizing the most important ones among them[2].

Definition 1 *Let $f(x, y)$ be a function whose partial derivatives f_{xx}, f_{xy}, f_{yx}, and f_{yy} exist. The matrix $Hf = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix}$ is termed the Hessian matrix of f. The Hessian matrix evaluated at a point (x_0, y_0) is defined by $\begin{pmatrix} f_{xx}(x_0, y_0) & f_{xy}(x_0, y_0) \\ f_{yx}(x_0, y_0) & f_{yy}(x_0, y_0) \end{pmatrix}$ and denoted by $Hf|_{(x_0, y_0)}$. The determinant $det(Hf)$ of the Hessian matrix Hf is called the Hessian determinant; when evaluated at the point (x_0, y_0), it is denoted by $det(Hf)|_{(x_0, y_0)}$.*

Definition 2 *A function $f(x, y)$ is termed k-fold continuously differentiable, or of class C^k, if the partial derivatives up to order k exist and are continuous. A smooth function is a function of class C^∞.*

[1] A differentiable manifold can be imagined as something looking like $\mathbb{R}^n$ but being smoothly curved. Examples of two-dimensional differentiable manifolds are the sphere or the torus in contrast to the cube or the cylinder.

[2] A discussion of other concepts being taken for granted within this chapter can be found in any standard book on calculus as, for example, in (Apostol, 1969).

In almost any geographic or cartographic application, *topographic surfaces* are regarded as functions $f(x, y)$, associating with each point (x, y) its respective altitude and being at least twice continuously differentiable, a view that is also widely employed when dealing with mappings describing socio-economic, physical, and other phenomena. Evidently, this conception is just an ideal one because, for example, overhanging rocks imply that there is no definite correspondence between certain points and their altitudes, or, for instance, break lines prevent $f(x, y)$ from being differentiable. In order to still apply the powerful tool of calculus, the original concept has to be modified by assuming that the continuously differentiable functions are not terrain as such but rather sufficiently close approximations of it. The remaining question, which seems to be deceptively simple in appearance but which leads rather deeply into abstract mathematics, is whether the theoretical requirements of differentiability and continuity of the derivatives suffice for functions to represent realizable topographic surfaces. It has already been demonstrated (Wolf, 1991b) that this need not always be true, because such mappings may nevertheless be endowed with a number of peculiarities that prevent the functions from being suitable models for representing the topography of a given area.

Another important concept closely related to topographic surfaces is *critical point theory*. Critical points[3] representing the peaks, pits, and passes of surfaces play a major role not only in cartography but also in a variety of other scientific applications where they represent either the extrema or the saddles of functions to be maximised or minimised. The importance of the critical points results from the fact that they contain significantly more information than any other point on the surface, because they provide information not only about a specific location but also about its surroundings (Peucker, 1973, Pfaltz, 1976, 1978, Peucker et al., 1976, 1978). As a consequence, their application not only facilitates the characterisation and visual analysis of the topography of a given area but also results in considerable savings in data capture and data management when they are employed within digital terrain models. Before stating two theorems that allow the classification of the critical points, their formal description will be given.

Definition 3 *A point* (x_0, y_0) *is a relative (local) maximum of* $f(x, y)$ *if and only if* $f(x, y) < f(x_0, y_0)$ *for all* $(x, y) \in U_\varepsilon(x_0, y_0)$.
A point (x_0, y_0) *is a relative (local) minimum of* $f(x, y)$ *if and only if* $f(x, y) > f(x_0, y_0)$ *for all* $(x, y) \in U_\varepsilon(x_0, y_0)$.
A point (x_0, y_0) *is a saddle of* $f(x, y)$ *if and only if* $f(x, y)$ *has a local maximum along one line leading through* (x_0, y_0) *and a local minimum along another line leading through* (x_0, y_0).

According to the above definition, saddle points are only those points with exactly two ridges (lines connecting passes with peaks) and exactly two courses (lines connecting passes with pits) emanating from them, thus excluding monkey saddles or the like. The following theorem enables the computation as well as the classification of the

[3] In accordance with Peucker (1973), who has introduced these ideas into geography and computer cartography, the terms *critical points* and *surface-specific points* will be alternately used.

critical points of a function $f(x, y)$ by applying the concepts of the partial derivatives and the Hessian determinant of $f(x, y)$.

Theorem 1 *(x_0, y_0) is a local maximum of a function $f(x, y)$, which is twice continuously differentiable in $\mathbb{R}^2$, if and only if $f_x(x_0, y_0) = f_y(x_0, y_0) = 0$, $det(Hf)|_{(x_0,y_0)} > 0$, and $f_{xx}(x_0, y_0) < 0$ (or equivalently $f_{yy}(x_0, y_0) < 0$).*
(x_0, y_0) is a local minimum of a function $f(x, y)$, which is twice continuously differentiable in $\mathbb{R}^2$, if and only if $f_x(x_0, y_0) = f_y(x_0, y_0) = 0$, $det(Hf)|_{(x_0,y_0)} > 0$, and $f_{xx}(x_0, y_0) > 0$ (or equivalently $f_{yy}(x_0, y_0) > 0$).
(x_0, y_0) is a saddle point of a function $f(x, y)$, which is twice continuously differentiable in $\mathbb{R}^2$, if and only if $f_x(x_0, y_0) = f_y(x_0, y_0) = 0$ and $det(Hf)|_{(x_0,y_0)} < 0$.
(x_0, y_0) is a non-degenerate critical point of a function $f(x, y)$, which is twice continuously differentiable in $\mathbb{R}^2$, if and only if $f_x(x_0, y_0) = f_y(x_0, y_0) = 0$ and $det(Hf)|_{(x_0,y_0)} \neq 0$.

An equivalent characterisation of the critical points of a function $f(x, y)$ can be given by examining the eigenvalues of the corresponding Hessian matrix (Nackman, 1982, 1984).

Theorem 2 *Let $f(x, y)$ be twice continuously differentiable in $\mathbb{R}^2$ and $(x_0, y_0) \in \mathbb{R}^2$. Further, let $f_x(x_0, y_0) = f_y(x_0, y_0) = 0$ and the determinant of the Hessian matrix Hf evaluated at (x_0, y_0) be unequal to zero. Then there is*
a local maximum at (x_0, y_0) if the number of negative eigenvalues of $Hf|_{(x_0,y_0)}$ is two,
a saddle at (x_0, y_0) if the number of negative eigenvalues of $Hf|_{(x_0,y_0)}$ is one,
and a local minimum at (x_0, y_0) if the number of negative eigenvalues of $Hf|_{(x_0,y_0)}$ is zero.

The number of negative eigenvalues of $Hf|_{(x_0,y_0)}$ is also termed the *index* of (x_0, y_0); thus, a local maximum is a critical point of index two, a saddle is a critical point of index one, and a local minimum is a critical point of index zero. The so-defined index of a critical point may also be interpreted as an "index of instability (because) a ball displaced slightly from a relative minima will "roll back" to that minima. It is a point of stable equilibrium; ... A ball displaced from a saddle point may or may not return to that point of equilibrium, depending on the direction of displacement; while a ball displaced from a relative maxima is completely unstable" (Pfaltz, 1976, p. 79, Pfaltz, 1978).

After having characterised the critical points of a twice continuously differentiable function $f(x, y)$ in a formal way, we will direct our attention to those functions whose surface-specific points are, without any exception, non-degenerate. Since practice has shown that degenerate critical points are extremely unlikely to occur in real-world applications, functions possessing exclusively non-degenerate critical points have been studied extensively by numerous authors.

Definition 4 *A smooth function is termed a Morse function if all of its critical points are non-degenerate.*

For Morse functions, the following theorems (whose importance will become obvious in the next two sections) hold.

Theorem 3 *Each Morse function on a compact manifold has only a finite number of critical points; in particular, all of them are distinct.*

Theorem 4 *The critical points of a Morse function are always isolated*[4].

Theorem 5 *Let f be a Morse function, which is defined on a simply connected domain bounded by a closed contour line, then the number of minima of f minus the number of saddles of f plus the number of maxima of f equals two.*

The concept of Morse functions – though not explicitly mentioned – has been used in almost any practical application, because these mappings represent the prototype of functions eligible to characterise topographic surfaces. One exception worth mentioning, however, is represented by the work of Pfaltz (1976, 1978), who has been the first to explicitly apply results of Morse theory (which is the study of the relationships between a function's critical points and the topology of its domain) in order to characterise the global topological structure of topographic surfaces in a formal way. Contrary to CAD applications, the domain of primary interest in almost any geographic and cartographic application is either a plane or the surface of a sphere, with the latter seeming to be more appropriate and leading to somewhat cleaner results. As a consequence, the remainder of this chapter will be confined to Morse functions defined over a bounded region of the plane with all points outside the boundary identified as a single pit or peak respectively, thus mapping the bounded region onto a sphere.

2.3 DATA STRUCTURES FOR THE TOPOLOGICAL CHARACTERISATION OF TOPOGRAPHIC SURFACES

It has already been pointed out that the accumulation of information (which has been enabled by recent trends in computer hardware) calls for sophisticated data structures in terms of spatial resolution and topology. Nevertheless, contemporary CAD and GIS systems still handle smooth object shapes by extending conventional polyhedral representation. This approach, however, leads to the polyhedral decomposition of smooth object shapes and thus has no relationship to the geometrical features of the smooth surfaces; in fact, it rather causes a variety of different problems. In order to solve these problems, it is essential to develop appropriate models that are based on shape features intrinsic to the smoothness of the surfaces. Since the detailed description of even the most important data structures currently employed within CAD and GIS systems is far beyond the scope of this chapter, we will confine ourselves to the presentation of four graph–theoretic approaches that provide us with a means of handling the critical points in an abstract way and that are suitable for describing the global topology of a surface.

The first of the four feature-based modelling methods for smooth surfaces to be discussed within this chapter are Reeb graphs (Shinagawa et al., 1991, Takahashi et al., 1995, Fomenko and Kunii, 1997, Takahashi et al., 1997). Although primarily used

[4] A critical point is called *isolated* if there is no other critical point sufficiently close to it.

within CAD systems to design the topological skeleton of an object shape, some geographic applications of this data structure (which represents the splitting and merging of contour lines of equal height) do still exist. Formally, a Reeb graph, whose definition is based on the fact that when moving from the uppermost contour of a surface to the lowermost one, a new contour appears at a peak, whereas an existing contour disappears at a pit, and, in addition, a contour splits or two contours merge at a pass, is characterised as follows.

Definition 5 *Let $f : M \to \mathbb{R}$ be a real-valued function on a compact manifold M. The Reeb graph of f is the quotient space of M defined by the equivalence relation $\sim$ in the following way: $x_1 \sim x_2$ holds if and only if $f(x_1) = f(x_2)$ and x_1 and x_2 are in the same connected component of $f^{-1}(f(x_1))$.*

According to the above definition, the Reeb graph is obtained from the respective manifold as topological quotient space where all the points having the same value under the Morse function and lying in the same connected component as the corresponding cross section are identified. Thus, the connected components of the part of the manifold situated strictly between two critical levels are represented by separate line segments, namely the edges of the graph, whereas each critical point corresponds to a vertex of the graph (see Chapter 6 for more information on Reeb graphs).

Departing from the fact that researchers in the fields of computer graphics and GIS had already extensively studied methods for extracting terrain features from discrete elevation data while their techniques did not guarantee the topological integrity of the extracted features, however, Takahashi et al. (1995) developed a number of algorithms to extract the critical points of a topographic surface and to construct two types of graphs, with one of them being the Reeb graph, for characterising the global topological structure of a topographic surface (see Chapter 3 for more details). The authors also demonstrated that their algorithms maintained topological invariants of smooth surfaces and they illustrated the efficiency of their approach by using data from a region situated near Lake Ashinoko in Japan.

Other powerful structures for visualising the topological behaviour of topographic surfaces are contour trees and surface trees respectively. These concepts, which were originally developed in the early sixties and re-invented by Morse (1968, 1969) and Mark (1977), represent special cases of the more general Reeb graph, as they are also based upon adjacency relationships between contour lines.

Contour trees and surface trees differ in this respect that Morse's formal approach rests upon particular but arbitrarily chosen, closed contour lines (e.g. all lines with a contour interval of 100 m), which represent the vertices of the so-called *graph of the contour map*. Two of these contour loops are termed adjacent, with the corresponding nodes being connected by an edge in the contour graph if no other selected contour loop lies between them. It can easily be proved that because of this construction rule, cycles may never occur, thus implying that these graphs will always be trees. Evidently, the termination points of these *contour trees* always represent areas containing pits or peaks, whereas the branching points always represent areas containing passes. In addition, the contour tree may contain vertices of degree two that are induced by the contour lines; these nodes, which generally represent areas without any critical point,

carry information about absolute altitudes and relative altitudes. From the contour tree one obtains the *surface tree* by decreasing the contour interval to zero and by removing all vertices of degree two by homomorphic contractions. Whereas the vertex set of the surface tree is identical with the set of the critical points, its edge set represents only a subset of the set of all critical lines of the corresponding topographic surface.

The widespread use of contour maps in a variety of different disciplines has led to an increased interest in contour trees as an efficient data structure for representing spatial relationships between contour lines. As in almost any publication dealing with contour trees a different algorithm for their construction is described, a great body of today's research work is concerned with the improvement of existing algorithms with regard to their run-time behaviour (van Kreveld et al., 1997, Carr et al., 2003; also see Chapter 5). In addition to this primarily theoretical work, numerous practical applications of contour trees are also reported. So, for instance, Mark (1977) used them for the topological analysis of geomorphic surfaces, Roubal and Poiker (1985) employed contour trees when developing a partially automated system for the extraction of contour lines, Kweon (1991) and Kweon and Kanade (1991, 1994) applied contour trees for extracting topographic terrain features like pits, peaks, passes, ridges, and ravines from contour maps, and Bajaj et al. (1997) employed them within a user interface component as a tool for assisting the user when analysing complex scalar fields interactively.

The next feature-based data structure for the characterisation of topographic surfaces to be discussed within the present chapter is the TIN data structure whereby *TIN* is the abbreviation for *triangulated irregular network*. This structure was introduced by Peucker (1973) and Peucker et al. (1976, 1978) as an alternative to existing digital terrain models based on regular grids, with the following two considerations motivating their work: firstly, the fact that with the changing roughness of terrain from one landform to another, a regular grid must be adjusted to the roughest terrain, thus containing a lot of highly redundant information in smooth terrain; and secondly, the observation that different uses of terrain models demand different representations, whereby these representations should be suited to the phenomena under study and not be imposed by a sampling rule. Contrary to grids, TINs allow the density of nodes to be varied laterally and be adapted to local detail with the result that the smoothly changing part of a surface can be adequately represented by relatively few nodes, whereas high-frequency undulations can be recorded by more frequent sampling.

Within the TIN data structure, as introduced by Peucker, a surface is modelled as a set of contiguous non-overlapping triangles whose vertices are located on the respective topographic surface. Evidently, the tiling of the surface in these irregularly shaped triangle facets can be represented by a planar graph. In addition, the topological structure of the surface can be taken into account by regarding the surface-specific points and the surface-specific lines; within the model under discussion, these points and lines form a subgraph of the graph describing the overall TIN data structure.

The ability of TINs to give suitable approximations of arbitrary topographic surfaces by using only a restricted amount of data (a feature that is due to their irregular structure and the application of the surface-specific points and the surface-specific lines) induces that TINs are nowadays the most prominent data structure for terrain representation and are widely employed in commercial packages. Today there exists a

great amount of literature on TINs, concerned with theoretical developments as well as interesting practical applications. However, as it is far beyond the scope of the present chapter to discuss even the most important of these approaches, we will turn our interest to another feature-based modelling method for smooth surfaces, namely surface networks.

2.4 SURFACE NETWORKS

This data structure has been introduced by Pfaltz (1976, 1978) with his approach being conducted by Mark (1977), Wolf (1988a,b, 1989, 1990, 1991a), Rana (2000b), Rana and Wood (2000), and Rana and Morley (2002). Surface networks represent special types of graphs with the vertex sets consisting of the critical points and the edge sets consisting of the critical lines of a Morse function defined over a domain that is simply connected and bounded by a closed contour line. The so-defined model-structure can be further enhanced by associating real numbers greater than zero with the edges and the nodes to indicate their importance for both the macro-structure and the micro-structure of the underlying surface. The graphs thus obtained – termed *weighted surface networks*[5] – represent a powerful tool for both characterising and generalising topographic surfaces. After their formal definition has been given, it will be shown that the generalisation process can be characterised by two graph–theoretic contractions that reduce the number of the graph's edges and vertices but preserve its topological structure, and, consequently, that of the underlying surface.

Figure 2.1 illustrates a surface containing pits, passes, and peaks together with the corresponding ridges and courses. P_0 denotes the set of all pits, P_1 the set of all passes, and P_2 the set of all peaks, whereas x_i $(i = 1, \ldots, m_1)$ specifies an individual pit, y_j $(j = 1, \ldots, m_2)$ specifies an individual pass, and z_k $(k = 1, \ldots, m_3)$ specifies an individual peak. As can be seen, in Figure 2.1 $m_1 = 1$ (surrounding pit), $m_2 = 5$, and $m_3 = 6$.

The substantial phenomena of any topographic surface now can be portrayed easily by an edge-weighted directed graph[6] with the vertices representing the pits, passes, and peaks, the edges depicting the courses and ridges, and the edge weights specifying differences in elevations (see Figure 2.2). It is easy to demonstrate, however, that not all graphs whose vertex sets are composed of the critical points and whose edge sets are composed of the critical lines can be regarded as abstract models of real topographic surfaces; they rather have to satisfy the properties listed below.

[5] Throughout the remainder of this chapter, the terms *surface network* and *weighted surface network* will be used synonymously.

[6] The basic graph–theoretic definitions can be found in (Bondy and Murty, 1978). Additionally, the following concepts from (Pfaltz, 1976) will be used in this chapter: A *circuit* is a closed walk. A graph is *connected*, if given any node u, one can reach all other nodes v by a walk that follows a sequence of edges, though not necessarily in the indicated direction. A graph is said to be *tripartite* if the vertex set can be partitioned into three subsets V_0, V_1, and V_2, so that every edge is incident with one node of V_{i-1} and one node of V_i for $1 \leq i \leq 2$. The *valency* $val(u, v)$ denotes the number of edges between vertex u and vertex v. $L(v) = \{u|(u, v) \in E\}$ denotes the set of all adjacent nodes of the vertex v lying to the "left" of it; $R(u) = \{v|(u, v) \in E\}$ specifies the set of all adjacent nodes of the vertex u lying to the "right" of it; $L(u)$ and $R(v)$ reflect a "left-to-right" partial ordering of the graph.

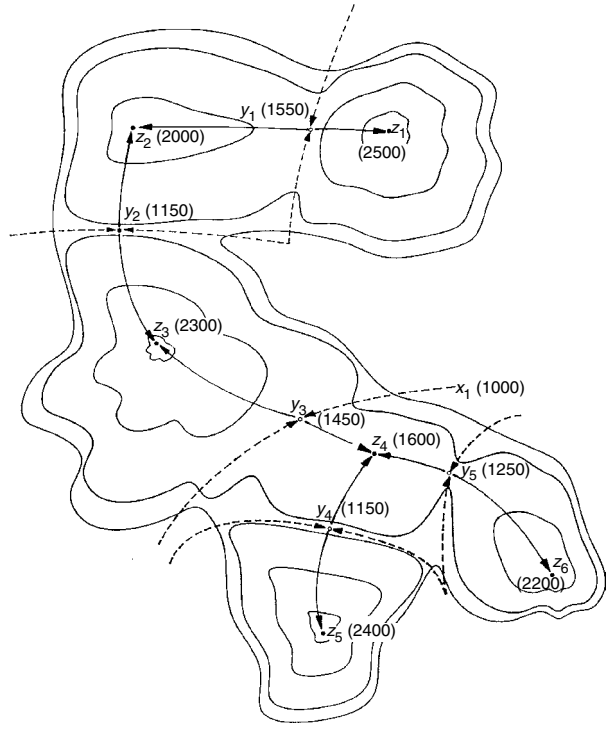

Figure 2.1 *Topographic surface with its critical points and critical lines; numbers in parentheses indicate altitudes of data points*

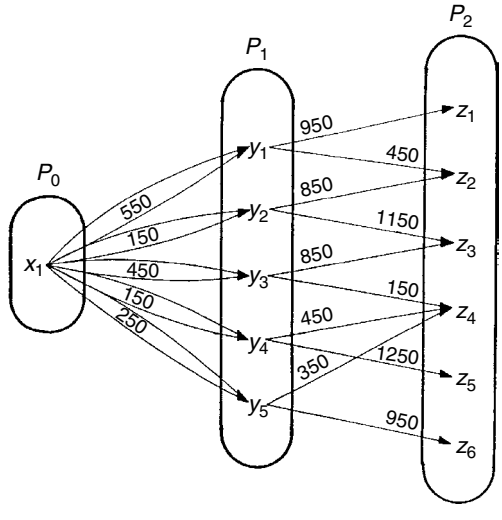

Figure 2.2 *Graph representing the topological structure of the surface illustrated in Figure 2.1. Edge weights, which are defined as $h(y_j) - h(x_i)$ and $h(z_k) - h(y_j)$ respectively, indicate differences in altitudes (h denotes the height of a specific data point). Edges with valency two are dotted twice*

Definition 6 *A weighted, directed, tripartite graph* $W = (P_0, P_1, P_2; E)$ *is called a weighted surface network (Pfaltz graph) if*

P0. *W is planar*
P1. *the subgraphs* $[P_0, P_1]$ *and* $[P_1, P_2]$ *are connected*
P2. $|P_0| - |P_1| + |P_2| = 2$
P3. *for all* $y \in P_1$, $id(y) = od(y) = 2$
P4. $val(x, y_i) = val(y_i, z) = 1$ *implies that there exists* $y_j \neq y_i$ *so that* (x, y_j), $(y_j, z) \in E$
P5a. (x, y) *is an edge of a circuit in the bipartite graph* $[P_0, P_1]$ *if and only if* $val(y, z) \neq 2$ *for all* $z \in P_2$
P5b. (y, z) *is an edge of a circuit in the bipartite graph* $[P_1, P_2]$ *if and only if* $val(x, y) \neq 2$ *for all* $x \in P_0$
P6. $w(e_i) > 0$ *for all* $e_i \in E$
P7. *for all* $x \in P_0$, $y_i, y_j \in P_1$, $z \in P_2$ *and* (x, y_i), (x, y_j), (y_i, z), $(y_j, z) \in E$ *holds* $w(x, y_i) + w(y_i, z) = w(x, y_j) + w(y_j, z)$
P8a. *if* $val(x, y) = 2$ *with* $e_{i_1} = (x, y)$ *and* $e_{i_2} = (x, y)$ *then* $w(e_{i_1}) = w(e_{i_2})$
P8b. *if* $val(y, z) = 2$ *with* $e_{i_1} = (y, z)$ *and* $e_{i_2} = (y, z)$ *then* $w(e_{i_1}) = w(e_{i_2})$.

Planarity, which is required by P0, is one of the nine properties that any surface network must exhibit, because an intersection of its edges would be equivalent to the intersection of the ridges and courses of the topographic surface, thus implying the impossibility of its realisation. P1 ensures that all pits and saddles are connected by course lines, and all passes and peaks are connected by ridge lines. P2 and P3 are direct consequences of the assumption that the respective surface is approximated by a Morse function. P4 guarantees that if there exists a path from pit x via pass y_i to peak z, which consists only of edges with valency one, then there exists another path from pit x to peak z via a distinct saddle y_j. P5a and P5b exclude special configurations that are non-realisable (Pfaltz, 1976, Wolf, 1988a,b, 1989, 1990, 1991a). P6 says that all edge weights must be greater than zero and thus have to be defined as $h(y_j) - h(x_i)$ and $h(z_k) - h(y_j)$ respectively, with h denoting the height of a specific data point. P7 ensures that for all paths from pit x to peak z the difference in elevation is the same, no matter which saddle point is passed. P8a guarantees that all course lines from a pit to a pass will have the same difference in altitude. P8b states the analogy for ridges.

The cartographic importance of Pfaltz graphs, however, does not only result from their applicability for describing the topological structure of topographic surfaces but rather also from the fact that these graphs can be condensed by two contractions that reduce the number of edges and vertices, but preserve the topological structure of the corresponding topographic surface. These two contractions will be discussed next.

Definition 7 *Let W be a weighted surface network and let* y^o *be a saddle point with* $R(y^o) = \{z^o, \overline{z}\}$ *and* $w(y^o, z^o) \leq w(\overline{y}_i, z^o)$ *for* $i = 1, 2, \ldots, n-1$. *Let* z^o *be a peak of degree n with* $L(z^o) = \{y^o, \overline{y}_1, \overline{y}_2, \ldots, \overline{y}_{n-1}\}$. *The* (y^o, z^o)*-w-contracted graph* W' *is the graph with vertex set* $V(W') = V' = V - \{y^o, z^o\}$, *edge set* $E(W') = E' = E + \{(\overline{y}'_1, \overline{z}'), (\overline{y}'_2, \overline{z}'), \ldots, (\overline{y}'_{n-1}, \overline{z}')\}$, *and edge weights:*

(a) $w(\overline{y}_i', \overline{z}') = w(\overline{y}_i, z^o) - w(y^o, z^o) + w(y^o, \overline{z})$ *for* $i = 1, 2, \ldots, n-1$
(b) $w(e') = w(e)$ *for all other edges* $e' \in E(W')$.

The whole set of operations taking the original surface network onto the condensed one is called (y^o, z^o)*-w-contraction with the "w" indicating that a weighted surface network is going to be contracted.*

The so-defined (y^o, z^o)-w-contraction removes the peak z^o and its highest adjacent saddle y^o together with all surface-specific lines incident with at least one of these critical points. This elimination, however, causes the loss of two fundamental features of the surface network, because (i) the condensed subgraph $[P_1', P_2']$ is no longer connected (violation of P1) and (ii) there exist passes $\overline{y}_i'$ [7] with $od(\overline{y}_i') = 1$ (violation of P3). In order to ensure that W' is a weighted surface network too, its edge set $E(W')$ must comprise the "old" set $E(W)$ as well as "new" links connecting $\overline{y}_i'$ with $\overline{z}'$. Since the inclusion of these edges into $E(W')$ can be regarded as a substitution of the paths $\langle[\overline{y}_i, z^o], [z^o, y^o], [y^o, \overline{z}]\rangle$ by $(\overline{y}_i', \overline{z}')$ for $i = 1, 2, \ldots, n-1$, it is reasonable to assign the values $w(\overline{y}_i, z^o) - w(y^o, z^o) + w(y^o, \overline{z})$ to the new links $(\overline{y}_i', \overline{z}')$. These weights can be justified cartographically, as they represent nothing else but the differences in elevations of paths starting at saddle $\overline{y}_i$, leading up to peak z^o, leading down to pass y^o, and finally ending in $\overline{z}$. Moreover, the selection of y^o guarantees that all weights are greater than zero, which is a prerequisite for the realisation of the corresponding topographic surface.

It has been proved in a formal way (Wolf, 1988b) that a (y^o, z^o)-w-contraction takes a weighted surface network onto another one and thus may be regarded as an elementary step of a cartographic generalisation process. Since a similar contraction for pits – a so-called (x^o, y^o)-*w-contraction* – can also be defined, any surface network can be condensed by repeated applications of (x^o, y^o)-w-contractions and (y^o, z^o)-w-contractions respectively, until a so-called *elementary surface network* is obtained (Wolf, 1988a,b, 1989, 1991a). Evidently, these surface networks whose vertex sets consist either of one pit, one pass, and two peaks, or of two pits, one pass, and one peak can always be obtained by a series of $|P_1| - 1$ (x^o, y^o)-w-contractions and (y^o, z^o)-w-contractions respectively, with $|P_1|$ denoting the number of saddle points of the given Pfaltz graph. Elementary surface networks, though theoretically interesting as they indicate the end of any generalisation process, are, however, "oversimplified" for any practical application. In order to overcome this deficiency, different criteria can be employed to terminate the contraction process at an earlier stage, thereby creating a surface network with a specified degree of simplicity.

2.5 SURFACE NETWORKS AND CARTOGRAPHIC GENERALISATION

A further improvement of the model developed so far can be achieved by associating weights also with the vertices of the surface network in order to indicate their importance for the macro-structure and the micro-structure of the corresponding topographic surface. These weights termed *importance I* of the critical points cannot be defined

[7] According to the above definition, $\overline{y}_i$ are the vertices incident with z^o but different from the node y^o, which is removed by the contraction.

absolutely, however, but must always take into consideration the individual character of the problem under discussion as well as the given topographic facts. Consequently, there exist several ways for the calculation of I that are all based on differences in altitudes between adjacent surface-specific points. A thorough discussion concerning the pros and cons of different definitions of I, such as the maximum, the minimum, or the sum of differences in elevation between a pit (peak) and all of its adjacent passes, and of other user-defined contractions can be found in (Wolf, 1988b, Rana, 2000b, Rana and Wood, 2000, Rana and Morley, 2002). As can be seen from Figure 2.2, defining I as the maximum of differences in altitude between a peak (pit) and all of its adjacent saddles results in the following arrangement with the corresponding values of I given in parentheses: z_4 (450), x_1 (550), z_2 (850), z_1 (950), z_6 (950), z_3 (1150), and z_5 (1250).

After having defined the importance I of the surface-specific points in a formal way, it is possible to specify an algorithm for the generalisation of topographic surfaces, which is based on the previously described (x^o, y^o)-w-contractions and (y^o, z^o)-w-contractions, and the weights associated with the vertices of the Pfaltz graph corresponding to the respective surface. The procedure, whose main advantage is to incorporate systematically the importance of the critical points into the generalisation process, thus avoiding their lexical elimination, runs as follows:

Algorithm for the generalisation of topographic surfaces

(0) Specify the desired degree of simplicity.
(1) Calculate the importance I of pits x_i and peaks z_k and arrange them in ascending order.
(2) Select the pit x^o or peak z^o which lies within the boundary contour and whose importance is minimal.
(3) Apply an appropriate (x^o, y^o)-w-contraction or a (y^o, z^o)-w-contraction respectively.
(4) If the specified degree of simplicity is achieved, stop. Otherwise go to (1).

Evidently, the application of the above algorithm in combination with the definition of I as the maximum of differences in elevation between a peak (pit) and all of its adjacent saddles induces that, first of all, a (y^o, z^o)-w-contraction with $y^o = y_3$ and $z^o = z_4$ will be applied, thus taking the surface network of Figure 2.2 onto the one depicted in Figure 2.3, with the corresponding ridge lines and course lines being illustrated in Figure 2.4.

It is evident that, because of their definition, weighted surface networks are a purely topological data structure. However, for a number of applications like the generalisation of topographic surfaces or the characterisation of certain topographic features – such as river junctions, bifurcations of ridges, and crests leading from peaks to river junctions – the additional consideration of geometrical features is indispensable (Wolf, 1988a,b, 1989, 1990, 1991a). This consideration of geometrical properties can be easily achieved by embedding the weighted surface network into a metric space, that is, by associating (x, y) coordinates with the critical points and the critical lines. The resulting surface networks termed *metric surface networks* represent the starting point of almost any practical application.

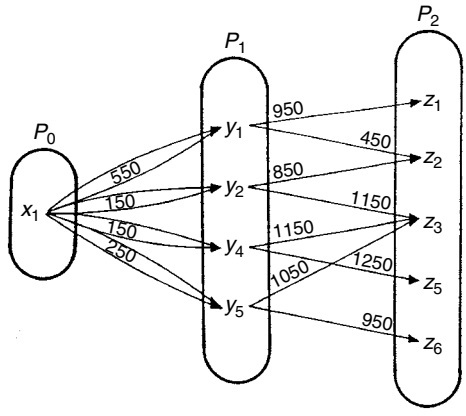

Figure 2.3 *Weighted surface network of Figure 2.2 after a* (y^o, z^o)*-w-contraction with* $y^o = y_3$ *and* $z^o = z_4$

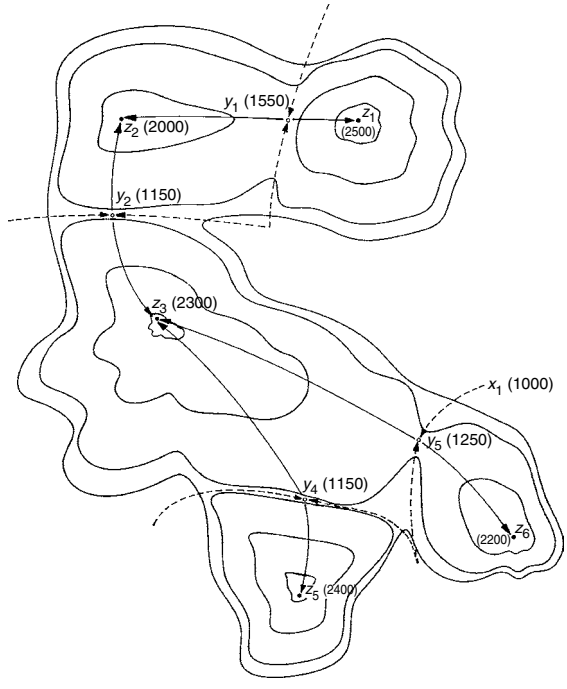

Figure 2.4 *Ridge lines and course lines associated with the Pfaltz graph illustrated in Figure 2.3*

The previously described algorithm has already been used to generalise a real topographic surface located in the Latschur Mountains in the southern part of Austria (Wolf, 1988a,b, 1989, Rana, 2000b, Rana and Wood, 2000, Rana and Morley, 2002). Figure 2.5 depicts the Pfaltz graph corresponding to the given topographic surface. Figure 2.6 illustrates the condensed surface network after thirty (x^o, y^o)-w-contractions and (y^o, z^o)-w-contractions respectively.

Figure 2.5 *Pfaltz graph corresponding to a topographic surface situated in the Latschur Mountains, Austria. Legend: circles – pits, triangles – peaks, squares – passes*

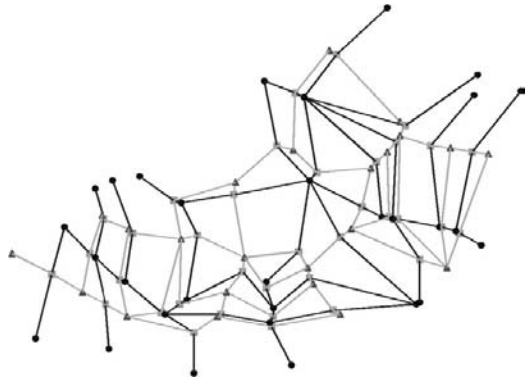

Figure 2.6 *Surface network of Figure 2.5 after thirty* (x^o, y^o)*-w-contractions and* (y^o, z^o)*-w-contractions respectively. Legend: circles – pits, triangles – peaks, squares – passes*

2.6 FUTURE DIRECTIONS

Although a great deal of theoretical as well as practical work still remains to be done, Pfaltz graphs seem to be a promising data structure for both the characterisation of the macro-structure and the micro-structure of a topographic surface as well as for their generalisation. Theoretical problems still to be solved include, for instance, the formal analysis of the connection between surface networks and other feature-based data structures – such as Reeb graphs, contour trees, and so on (Mark, 1977, Wolf, 1993, Takahashi et al., 1995) – or the investigation of the relationships between Pfaltz graphs and hydrologic concepts, such as Werner's interlocking ridge

and channel networks (Werner, 1988, Wolf, 1992). The practical problems to be tackled are, for example, the automated generation of surface networks from digital elevation models (Wood, 1996a, 1998), the development of improved algorithms combining topological and geometric features, or the inclusion of the developed procedures into software packages handling spatial data structures.

3

Algorithms for Extracting Surface Topology from Digital Elevation Models

Shigeo Takahashi

3.1 INTRODUCTION

Contemporary *geographical information systems (GISs)* generally maintain terrain surfaces in the form of *digital elevation models (DEMs)* and visualise them by extracting their topographical features as landmarks for understanding the surface shapes. Such features include critical points such as *peaks*, *passes*, and *pits*, and feature lines such as *ridges* and *course lines* traversing between them.

Since these features come from the theory of differential topology, they can enjoy some topological formulas. The most important formula is the *Euler–Poincaré formula*. As mentioned in Chapter 2, a terrain surface is supposed to be a bounded region of a sphere where all the points outside can be identified with the virtual pit, which is the bottom pit of the sphere. By taking into account this virtual pit, the extracted critical points must satisfy the Euler–Poincaré formula: $\#\{\text{peak}\} - \#\{\text{pass}\} + \#\{\text{pit}\} = 2$, where #{peak}, #{pass}, and #{pit} represent the numbers of peaks, passes, and pits, respectively[1]. However, conventional methods do not maintain the Euler–Poincaré formula that proves the correctness of the extracted critical points. Figure 3.1 shows such an example in which the conventional eight-neighbour method (Peucker and Douglas,

[1] By excluding the virtual pit, this formula becomes $\#\{\text{peak}\} - \#\{\text{pass}\} + \#\{\text{pit}\} = 1$. This is called the *mountaineers' equation* (Griffiths, 1981)

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

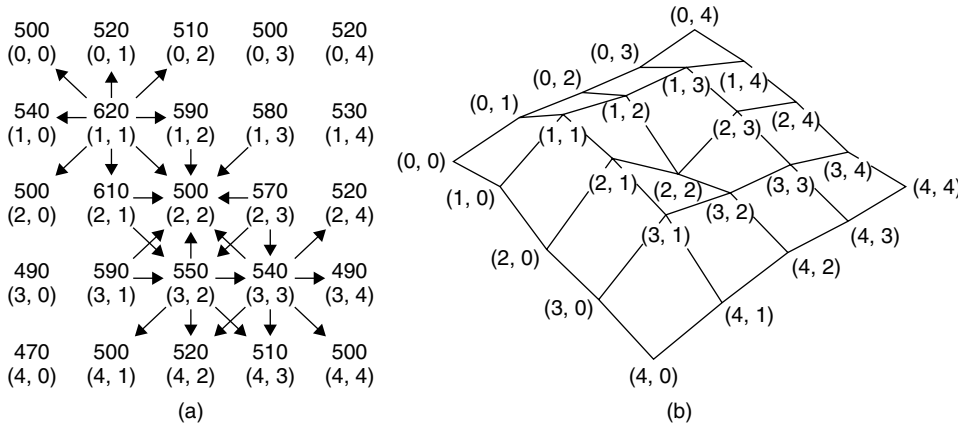

Figure 3.1 *An example of (a) a DEM and (b) its projection: the eight-neighbour method fails to extract correct critical points from this DEM*

1975) fails to extract correct critical points. As described earlier, the extracted critical points must satisfy #{peak} − #{pass} + #{pit} = 2 in this case. Nonetheless, the eight-neighbour method extracts the point (1, 1) as a peak, (3, 2) and (3, 3) as passes, and (2, 2) as a pit, and thus violates the Euler–Poincaré formula.

The correctness of the critical point extraction is vital for further analysis of terrain *surface topology* when we try to capture the global configurations of the critical points and feature lines. Such a global configuration is efficiently captured by a *critical point graph* that represents critical points as its vertices and relationships between them as its edges. While several critical point graphs (Reeb, 1946, Pfaltz, 1976, Nackman, 1984) including *surface networks* (Pfaltz, 1976) (see Chapter 2) were formulated mathematically until the mid-1980s, their practical implementation has been established quite recently. This is because the input data is different from an ideal smooth surface in that it usually involves unexpected noise and degeneracy arising from discrete sampling and quantization.

The main challenge of this chapter is to provide mathematical fundamentals for identifying such critical points and feature lines, and noble implementation of the associated algorithm (Takahashi et al., 1995). Here, the present algorithm is valid in the sense that the extracted features definitely satisfy the Euler–Poincaré formula derived from the studies in differential topology, and also robust in the sense that it can extract correct features even when the input data involves unexpected noise and degeneracy. The network of the ridge and course lines offers an excellent partition of the input terrain surface, which is effectively captured by a critical point graph called the *surface network* (Pfaltz, 1976). This chapter also presents an algorithm that converts the surface network into a *level-set graph*[2] called the *Reeb graph* (Reeb, 1946), which is also a critical point graph that represents the splitting and merging of cross-sectional contours with respect to the height value. Several examples and potential applications are also included to demonstrate the feasibility of the present framework.

[2] Here, the level set means a set of points of constant value for a given scalar function.

The organisation of this chapter is as follows: Section 3.2 describes the mathematical definition of critical points and an algorithm for extracting them from the input DEM correctly. Section 3.3 formulates the ridge and course lines and the surface network, and then explains how to extract these features from the given DEM. Section 3.4 provides an excellent algorithm that converts the surface network to the Reeb graph by taking into account the mathematical properties of smooth surfaces only. After presenting several examples and potential applications in Section 3.5, Section 3.6 concludes this chapter with references to related work.

3.2 EXTRACTING CRITICAL POINTS

We begin with the definition of a DEM, which is described as follows. The *DEM* is a set of sample points $\{(x_i, y_i, f(x_i, y_i))|i = 1, 2, \ldots\}$ on a single-valued function

$$z = f(x, y) \tag{3.1}$$

where z is the height in the Cartesian coordinate system spanned by the x-, y-, and z-axes. This definition actually helps us extract topographical features and their associated graph representations systematically.

3.2.1 Critical points

Tracking cross-section contours of the terrain surface according to the height value will produce their topological changes such as appearance, merging, splitting, and disappearance. A *critical point* is defined to be a point where such a topological transition in cross-sectional contours takes place.

More mathematically, a critical point of a height function (Equation (3.1)) is defined to be a point that satisfies

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0 \tag{3.2}$$

The *Morse lemma* (Milnor, 1963) claims that an infinitesimal neighbourhood around a critical point of equation (3.1) has a local coordinate system where f has one of the following quadratic forms:

$$f = \begin{cases} -x^2 - y^2 & \text{peak} \quad (\text{index } 2) \\ -x^2 + y^2 & \text{pass} \quad (\text{index } 1) \\ x^2 + y^2 & \text{pit} \quad (\text{index } 0) \end{cases} \tag{3.3}$$

Here, the index means the number of negative eigenvalues of the Hessian matrix at the critical point, that is,

$$Hf = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial x \partial y} & \dfrac{\partial^2 f}{\partial y^2} \end{pmatrix} \tag{3.4}$$

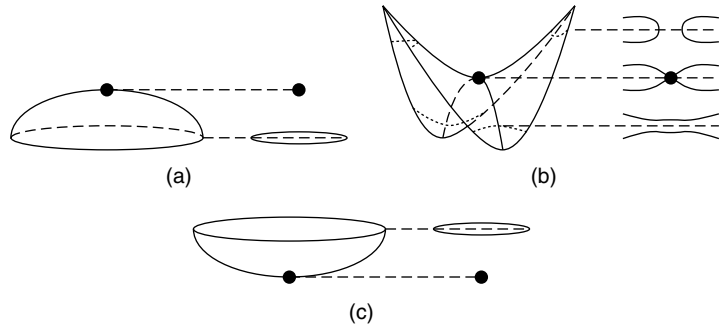

Figure 3.2 *Critical points: (a) a peak, (b) a pass, and (c) a pit*

As shown in equation (3.3), there are three types of critical points for the surface case: a *peak* (index 2), a *pass* (index 1), and a *pit* (index 0).

Figure 3.2 depicts a topological transition at each type of critical point. At a peak, a new contour appears (Figure 3.2(a)) when lowering the height value while an existing contour disappears at a pit (Figure 3.2(c)). This implies that a peak is higher than all other points in its neighbourhood, while a pit is lower. On the other hand, a pass splits an existing contour into two, or merges two contours into one (Figure 3.2(b)). A critical point usually allows one of the above topological transitions in its corresponding cross-sectional contours. In this case, a critical point is said to be *non-degenerate* and the corresponding Hessian matrix (Equation (3.4)) has full rank. Otherwise, a critical point is *degenerate*. One simple example of the degenerate critical point is a point at which three or more contours merge into one, or one contour splits into three or more simultaneously[3].

If all the critical points are non-degenerate, they must satisfy the aforementioned *Euler–Poincaré* formula:

$$\#\{\text{peak}\} - \#\{\text{pass}\} + \#\{\text{pit}\} = 2 \tag{3.5}$$

Note that the formula (3.5) holds if the given surface is topologically equivalent to a sphere. In our framework, as described in Chapter 2, we put the given DEM on the top of a sphere and take into account the bottom pit of the sphere when applying the Euler–Poincaré formula. This is shown in Figure 3.3, where we call the bottom pit of the sphere a *virtual pit*. Considering the DEM together with the virtual pit, we can preserve the integrity of the extracted critical points by applying the Euler–Poincaré formula (3.5).

3.2.2 Algorithm for extracting critical points

Recall that the conventional eight-neighbour method cannot extract critical points that satisfy the Euler–Poincaré formula (3.5). This is because the DEM is a set of discrete samples and lacks smooth interpolation over the domain of *xy*-coordinates. To ensure that the extracted critical points satisfy the Euler–Poincaré formula, we have

[3] A monkey saddle is a degenerate critical point.

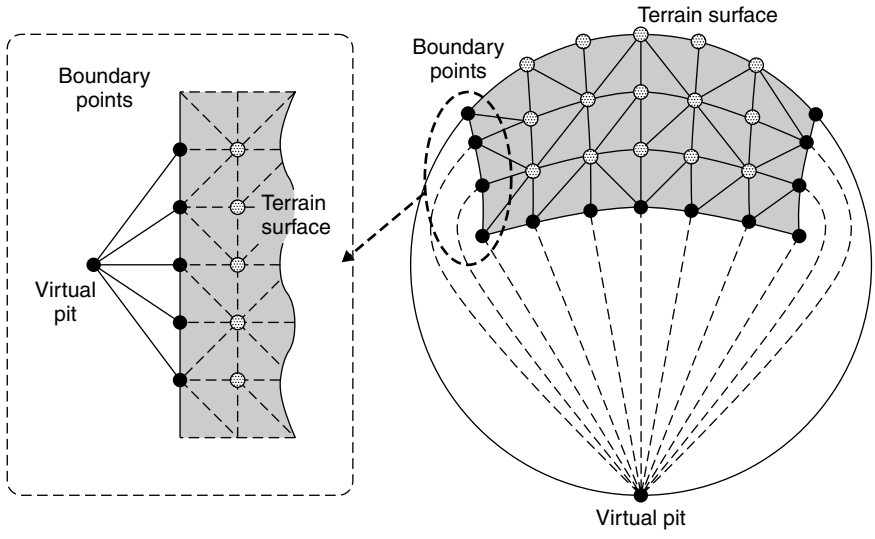

Figure 3.3 *A terrain surface and a virtual pit on a sphere*

to determine a unique surface interpolation from the given samples. For this purpose, we use *triangulation* because it offers the most commonly used linear interpolation and does not incur unwanted critical points that are likely to appear on higher-order interpolating surfaces. Note that the contour transitions with respect to the height value depend on the manner in which we triangulate the sample points. From this viewpoint, a method similar to the *Delaunay triangulation* should be employed in our framework because it can avoid thin triangles that are undesirable for sound linear interpolations.

For example, the grid samples in Figure 3.1 can be triangulated as shown in Figure 3.4. This is accomplished by partitioning the grid like a checkerboard and then splitting each square by either of the two triangles so that the new triangles form a smoother angle. Data-dependent triangulations (Dyn et al., 1990, Brown, 1991) will be the other candidates for this purpose.

The triangulation allows us to define the neighbours of each sample point and then introduce the criteria for critical points. Suppose that all critical points are non-degenerate at this stage. Here, the neighbours of the point P are defined to be points that are adjacent to P in the triangulation. In our implementation, each point P has a circular list of its neighbours in counter-clockwise (CCW) order with respect to the xy-coordinates. Now we are ready to introduce the criteria for critical points as follows:

peak	$\lvert\Delta_+\rvert = 0, \quad \lvert\Delta_-\rvert > 0,$	$N_c = 0$
pit	$\lvert\Delta_-\rvert = 0, \quad \lvert\Delta_+\rvert > 0,$	$N_c = 0$
pass	$\lvert\Delta_+\rvert + \lvert\Delta_-\rvert > 0,$	$N_c = 4$

where the following notation are used.

n the number of the neighbors of P

Δ_i the height difference between $P_i (i = 1, 2, \ldots, n)$ and P

Δ_+ the sum of all positive $\Delta_i (i = 1, 2, \ldots, n)$

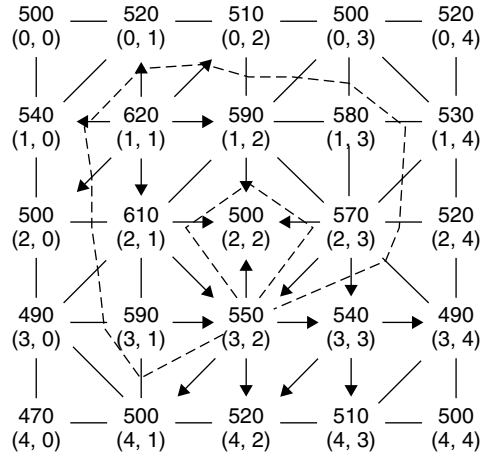

Figure 3.4 *A triangulation of the DEM in Figure 3.1 and contours at the height 550*

Δ_- the sum of all negative $\Delta_i (i = 1, 2, \ldots, n)$
N_c the number of sign changes in the sequence $\Delta_1, \Delta_2, \ldots, \Delta_n, \Delta_1$

Thanks to the above criteria, we can maintain the Euler–Poincaré formula (3.5) in the case of Figure 3.1. As shown in Figure 3.4, we extract only one pass $(3, 2)$ as well as the peak $(1, 1)$ and pit $(2, 2)$. Note that Figure 3.4 also shows contour lines at the height of the pass $(3, 2)$ at which the contour lines intersect each other.

Another important issue is how to handle the boundary sample points associated with the virtual pit. In our implementation, the virtual pit is assumed to be a point of the height $-\infty$. After triangulating the sample points, the virtual pit is inserted to the circular list of the boundary points so that the virtual pit and two neighbouring points on the boundary form a triangle as shown in Figure 3.3. In this process, the virtual pit is considered as a point exterior to the sample domain with respect to the (xy)-coordinates. Refer to Chapter 2 for more details.

3.2.3 Handling degenerate critical points

Now we can turn our attention to the degenerate critical points, which are classified into two cases: *level regions* and *duplicate passes*.

A level region is defined to be a set of connected sample points that are at the same height in the triangulation. This usually results from the discrete quantization, that is, the limited precision of height values. One solution to this issue is to group a set of level points together and regard the group as a single point. However, this is not a good idea when the region surrounds other critical points such as pits and peaks in its interior as illustrated in Figure 3.5(a).

The solution employed here is to introduce another ordering of the sample points in addition to the height ordering. This means that we compare two points using the second ordering if they are exactly at the same height. Actually, we have several

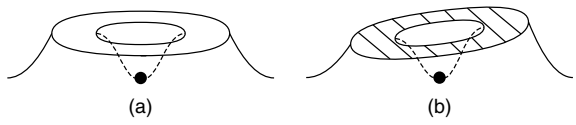

Figure 3.5 *A level region: (a) a level region surrounding a pit and (b) the effect of introducing the second ordering*

choices for the second ordering. The first and simplest choice is to use index numbers that are assigned to the sample points because every sample point is supposed to have a different index number. In our framework, however, we use the lexicographical ordering with respect to the xy-coordinates as the second ordering. More specifically, we compare the x-coordinates of the two points if two samples have the same height values. If they still have the same x-coordinates, we then compare their y-coordinates. Because the DEM is a set of samples on a single-valued function $z = f(x, y)$, there are no two samples that have identical xy-coordinates. Introducing this ordering is equivalent to inclining the height axis slightly only for the level regions as illustrated in Figure 3.5(b). This solution enables uniform data manipulation by converting the degenerate critical points to non-degenerate ones.

The second case is a duplicate pass at which, when the height value reduces, three or more cross-sectional contours merge into one contour, or one existing contour splits into three or more contours. An example of a duplicate pass and its neighbours including their height values is illustrated in Figure 3.6(a), where the shaded regions indicate cross sections at the height of the pass. In this case, it is necessary to decompose the duplicate pass into non-degenerate ones, because three contours are merged at the pass simultaneously. We then count the number of the passes after the decomposition. The criterion for passes is now modified as follows:

$$\textbf{pass} \quad |\Delta_+| + |\Delta_-| > 0, \quad N_c = 2 + 2m \ (m = 1, 2, \ldots)$$

where m is the number of the decomposed passes.

In our framework, we simply count the number of non-degenerate passes for the decomposition. Alternatively, we can actually split such a duplicate pass into several non-degenerate ones by inserting new edges and faces to the existing triangulation over the DEM. Such an example can be found in (Edelsbrunner et al., 2003b).

Consider how the algorithm handles the duplicate pass P shown in Figure 3.6(a). First, the algorithm generates $\{P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$ as the CCW neighbour list of P. After calculating Δ_+, Δ_-, and N_c of P, the algorithm simplifies the neighbour list as follows. While scanning the neighbour list, the algorithm defines a sequence of neighbours higher than P as an *upper sequence*. In this example, the upper sequences are $\{P_1\}$, $\{P_3, P_4\}$, and $\{P_6\}$. A *lower sequence* is defined in a similar manner. The simplified list is then obtained by choosing the highest neighbour out of each upper sequence and the lowest neighbour out of each lower sequence. In this example, the neighbour list is reduced to the list $\{P_2, P_3, P_5, P_6, P_7, P_1\}$ by removing P_4, because P_3 is higher than P_4 in the upper sequence $\{P_3, P_4\}$ (Figure 3.6(b)). Here, the simplified list is supposed to begin with the lowest neighbour, so that the algorithm can assign four alternating upper and lower neighbours to each decomposed (i.e. non-degenerate) pass.

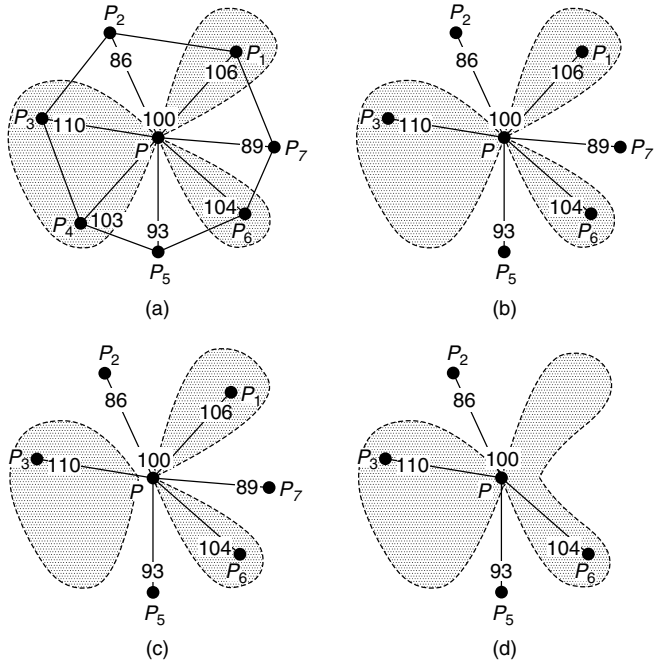

Figure 3.6 *A neighbour list of a duplicate pass: the height values of the sample points are indicated and the cross sections at the pass are shaded. (a) The original neighbour list, (b) the reduced neighbour list, (c) four representative neighbours and the corresponding imaginary cross sections for the first decomposed pass, and (d) for the second decomposed pass*

Since three contours merge at P simultaneously, the number of non-degenerate passes m is equal to 2. In order to decompose the duplicate pass, the algorithm first selects the last four neighbours P_5, P_6, P_7, and P_1 as the *representative neighbours* for the first decomposed pass. Figure 3.6(c) illustrates how the corresponding two contours merge while the third contour will come to join later. The same procedure is then carried out for the second decomposed pass after the last two elements P_7 and P_1 are eliminated from the list. This time, the third contour has intersected with the existing one as illustrated in Figure 3.6(d). In this way, the duplicate pass P is correctly resolved into two non-degenerate passes as shown in Figures 3.6(c) and (d), where four alternating upper and lower neighbours are appropriately assigned to each non-degenerate pass. The algorithm stores these corresponding four representative neighbours for each pass, for later use in tracing ridge and course lines (see Section 3.3).

3.3 CONSTRUCTING THE SURFACE NETWORK

This section describes the mathematical definitions of the ridge and course lines, followed by the definition of the surface network. After these mathematical fundamentals, an algorithm for constructing the surface network from the DEM together with the extracted critical points will be explained.

3.3.1 Ridge and course lines

The definitions of ridge and course lines are described as follows (Nackman, 1984). Let us represent a curve on the terrain surface as follows:

$$\boldsymbol{C}(t) = (C_x(t), C_y(t)) \tag{3.6}$$

where $t (\in \mathbb{R})$ is a parameter. Suppose that the curve $\boldsymbol{C}(t)$ satisfies the following differential equation:

$$\frac{\mathrm{d}\boldsymbol{C}}{\mathrm{d}t}(t) = -\left(\frac{\partial C_x}{\partial t}(t), \frac{\partial C_y}{\partial t}(t)\right) \quad \text{and} \quad \boldsymbol{C}(0) = \boldsymbol{C}_0 \tag{3.7}$$

where $\boldsymbol{C}_0$ is a point contained in the neighbourhood of the pass P. The curve $\boldsymbol{C}(t)$ is called a *ridge segment* if it converges to the pass P when $t \to \infty$. Conversely, consider the curve $\boldsymbol{C}(t)$ that satisfies the following differential equation:

$$\frac{\mathrm{d}\boldsymbol{C}}{\mathrm{d}t}(t) = \left(\frac{\partial C_x}{\partial t}(t), \frac{\partial C_y}{\partial t}(t)\right) \quad \text{and} \quad \boldsymbol{C}(0) = \boldsymbol{C}_0 \tag{3.8}$$

where $\boldsymbol{C}_0$ is again a point near the pass P. The curve $\boldsymbol{C}(t)$ is called a *course segment* if it converges to the pass P as t approaches ∞.

Actually, a *ridge line* is defined to be either a single ridge segment or a chain of connected ridge segments, while a *course line* is either a single course segment or a chain of connected course segments. These ridge and course lines cross every contour at right angles, and follow the steepest ascent and descent paths on the terrain surface, respectively.

It should be noted that the networks of ridge and course lines are dual of each other on the terrain surface. See Plate 1(a) for an example. This is because each ridge cycle contains only one pit in its inside while each course cycle contains only one peak. Furthermore, these two networks only intersect at passes, where two ridge and two coarse lines alternately appear when seen from the top. This systematic partition of the terrain surface helps us seek further applications, one of which will be presented in Section 3.5.2.

3.3.2 Surface network

Recall that a *critical point graph* is defined to be a graph such that it represents critical points as its vertices and relationships between them as its edges. The *surface network* (Pfaltz, 1976) is one of such critical point graphs, where its edge represents either a ridge or course line. Note that a ridge line ascends from a pass to a peak in the steepest direction on the terrain surface while a course line descends from a pass to a pit. More properties of the surface network are described in detail in Chapter 2. Figure 3.7 illustrates the surface network with cross-sectional contours. Here, a solid line represents a ridge line and a broken line represents a course line. As shown in

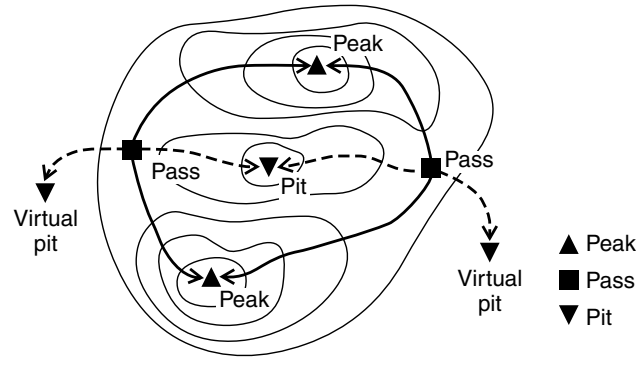

Figure 3.7 *The surface network and cross-sectional contour lines*

this figure, the surface network captures the configuration of the ridge and course lines successfully.

3.3.3 Algorithm for constructing the surface network

The algorithm for constructing the surface network is now presented as follows. What is needed here is to trace ridge and course lines emanating from passes on the terrain surface. As described in Section 3.2, the previous algorithm holds the corresponding four representative neighbours for each pass. Actually, these four representative neighbours serve as starting points for tracing such ridge and course lines in this algorithm, where the two upper neighbours lead to peaks (a peak) while two lower neighbours lead to pits (a pit). The surface network is then constructed by simply connecting every pass with the peaks and pits reachable from the pass through the ridge and course lines.

A ridge (course) line is traced in the algorithm as follows: Suppose we are at the starting point S. Since the ridge (course) line ascends (descends) in the steepest direction on the surface, we move to the highest (lowest) neighbour of S. This step can be carried out one by one until we reach a peak (pit). Note that if the two neighbouring samples are at the same height, we compare these two using the second ordering (i.e. the lexico-graphical ordering with respect to the xy-coordinates), which is introduced in Section 3.2. This implies that the number of the tracing steps is finite because the number of sample points on the DEM is also finite and we never visit the same point twice in a single tracing step. In this way, the surface network is constructed.

3.4 CONVERTING THE SURFACE NETWORK TO THE REEB GRAPH

The surface network is originally introduced to capture the configuration of critical points and their associated ridge and course lines on the terrain surface. Surprisingly, however, it can also work as an intermediate representation for constructing a *level-set graph* called the *Reeb graph*, which represents the splitting and merging of cross-sectional contours with respect to the height. This section first provides the definition

of the Reeb graph and then an algorithm for converting the surface network to the Reeb graph. In addition, several mathematical statements are also proved to justify the present algorithm.

3.4.1 Reeb graph

Along with the surface network, the *Reeb graph* (Reeb, 1946) is also one of the critical point graphs. In addition, it represents the topological transitions in cross-sectional contours as the corresponding height value changes. Let f be the height function of the terrain surface, and let P and Q be points on the surface. The Reeb graph of the height function f is obtained by identifying P and Q if the two points are contained in the same connected component of the cross-sectional contour of the surface at the height $f(P)(= f(Q))$. This means that a single connected component in a cross-sectional contour corresponds to a point on the edge of the Reeb graph (Figure 3.8). In particular, the vertex of the Reeb graph corresponds to the critical point of the height function f because at the height there must be some topological transition in the cross-sectional contour. Note that the definition of the Reeb graph is also presented in Chapter 2.

Figure 3.8(a) shows a mountain shape and its critical points, and Figure 3.8(b) shows the corresponding Reeb graph. In the remainder of this chapter, the critical points of the Reeb graph are arranged from top to bottom according to their height values, and represented by the symbols as shown in Figure 3.8(b) according to the type of each critical point. The Reeb graphs are also used for designing smooth surfaces (Shinagawa et al., 1991, Shinagawa and Kunii, 1991, Ikeda et al., 1992, Takahashi et al., 1997, Lazarus and Verroust, 1999) because they effectively represent the topological transitions of cross-sectional contours with respect to the height value.

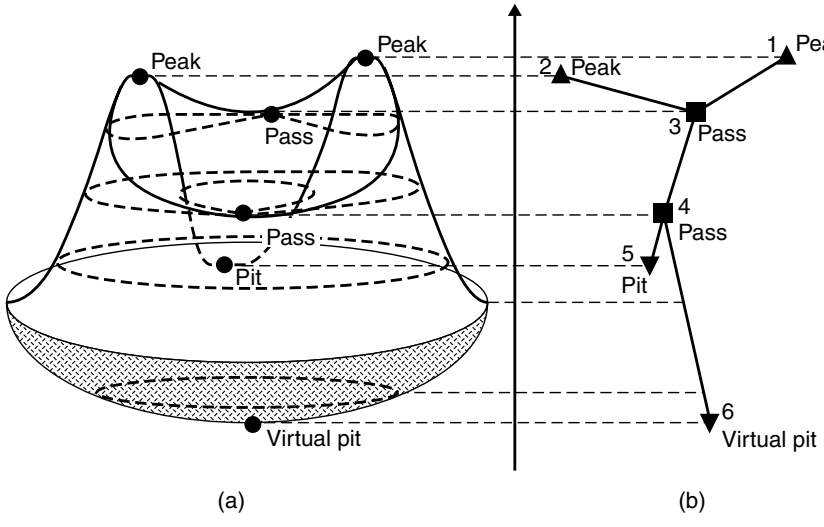

Figure 3.8 *(a) A mountain shape with its critical points and (b) the corresponding Reeb graph*

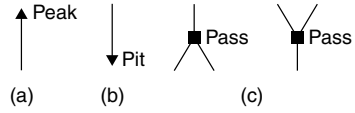

Figure 3.9 *Components of the Reeb graph around (non-degenerate) critical points: components around (a) a peak, (b) a pit, and (c) passes*

The above definition of the Reeb graph leads us to the following two statements.

Statement 1 *If all the critical points of the height function f are non-degenerate, the vertices (i.e. critical points) of the Reeb graph of f have the following properties (Figure 3.9):*

(1) If the critical point is a peak, it has only one downward edge, that is, the opposite end vertex is lower than the peak (Figure 3.9(a)).
(2) If the critical point is a pit, it has only one upward edge, that is, the opposite end vertex is higher than the pit (Figure 3.9(b)).
(3) If the critical point is a pass, it has either (a) one upward edge and two downward edges; or (b) one downward edge and two upward edges (Figure 3.9(c)).

The above statement directly follows from the contour transitions around the critical points, which are illustrated in Figure 3.2.

Statement 2 *Let $z = f(x, y)$ be a height function of a smooth surface. If f is represented by a single-valued function, the Reeb graph of f with the virtual pit becomes a tree, that is, the Reeb graph has no cycles.*

Suppose that the Reeb graph has a cycle. This means that the corresponding surface involves a torus according to the definition of the Reeb graph, which results in a contradiction.

3.4.2 Relationships between the surface network and the Reeb graph

Before going into details, we consider the two statements below, which help us understand relationships between the edges of the surface network and of the Reeb graph.

Statement 3 *An edge of the surface network corresponds uniquely to a path*[4] *in the Reeb graph (Figure 3.10).*

[4] A path of a graph is defined to be an alternating sequence of vertices and edges, which begins and ends with vertices, in which each edge is incident to the two vertices immediately preceeding and following it, and in which all the vertices are distinct (Harary, 1971).

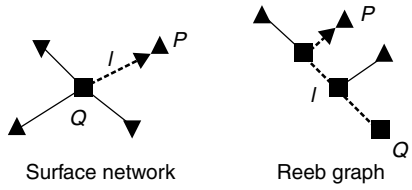

Figure 3.10 *Relationship between the edges of the surface network and the Reeb graph*

Consider a ridge edge l of the surface network and its corresponding end pass Q and peak P. Clearly, the corresponding Reeb graph has the identical pass Q and peak P as its vertices. This is illustrated in Figure 3.10. Note that the edge $\overline{QP}$ of the surface network represents a ridge line and thus a monotonous ascent path on the terrain surface. This means that there exists a path in the Reeb graph that monotonously ascends from the pass Q to the peak P. Here, the path is uniquely determined because the Reeb graph has no cycles due to Statement 2. Consequently, the statement is proved for the ridge edges of the surface network. The same can be applied to course edges, which concludes the proof.

We call a path in the Reeb graph a *monotonous ascent path* (*descent path*) if it corresponds to a ridge (course) line on the terrain surface (i.e. a ridge (course) edge in the surface network).

Statement 4 *For each edge incident to a pass in the Reeb graph, there exists either a monotonous ascent path or a monotonous descent path that contains the edge.*

Suppose that the pass has a "Y"-shaped branch in the Reeb graph, that is, the pass has two upward edges and one downward edge. In this case, the pass becomes a contact point between the two cross sections on the corresponding horizontal plane as shown in Figure 3.11. Thus, our task here is to make sure that the two ridge lines emanating from the pass definitely go into the two distinct cross sections respectively when they are projected onto the horizontal plane. This can be verified because the starting point for each ridge line is the representative neighbour in the upper sequence (see Section 3.2), and hence it is involved in the corresponding cross section individually. Note that the downward edge is obviously traced by the course lines because the pass has only one downward edge. Similar arguments can be applied to the passes that have branches upside down, which proves the correctness of the statement.

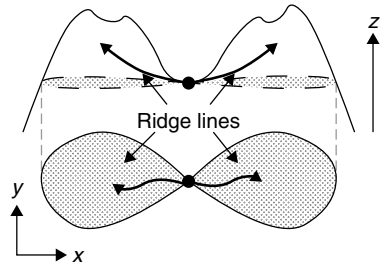

Figure 3.11 *Ridge lines emanating from a pass*

3.4.3 Algorithm for converting the surface network to the Reeb graph

Now we provide an algorithm for converting the surface network to the Reeb graph, followed by the mathematical statements that justify the validity of the algorithm. Here, it should be noted that the algorithm only depends on the mathematical properties of smooth surfaces, and does not depend on the discrete representations any more. In other words, once we successfully construct the surface network, we can obtain the Reeb graph only from its graph representation without referring to the original DEM.

The basic idea of the algorithm is to convert the ridge and course edges of the surface network to the edges of the Reeb graph by fixing the edges of the Reeb graph from its ends to the centre. This is done by first determining the edges incident to peaks and pits, and then changing passes into peaks or pits if each of the passes has already fixed two of its three incident edges in the Reeb graph. Recall that a pass of the Reeb graph has three incident edges, while a peak and a pit have only one edge as described in Statement 1.

Figure 3.12 shows how the algorithm converts the surface network of Figure 3.7 into the Reeb graph of Figure 3.8(b). Here, the graphs on the left correspond to the surface networks and the graphs on the right correspond to the Reeb graph, where the critical points of the Reeb graph are arranged from top to bottom according to their height values.

Figure 3.12(a) shows the initial states of the surface network and the Reeb graph. The vertex 1 is the first peak to be handled in the algorithm where its sole edge in the Reeb graph is determined. In this case, the algorithm adds the edge $\overline{13}$ to the Reeb graph because the vertex 3 is the highest vertex adjacent to 1 in the surface network. After having fixed the edge incident to 1 in the Reeb graph, the algorithm changes the connectivity of the surface network by removing the edge $\overline{13}$ and changing the edge $\overline{14}$ to the edge $\overline{34}$ (Figure 3.12(b)). A similar conversion process is applied to the peak vertex 2 (Figure 3.12(c)). Here, a new edge is not added to the surface network if the surface network already has an identical edge. Similar conversion processes are carried out for the pit vertices 5 and 6, where 6 represents the virtual pit. After all the peaks and pits are processed, the algorithm tries to find passes that have two fixed edges in the Reeb graph. At this stage, the passes 3 and 4 are the cases because two of the three incident edges have already been fixed for each of the passes in the Reeb graph. Here, the algorithm changes a pass to a peak if all the remaining incident edges are downward in the surface network, while it changes a pass to a pit if all the edges are upward. In this example, the vertex 3 is changed to a peak and the vertex 4 is changed to a pit in the surface network (Figure 3.12(d)). We repeat these conversion processes until all the edges of the Reeb graph are determined. Figure 3.12(e) shows the final results of this conversion process.

Note that if we cannot extract any passes from the terrain surface, we definitely have only a peak and a pit and just connect them with an edge to construct the Reeb graph.

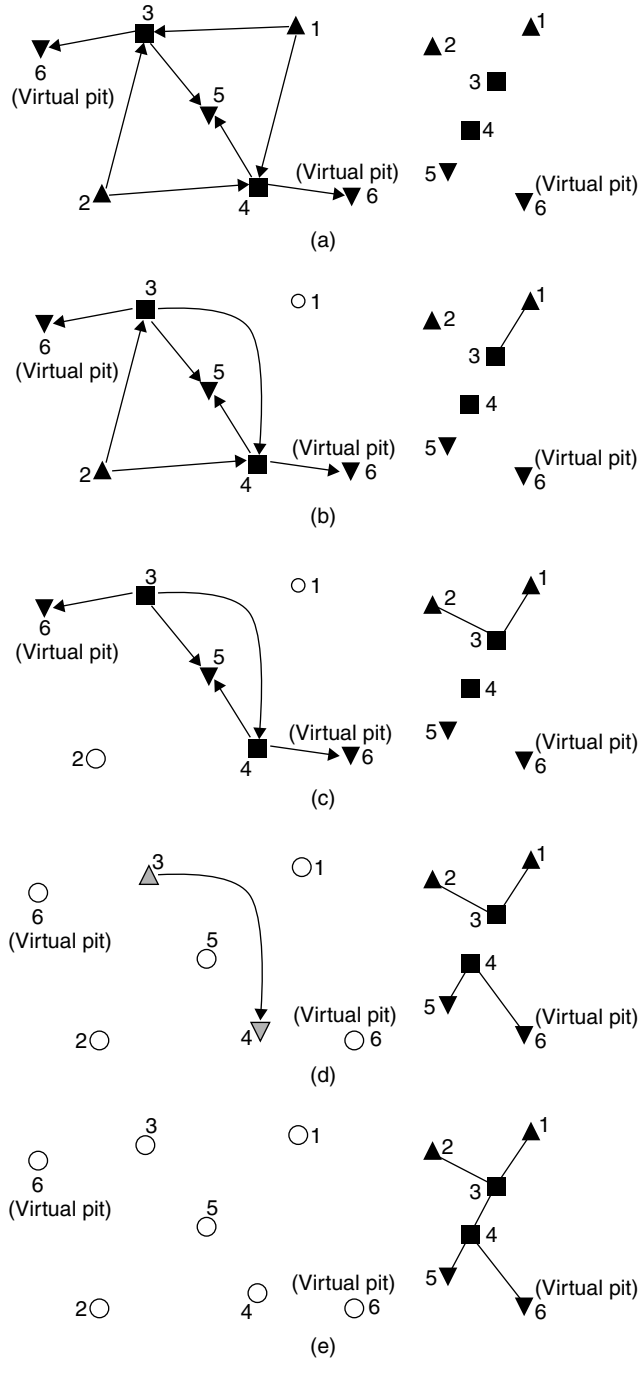

Figure 3.12 *Steps of the conversion algorithm: the graphs on the left show the surface networks and the graphs on the right show the Reeb graphs*

3.4.4 Validity of the algorithm

The validity of the conversion algorithm is finally justified by the following statements.

Statement 5 *The algorithm in Section 3.4.3 correctly converts the edges of the surface network to those of the Reeb graph.*

Let P be a peak and let $P_0, P_1, \ldots, P_n$ be its adjacent vertices in the surface network, where P_0 is the highest of all the adjacent vertices. From Statement 1, P has only one downward edge in the Reeb graph. From Statements 3 and 4, the paths monotonously ascending from $P_1, P_2, \ldots, P_n$ to P must pass through P_0 in the Reeb graph. This concludes that P_0 is on the way of the monotonously ascending paths from any of the vertices $P_1, P_2, \ldots, P_n$ to P. Hence, the edges of the surface network are correctly converted to those of the Reeb graph in the algorithm. The same procedure can be applied to the edges incident to pits.

Statement 6 *The algorithm in Section 3.4.3 takes a finite number of steps to finish the conversion.*

Let us show that the number of fixed edges in the Reeb graph increases monotonously. Recall that the Reeb graph is a tree, from Statement 2. In the algorithm in Section 3.4.3, all the peaks and pits are processed first. This means that the Reeb graph is determined from its end vertices and edges. After cutting off these end vertices and edges, the remaining undetermined part of the Reeb graph is still a tree. Since a tree has at least two end points (Harary, 1971), the algorithm changes two passes to peaks or pits at least because they must have two already fixed edges in the Reeb graph. In this way, the undetermined part of the Reeb graph shrinks step by step through the conversion processes until all its edges are fixed. This proves that the number of fixed edges in the Reeb graph increases monotonously.

3.4.5 Simplifying the Reeb graph

It is possible that the present algorithm extracts too many minor critical points because they are sensitive to high-frequency noise arising from discrete sampling and quantization. These minor critical points may hide the important surface topology of the terrain surface, and thus should be eliminated by consulting the global structure of the surface. This leads to the hierarchical representation of the surface topology that controls the surface shape details from a topological viewpoint. One of the excellent frameworks for this representation has been described in Chapter 2, in which a *weighted surface network* is used to estimate the importance of critical points. This section, on the other hand, describes a framework for estimating such importance by taking into account the Reeb graph instead.

Actually, the surface topology can be simplified by removing edges of lesser importance from the Reeb graph. Figure 3.13 shows such edge patterns to be eliminated through the simplification process. Here, each edge incident to either a peak or a pit is

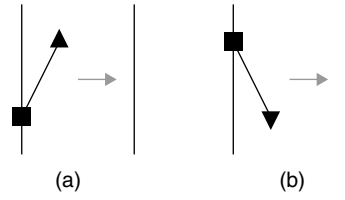

Figure 3.13 *Edge patterns eliminated in the simplification of the Reeb graph: (a) an edge between a peak and a pass and (b) an edge between a pit and a pass*

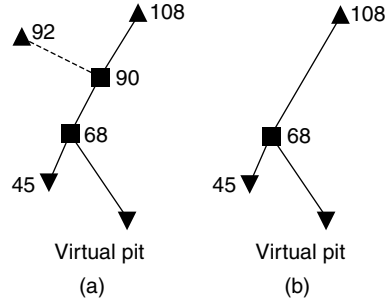

Figure 3.14 *A simplification step in the Reeb graph*

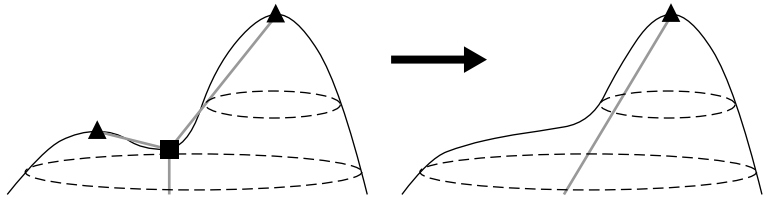

Figure 3.15 *Smoothing operation that corresponds to the contraction step*

examined to estimate its importance, and the edge of the least importance will be cut off. This contraction step is repeated until all the edges of the Reeb graph have higher importance values than the given threshold.

As an importance value for an edge, the difference in height between the two end critical points can be used. Figure 3.14 shows such an example. In Figure 3.14(a), the broken segment will be eliminated first because it has the smallest difference in height compared to all the edges incident to peaks and pits. Figure 3.14(b) shows the Reeb graph after the simplification step. Meanwhile, in the corresponding terrain surface, this single simplification step amounts to smoothing out a small peak as shown in Figure 3.15. It is also noted that the present framework estimates each local critical point by always referring to the Reeb graph that represents the topological skeleton of the surface. This means that the framework correctly finds the global surface topology by cutting off the minor local features.

3.4.6 Identifying contour embeddings on cross sections

Although the Reeb graph itself provides much information about the global structure of terrain surface, it also helps us identify the planar configuration of cross-sectional contours at any height value. According to Shinagawa et al. (1991), a pass has four types of contour transitions, taking into account their planar relative arrangement on the corresponding horizontal plane. These four types are indicated by the four vertical arrows in Figure 3.16, where the transitions on the right have an inclusion relationship, while the transitions on the left have no such nested structures. We use the term *contour embeddings* to denote this kind of contour configurations here.

Suppose that we scan the topological transitions of cross-sectional contours when lowering the corresponding height value. A *solid contour* is defined to be a contour whose interior points on the terrain surface with respect to the *xy*-coordinates are higher than its boundary points, while a *hollow contour* is defined to be a contour whose interior points are lower. Since the DEM represents a single-valued function, solid contours always expand when reducing the corresponding height value while hollow contours always shrink.

This allows us to specify the type of contour transition at a pass by identifying each of its incident edges as either solid or hollow, as shown in Figure 3.16. For example, we can find the contour embeddings from the Reeb graph shown in Figure 3.8(b) as follows: The first step is to start from the virtual pit 6. Since the DEM represents a single-valued function, the edge incident to the virtual pit is easily identified as solid. In this case, the edge $\overline{64}$ is identified as solid. Now we move from 6 to 4 by following the solid edge, and then identify the type of the pass 4 by referring to the four types of contour transitions in Figure 3.16. The figure suggests that the pass 4 corresponds to the component shown in Figure 3.16(c) because we reach the pass through the solid edge from the lower side. This lets us identify the edge $\overline{45}$ as hollow and the edge $\overline{43}$ as solid. Next, we continue to ascend through the Reeb graph up to the pass 3. At this point, we learn that the pass has the component shown in Figure 3.16(b), and identify

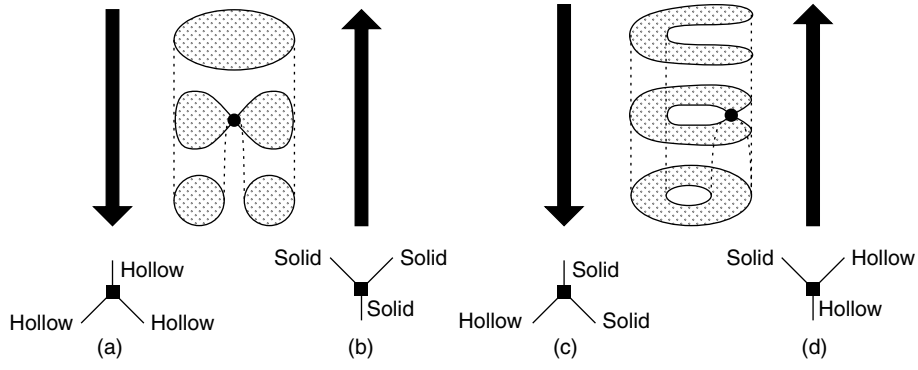

Figure 3.16 *Embeddings of contour transitions at passes: (a) one hollow contour splits into two hollow contours, (b) two solid contours merge into one solid contour, (c) one solid contour splits into one parent solid contour and one child hollow contour, and (d) one parent hollow contour and one child solid contour merge into one hollow contour*

the edges $\overline{13}$ and $\overline{23}$ both as solid. In this way, we can extract the contour embeddings by carefully tracing the constructed Reeb graph.

3.5 EXAMPLES

This section first shows geographical features extracted using the present algorithms, and then its application example.

3.5.1 Characterise terrain surfaces

The present algorithms are applied to the DEMs of the Hakone area, which is one of the well-known tourist areas in Japan because of its scenic crater lake called Lake Ashi. Plate 1(a) shows the ridge and course lines together with the critical points, which are obtained using the present algorithms. Here, the red, green, and light blue points represent peaks, passes, and pits, and the yellow and purple lines trace ridge and course lines, respectively. Note that the extracted critical points satisfy the Euler–Poincaré formula, and the ridge and course networks are dual of each other on the semi-transparent terrain surface. It can be seen in the figure that the present algorithms successfully trace the outer rim of the crater as ridge lines. The side view of the resultant Reeb graph with the semi-transparent terrain surface is shown in Plate 1(b), where the edge incident to the virtual pit is omitted. These results demonstrate the feasibility of the present algorithms.

3.5.2 Guide-map generation

Hand-drawn area guide maps often convey intuitive information about the configuration of landmarks on the terrain surface. In order to generate such guide maps automatically from the DEM, it is necessary to identify feature areas such as mountains and lakes by taking into account the terrain surface topology. The network of ridge and course lines offers an excellent partition of the terrain surface for this purpose, where each peak is surrounded by a course cycle while each pit is surrounded by a ridge cycle.

Once we identify the feature areas, we can assign a vista point to each area to simulate hand-drawn guide maps that deviate slightly from the exact perspective projections. This sort of non-perspective projection is called *surperspective projection* (Takahashi et al., 2002) in this chapter. Plate 1(c) presents such examples in which the areas containing the mountain and lake are deformed in the projected images (Takahashi et al., 2002). The upper left window shows an ordinary perspective image of the Hakone area. Since the mountain hides the lake in this figure, we move the mountain to this side to make the lake clearly visible as shown in the upper middle window. On the other hand, the view direction of the lake should be changed so that we can recognise the shoreline of the lake easily as shown in the upper right window. The bottom window presents the final surperspective guide-map image obtained by applying

the last two effects together. In this way, the present projection framework actually provides a useful means of identifying the significant landmarks of the terrain surface.

3.6 CONCLUDING REMARKS

This chapter has presented mathematical fundamentals and its associated algorithms for extracting surface topology from DEMs robustly. In the present framework, critical points such as peaks, passes, and pits are extracted so that they satisfy the Euler–Poincaré formula. The surface network is constructed from the extracted critical points by tracing the ridge and course lines traversing between them. This chapter also presented an algorithm for converting the surface network to the Reeb graph by only taking into account the properties of smooth surfaces. Examples are shown to demonstrate the feasibility of the present algorithms.

The concept of the surface network has inspired other algorithms that characterise the terrain undulations. Wolf (Wolf, 1990, 1991a) developed the concept of a weighted surface network that has a weight value for each of its edge. Wood et al. presented an algorithm for calculating the surface network by fitting a bivariate quadratic surface to the given DEM (Wood, 1998). Edelsbrunner et al. presented another interesting algorithm that partitions a DEM into topological primitives called *Morse–Smale complexes* (Edelsbrunner et al., 2002) by tracing ridge and course lines, and introduced geometric measures (Edelsbrunner et al., 2002) to control their multi-resolution representations.

Note that the definitions of ridge and course lines employed in this chapter are orientation-dependent features, that is, they depend on the direction of the height axis. There is another definition of such feature lines that is orientation-independent. In that case, ridge points are defined to be local positive maxima on the curve of the maximal principal curvature while course points are defined to be local negative minima on the curve of the minimal principal curvature. This definition has close relationships with shape curvatures and *medical axis transforms* that have been used as common tools in characterising surface properties. Readers can refer to a textbook (Porteous, 1994) and papers (Anoshkina et al., 1994a,b).

Calculating level sets of surfaces has been one of the significant topics in the field of shape modelling. Primary level-set computation was studied for characterising DEMs (Kweon and Kanade, 1994, Takahashi et al., 1995), and then free-form surfaces handled in contemporary computer-aided design (CAD) systems (Lazarus and Verroust, 1999).

Extending these level-set-based frameworks to one-dimensional higher cases has also become important. Examples include volume data, which can be considered as a set of voxel samples on the single-valued function $w = f(x, y, z)$. Actually, level sets for volumes offer a crucial insight into its complicated inner structures. Bajaj et al. developed a framework for exploring complicated surfaces and volumes, by computing level-set graphs called *contour trees* (Bajaj et al., 1997) that are closely related to the Reeb graphs. They used an algorithm that constructs the contour trees with minimal computational complexity, which was successfully developed by van Kreveld et al., 1997[5]. The computational complexity was then improved by Tarasov

[5] See Chapter 5 for details.

and Vyalyi, 1998. Furthermore, as an extension of the algorithm, Carr et al., 2003 developed an excellent algorithm that computes contour trees from objects of any dimension. While these algorithms are elegant from a computational point of view, they only pursue the changes in the number of connected components and do not track the change in topology (i.e. genera) of varying isosurfaces. Recently, Pascucci and Cole-McLaughlin, 2002) formulated an algorithm for identifying the topology (i.e. genus) of an isosurface at any point of the contour tree.

The framework described in this chapter has also been extended to volumes in order to generate their comprehensive visualisation images (Takahashi et al., 2004). In this framework, spatial configurations of isosurfaces are also analysed to emphasise significant inclusion relationships of isosurfaces in the final visualisation results.

ACKNOWLEDGEMENTS

I have benefited from discussions with Yoshihisa Shinagawa, Bianca Falcidieno, Michela Spagnuolo, Chiew-Lan Tai, Issei Fujishiro, Yuriko Takeshima, and Gregory M. Nielson. This work has been partially supported by Japan Society of the Promotion of Science under Grant-in-Aid for Young Scientists (B) 14780189.

4

Construction of Metric Surface Networks from Raster-Based DEMs

Bernhard Schneider and Jo Wood

4.1 INTRODUCTION

Although the Surface Network has been recognised as a powerful tool for surface analysis for several decades, remarkably few authors have actually presented algorithms for the extraction of this topological network from digital elevation models (DEMs). Among the methods to extract surface networks, either fully or partially, from raster-based DEMs are the works of Fowler and Little (1979), Takahashi (presented in Chapter 3), Wood (1998), Wood and Rana (2000), and Schneider (2003). This article discusses the latter two approaches.

Starting from a same data model and from a common comprehension of the problem at hand, Schneider (2003) and Wood (1998), Wood and Rana (2000) develop two contrasting approaches yielding different results. Both authors work on raster-based DEMs, and both extract the surface network elements from local surface patches specified from square windows of $n \times n$ cells. While Schneider relies on the mathematically simple bilinear interpolation scheme, Wood uses the more complex and more adaptable bi-quadratic interpolation. The method of Schneider is rigorous in terms of continuity, and, thus, it is deterministic in the sense that the results are uniquely defined for each given data set. Abandoning this determinacy, the method of Wood offers the user the means to account for specific topographic properties (i.e. roughness and level of generalisation) and, most importantly, to specify the desired analysis scale. The method starts from the (geo-)morphologic form of the network elements and, hence, utilises

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

the semantics of surface forms, thereby introducing scale as an inherent component of the extraction process.

The comparison of the two methods highlights the difficulties that impede the construction of surface networks. Designing extraction algorithms involves setting priorities and finding compromises when dealing with these difficulties. This chapter offers an explanation and evaluation of the impact of these properties and compromises in the network extraction process.

4.1.1 Continuity constraints and definitions

Fowler and Little (1979) propose deriving the network directly from raster data, without prior specification of a continuous surface. Critical points (i.e. pits, peaks, and passes) are detected with the method of Peucker and Douglas (1975) in which the classification of a raster point depends on the relative elevations of the eight direct neighbours. The approach has a number of weaknesses that are related to the absence of a continuous surface:

- The locations of the critical points and of the vertices of the critical lines (i.e. valleys and ridges) are limited to the given raster points.
- Horizontal areas are not taken into account. Thus, the detection of critical points may be incomplete.
- The method detects too many passes, especially along crests and valley lines. As a result, the Euler formula is not satisfied (Takahashi et al.,1995).

Takahashi et al. (1995) reason that it is not possible to extract critical points correctly and to derive a consistent surface network solely from a set of discrete data points. Hence, a surface must be specified from the raster data prior to the surface network extraction.

On surfaces expressed as bivariate functions $z = f(x, y)$, pits, peaks, and passes are defined as points where the first derivatives in x and y are 0. Second derivatives are used to distinguish between the three types of critical points. For this reason, many authors (e.g. Wolf, 1991b, Rana and Wood, 2000) request surfaces to be second-order continuous for the definition of the surface network elements. Unfortunately, real surfaces often have breaks in slope at many points and along many lines. Furthermore, their digital representations are not always continuously differentiable everywhere, if, for instance, the surface is represented by linear triangle facets of a triangulated irregular network (TIN). (Henceforth, k-order differentiability is referred to as C_k-*continuity*.)

The following definitions of critical points and lines are valid for all globally C_0-continuous surfaces that are composed of piecewise C_1-continuous surface patches (De Floriani and Puppo, 1992). They are consistent with the definitions given by Wolf (1991b) and are considered alternatives to (or generalisations of) the usual mathematical definitions. The definitions require the prior definition of the term "region":

- A *region* is the area of a C_0-continuous surface enclosed by a closed boundary without gaps or self-intersections. A region includes its boundary and does not have holes.

- A point P is *contained by a region* if it is inside the region but not on the boundary.
- A point P with elevation z on a C_0-continuous surface is called *pass* if the boundaries of all regions containing P but no other critical points of the surface intersect the contour of elevation z at least four times.
- A point P with elevation z on a C_0-continuous surface is called *pit* (*peak*) if there exists a region containing P where all elevations are higher (lower) than z.
- *Valleys* (or *valley lines*) are paths of steepest descent starting at passes. At each point P of the path of steepest descent where the surface is continuously differentiable, the tangent to the path of steepest descent coincides with the surface's aspect at P. If the surface is not continuously differentiable at P, P is at the border of two or more continuously differentiable surface patches. In this case, the path of steepest descent continues on the surface patch with the steepest slope at P. If all adjacent surface patches are higher than P in the vicinity of P, the path continues along the patch border with the steepest slope.
- *Ridges* (or *ridge lines*) are defined analogously to valley lines.

4.2 CONCEPTUAL MODEL

4.2.1 Basic conceptual model of the metric surface network

In the metric representation of the surface network (also called geometric or weighted), all points and lines are stored with their coordinates. Passes, pits, and peaks together with the valleys and ridges form the frame of the network (Figure 4.1). Furthermore, a *drainage area* (or *dale*) bound by sequences of ridges, passes, and peaks can be identified for each pit. This way, each pit is separated from its surrounding pits by ridges. Likewise, there exists a *mountain* (or a *hill*) for each peak, and there is always a valley line between two neighbouring peaks.

4.2.2 Considering the edge of the area of interest

The edge of the area of interest impairs the completeness and consistency of the constructed surface network, mainly because of two reasons:

- Valleys and ridges may not be found because the pass forming their starting point is located outside the area of interest. Imagine a pit with a valley leading towards it. If the start of this valley, that is, the corresponding pass is located outside the area of interest, the valley cannot be extracted, causing incompleteness. As a result, the separation between two peaks may be missing, adding inconsistency.
- Pits and peaks may be located outside the area of interest. Imagine a valley starting at a pass and leaving the area of interest at some point at the area's edge. The tracing of the valley is prematurely terminated and the valley does not end in a pit, preventing consistency of the surface network.

Three approaches have been suggested in the literature to cope with the problem:

- Analysis of surface networks started with the assumption that the area of interest has constant elevation, that is, that the area of interest is bound by a single

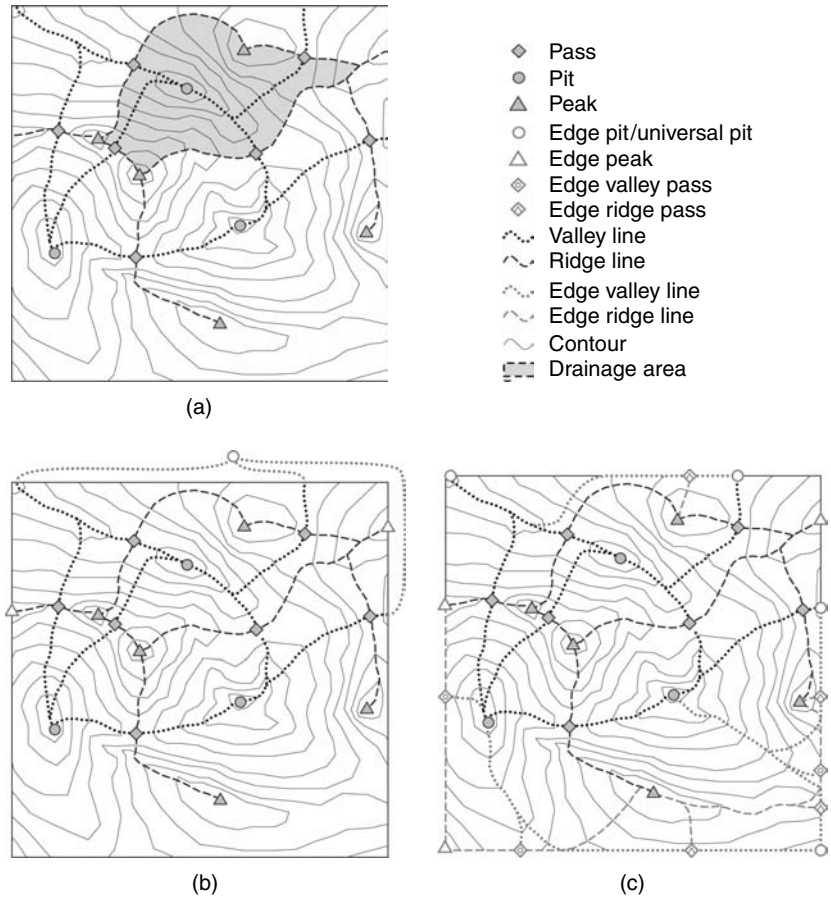

Figure 4.1 *Metric surface network of a hypothetical surface sketched with the help of contours. (a) A selected drainage area is highlighted; (b) the network is complemented with a universal pit; (c) the network is complemented with edge features*

contour (Pfaltz, 1976). This supposition allows imbedding the topology of the surface network consistently in the graph theory. Although the – usually rectangular – digital models of terrain do not conform to this assumption in general, one can complement the terrain model with an infinite plane of constant elevation outside the model extent. For analytical convenience, this plane is either lower than the terrain model's minimum elevation (forming a *universal pit*, cf. Figure 4.1(b)) or higher than the model's maximum elevation (forming a *universal peak*). In the former case, the universal pit can intuitively be interpreted as the ocean all rivers eventually drain into. As a result, new critical points (i.e. pits *or* peaks) are inserted at the model's edge, and valleys or rivers are (topologically) extended over the model's edge to reach the universal pit or peak, respectively.

- Schneider (2003) argues that the information content of the surface network is maximised if the exterior of the terrain model is considered void. As a result,

pits *and* peaks may also occur at the model's edge, for example, where a valley leaves the area of interest. Furthermore, two new types of passes are introduced. Imagine a valley entering the area of interest. At the entry point, the elevation profile along the model's edge has a local minimum. At this location, an *edge valley pass* is identified as forming the start of the valley. Additionally, two ridges start at this pass, both raising along the model's edge to either side of the pass. The ridges may deviate from the model's edge if the direction of steepest ascent (i.e. the negative aspect vector) points to the inside of the model at some location along the edge. Analogously, edge ridge passes are the start of a ridge and two valleys. This approach of dealing with the model's edge adds four new types of critical points to the conceptual model (Figure 4.1(c)) and makes virtual network elements obsolete.

- A rectangular surface model may well contain complete "subgraphs" within its bounds that are topologically consistent with the Euler formula. Analysis may be performed on these complete subgraphs while excluding the incomplete graphs that intersect with the model's rectangular border. This is akin to the hydrological processing of irregularly bounded drainage basins with a rectangular DEM.

4.2.3 Intersection of valley and ridge lines

Previous publications state that valleys and ridges must not intersect, unless meeting a pass (Pfaltz, 1976, Wolf, 1984, 1990). While this rule is valid for C_k-continuous surfaces, $k > 0$, it may be violated in specific configurations of real topographies.

Figure 4.2 depicts a pass located on a crest (left side) forming the start of a valley line. This valley runs down a gulch that opens into an alluvial fan at some point P. The steepest path of descent at that transition point (from concave gulch to convex fan) is not necessarily straight onto the fan, but possibly to the left or right side of it. If, at the same time, there exists a pass at the bottom of the alluvial fan (right in Figure 4.2), a ridge raises over that fan towards the transition point P. Again, the continuation will be the steepest path of ascent from that point and may be to the left or right of the gulch. As a result, the two critical lines may intersect at P. Such intersections must be explicitly addressed because they have fundamental implications for the topological surface network graph. They can either be tolerated (thus failing to be consistent with Euler's formula), or they must be treated as special cases by, for example, inserting extra passes or classifying their intersections separately.

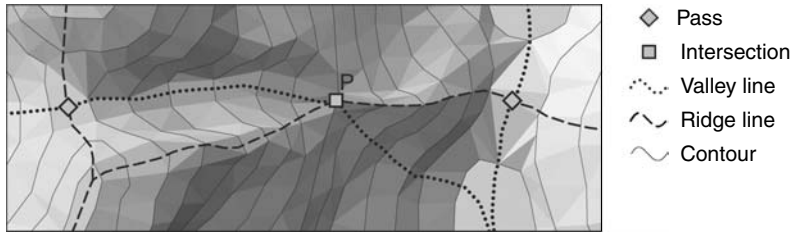

Figure 4.2 *Extended geometric surface network*

4.2.4 Horizontal areas

Horizontal areas are frequent features in real surfaces. (One may argue that, in topographic surfaces, *exactly* horizontal areas other than water surfaces are rare.) In digital surface representations, they are even more frequent, because the z-values of data points may be rounded to the whole number, or because all corner points of triangular patches may lie on the same contour. Horizontal areas may be polygons, or they may only consist of linear features such as horizontal crests.

If the immediate neighbourhood of a horizontal area is lower than the area itself, the horizontal area represents a peak. Likewise, horizontal areas may represent pits or passes. In any case, the definitions given in the introduction remain valid if "point" is comprehended as morphometric feature represented by a point, a line, or an area.

4.3 EXTRACTION FROM BILINEAR SURFACE PATCHES

4.3.1 The bilinear interpolation scheme

As has been stated in the introduction, a surface must be specified from the data prior to the surface network extraction. The bilinear interpolation is a straightforward and suitable choice yielding a C_0-continuous surface (Figure 4.3). For each raster cell, a surface patch is specified with

$$z = axy + bx + cy + d \tag{4.1}$$

where the coefficients a, b, c, and d are determined with the help the four corner points. The resulting surface patch is deterministic, that is, it is specified from the data directly, no intermediary data (e.g. vectors) need to be calculated with some complementary algorithms, and no parameters to be specified by the user are involved. The global surface composed of the bilinear patches is, of course, characterised by apparent artefacts caused by the regular pattern of break lines, that is, by the borders between the cells along which the surface is not continuously differentiable. On the other hand,

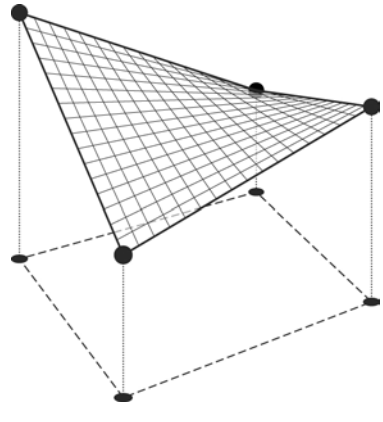

Figure 4.3 *Bilinear surface patch specified over a raster cell*

the bilinear interpolation scheme is conservative in the sense that the resulting surface does not overshoot or exhibit other artefacts known from higher-degree polynomial surfaces (Florinsky, 2002, Schneider, 2001). Furthermore, the bilinear scheme is computationally efficient, and extraction of topographic features is straightforward.

The derivatives of equation (4.1) with respect to x and y are

$$z'_x = ay + b \tag{4.2}$$

$$z'_x = ax + c \tag{4.3}$$

Thus, slope does not change if one moves parallel to the x- or y-axis. Hence, the profiles of the surface parallel to the coordinate axes are straight lines. This observation facilitates the detection of critical points and the tracing of critical lines.

4.3.2 Extraction of critical points

Bilinear surfaces do not have local maxima or minima. (There is only one point where the derivatives in x and y are both 0, and this point is a pass.) For this reason, the extremes of quadrilateral bilinear surface *patches* with straight borders parallel to the coordinate axes are at their corner points. Thus, only these corner points, that is, the given DEM data points, need to be analysed to find pits and peaks.

According to the above definitions, a grid point T is, for instance, a peak if there exists a region containing T in which the four adjacent bilinear surface patches are lower than T (Figure 4.4). This is the case if the corresponding tangential planes at T are lower than T. Such configurations can be identified by comparing the direct neighbours of T because each tangential plane at T is determined by the according straight lines from T to the direct neighbours. For example, if both Q and S are lower than T, then the tangential plane – which is defined by the three points QST – is lower than T within cell C_1. Thus, it follows that T is a peak if the four direct neighbours Q, S, U, and W are lower than T. Pits are detected analogously.

Passes may – in contrast to pits and peaks – occur not only at grid points but also within grid cells. If a grid point T is a pass, then its direct neighbours must be alternately

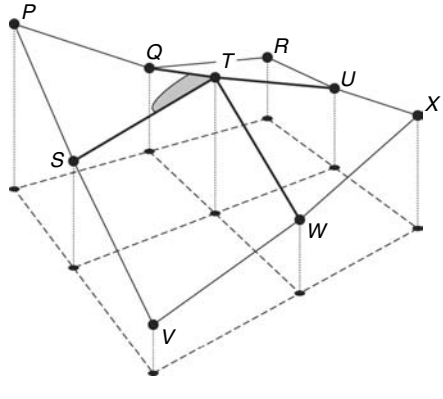

Figure 4.4 *Scheme of eight-point neighbourhood of raster point T and tangent plane at this point in cell PQTS*

higher and lower (e.g. Q and W are higher than T; S and U are lower than T). If a pass is located within a cell, then the corner points of this cell are alternately higher and lower. For instance, if P is higher than S and Q, and T is higher than S and Q, then there exists a pass within the bilinear surface patch (Figure 4.4). The surface at the pass is horizontal, that is, the first derivatives in x and y are 0, which facilitates the calculation of the pass' exact coordinates.

4.3.3 Tracing critical lines

Critical lines are traced and constructed vertex by vertex. At each new vertex, the local configuration of the raster points (i.e. of the adjacent surface patches) is examined, and the next vertex calculated accordingly. A number of different situations can occur, all of which are listed in Table 4.1. Figure 4.5 illustrates a typical situation.

The new vertex B of a valley line is found to be located on a cell edge (Case ***b*** in Table 4.1). The elevations of the corner points of the two adjacent cells 1 and 3 are examined, and the edge is found to be a ravine, that is, it is water-collecting (Case ***ba***). Furthermore, the edge drops towards the raster point with elevation 42. Hence, the next vertex C is located on this raster point (Case ***baa***). Vertex C becomes the new vertex (Case ***a*** in Table 4.1), and again the elevations of all corner points of the four adjacent cells are analysed. As in the previous situation, the cell edge towards the raster point with elevation 40 is a ravine (Case ***aa***). However, after one-third of the edge, Cell 4 becomes lower than the cell edge. (The surface patch of Cell 4 is horizontal along Profile p.) As a result, the edge stops to be water-collecting after one third, and the path of steepest descent deviates from the edge to enter Cell 4. This location on the edge is inserted as new vertex D (Case ***aab***).

Table 4.1 *Vertex locations, possible continuation of critical lines, and possible locations of next vertex*

Vertex location	Possible continuation	Possible next vertex
a raster point	***aa*** along cell edge	***aaa*** next raster point along edge
		aab point on cell edge
	ab through cell interior	***aba*** diagonal raster point
b point on cell edge	***ba*** along cell edge	***abb*** point on opposite cell edge
		baa next raster point
		bab point on cell edge
	bb through cell interior	***bba*** diagonal raster point
		bbb point on opposite cell edge
c point inside cell	***ca*** any direction	***caa*** raster point
		cab point on cell edge

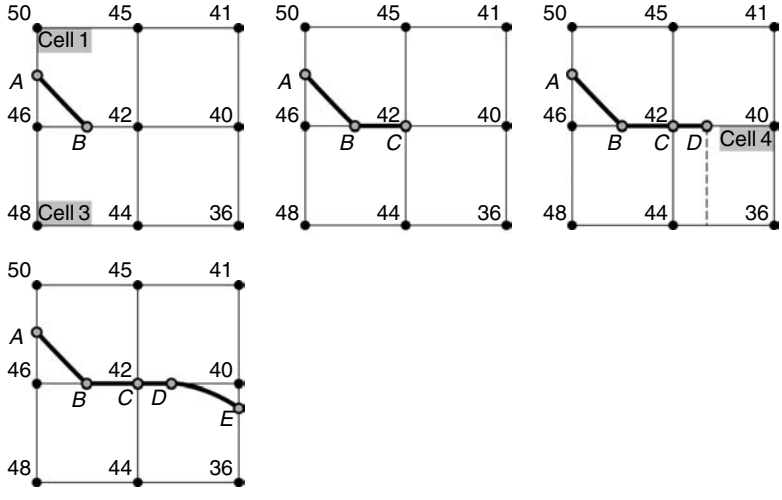

Figure 4.5 *Four steps of the tracing of a valley line*

From D (Case ***b***), the critical line continues through the cell (Case ***bb***). Two possible methods to trace the path of steepest descent over the bilinear surface patch are as follows:

- For each point of the surface patch, there exists an aspect vector. These vectors define a vector field. The path through a vector field is called *streamline* and can be calculated. In the case of a bilinear surface, the resulting curve is a hyperbolic function.
- Instead of calculating the entire path through the cell at once, it is approximated with small steps. At a point of the path, the aspect of the bilinear surface is calculated, and a small step of predefined length (e.g. a specific fraction of the cell size) is marked off along this direction. This process is repeated until the edge of the cell is reached, yielding Vertex E in Figure 4.5 (Case ***bbb***).

Critical lines are terminated at one end by ordinary passes, that is, at passes represented by point objects, two valley and two ridge lines intersection. (Non-ordinary passes, i.e. passes represented by line and area objects, are discussed in the following section below.)

- If the pass coincides with a grid point, the steepest paths will continue along the cell edges (Case ***aa*** in Table 4.1). The paths may reach the next grid point along the edge (Case ***aaa***), or they may depart from the edge at some point between the two grid points (Case ***aab***), depending on the two adjacent bilinear surface patches. (Edge passes are always located at grid points. Hence, the construction of valleys and ridges starting at these passes is started with this procedure, although only three critical lines need to be traced.)
- If the pass is inside a grid cell (Case ***c*** in Table 4.1), then the first segments of the steepest paths are straight and parallel to the cell diagonals until they reach the cell's edges (Case ***cab***) or corner points (Case ***caa***). (It can be proven that the paths of

steepest descent and ascent on a bilinear surface starting at the surface's pass are straight and form an angle of 45° with the coordinate axes.)

4.3.4 Dealing with horizontal areas

If two adjacent grid points have the same elevation, they constitute a horizontal edge. Multiple horizontal edges may form *horizontal edge groups* that may include horizontal cells (where all four corner points have the same elevation).

If a pass is represented by a horizontal edge group, it may be the start of more than two valley lines and two ridge lines. The number of steepest paths is computed by analysing all grid points directly neighbouring the horizontal edges. The exact location of each steepest path's start depends on the individual configurations of the transition from the horizontal edge group to its neighbourhood, and on the way this transition is morphologically interpreted. Likewise, a heuristic approach needs to be applied to trace steepest paths through horizontal regions.

4.4 EXTRACTION BASED ON BI-QUADRATIC POLYNOMIAL SURFACE APPROXIMATION

Like the method discussed above, the approach of Wood (1998) and Wood and Rana (2000) is based on the specification of local surface patches from the DEM data. However, while the former method is based on a purely mathematical comprehension of the surface network elements, the approach of Wood (1998) and Wood and Rana (2000) builds from a (geo-) morphometric analysis of the surface portrayed by the data. The surface patches determine the morphometric feature–type of all raster points, which forms the bases for identifying passes and, hence, for initiating the tracing of valleys and ridges. This approach offers the possibility of assigning a thresholded weighting to pass features, and thus, a control over the complexity of the derived network.

4.4.1 Morphometric feature–type classification

All points of the surface are assigned one of the morphometric classes *pit*, *peak*, *pass*, *valley*, *ridge*, and *plane* (Peucker and Douglas, 1975). The classification is based on slope and curvature measures at the surface points. More specifically, cross-sectional curvature needs to be measured to differentiate between ridge, valley, and plane:

- At non-horizontal points, *cross-sectional curvature* is the curvature at the point of interest of the line produced by the intersection of the surface and the plane defined by surface normal and the aspect vector at that point.

  At horizontal points (slope = 0), cross-sectional curvature is not defined.

Furthermore, maximum and minimum convexity values need to be derived from the surface (Young, 1978). Table 4.2 lists the six morphometric types and the corresponding slope and curvature values.

Table 4.2 *Feature-type classification rules*

Feature name	Slope	Cross-sectional curvature	Maximum convexity	Minimum convexity
Pit	0	–	<0 (concave)	<0 (concave)
Peak	0	–	<0 (convex)	>0 (convex)
Pass	0	–	<0 (convex)	<0 (concave)
	0	–	>0 (concave)	>0 (convex)
Valley	<0	<0	–	–
	0	(convex)		
	0	–	<0	0
Ridge	>0	>0	–	–
	0	(concave)		
	0	–	>0 (concave)	0
Plane	>0	0	–	–
	0			
	0	–	0	0

4.4.2 Slope and curvature tolerances

Only a small number of raster points, if any, have a slope value of exactly 0. Consequently, the strict application of the above classification rules yields a very small number of pits, peaks, and passes. To account for this problem, a slope tolerance is introduced. All raster points with absolute slope values smaller than the slope tolerance are considered horizontal. This tolerance value can be used to control the complexity of the derived network by determining the number of initial passes at which ridges and channels are deemed to intersect. A workable method for determining a suitable tolerance in the selection of the region of interest in a conic section is given below.

A similar problem occurs with the classification of plane cells that requires curvature values to be 0. Strictly applying the above rules results in most (or all) raster points – except the critical points – to be classified as either valleys or ridges. In order to narrow the linear features and to increase the number of raster points classified as planes, a curvature tolerance is introduced. Again, absolute curvature values smaller than the tolerance are considered to be 0.

4.4.3 Bi-quadratic polynomial approximation

As has been stated above, the feature-type classification requires (local) surface to be specified from the data. A suitable and sound choice is the bi-quadratic polynomial (Evans, 1980)

$$z = ax^2 + by^2 + cxy + \mathrm{d}x + ey + f \tag{4.4}$$

Evans (1980) presents a method to fit bi-quadratic polynomials to 3×3 windows of raster-based DEMs with the method of least square differences (Figure 4.6). Wood (1996a,b, 1998) extends this approach to fit the polynomial to a quadratic $n \times n$ window of arbitrary size n (where n needs to be an odd number). In this way, scale dependency is explicitly introduced to surface analysis: larger analysis window sizes (i.e. larger n's) correspond to the analysis on a smaller level of scale. The (theoretically) largest possible analysis scale is given by the resolution of the raster DEM, the smallest scale by the size of the model (i.e. by the smaller of columns and rows).

Bi-quadratic surfaces can be interpreted as defining conic sections (Figure 4.7). Neglecting the case where a, b, and c of equation (4.1) are 0 (i.e. where the surface is planar), three possible conic section types can be identified as follows (e.g. Kindle, 1950):

- If $4ab - c^2 > 0$, the conic section is elliptic.
- If $4ab - c^2 = 0$, the conic section is parabolic.
- If $4ab - c^2 < 0$, the conic section is hyperbolic.

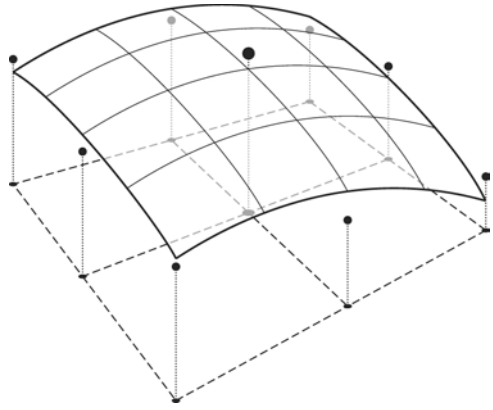

Figure 4.6 *Bi-quadratic surface patch fitted to a* 3×3 *window*

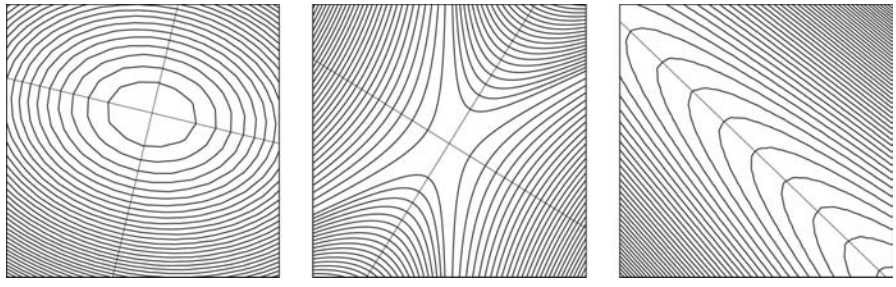

Figure 4.7 *Contours and semi-axes of bi-quadratic surfaces forming conic sections. From left to right: (a) elliptic; (b) hyperbolic; and (c) parabolic conic sections*

The names of the conic section types indicate the form of the isolines of each of the three surfaces. Furthermore, the conic section types correspond to the surface-specific features identified by Fowler and Little (1979): elliptic surfaces represent pits or peaks, hyperbolic surfaces are passes, and parabolic surfaces correspond to channels and ridges. The second derivatives of the surface distinguish between the possible convex and concave forms.

However, if a bi-quadratic surface defined over an $n \times n$ window is classified as elliptic or hyperbolic, the centre point of the window is not necessarily a pit or a peak. Only if the centre of the conic section is sufficiently close to the window centre, this inference is justified. Otherwise, the DEM point is a valley, a ridge, or a plane.

In order to decide whether the two centres are "sufficiently close", first, a region of interest around the centre of the $n \times n$ window is defined, and, second, the semi-axes of the conic sections are calculated (Wood, 1998). The *region of interest* is a circle around the centre of the window with radius r. By altering the radius r, the results of the extraction process can be influenced by effectively controlling the slope tolerance described above. It is useful to set r in accordance with the size n of the square $n \times n$ DEM window (e.g. $r = \sqrt{0.5n} \cdot c$, where c is the DEM cell size). The *semi-axes* of the conic section are calculated from the coefficients of equation (4.4) (Wood, 1998).

The number of intersections between the circular area of interest and the semi-axes determines the feature type of the point analysed

- both semi-axes intersect the area of interest
  - and the conic section is elliptic
    → the point is a pit or a peak
  - and the conic section is hyperbolic
    → the point is a pass
- one semi-axis intersects the area of interest
  → the point is a valley or a ridge
- no semi-axis intersects the area of interest
  → the point is a plane.

As with the bilinear patches described previously, this method also allows sub-pixel routing of linear features over the surface.

4.4.4 Deriving the network

The surface network is derived from a raster-based DEM in three sequential steps:

1. Identify pits, peaks, and passes with the method explained above.
2. Starting at the passes, trace two steepest paths of descent and two steepest paths of ascent as follows:
   - If no semi-axes intersects the region of interest, follow the aspect direction up or down, respectively;

- If one semi-axis intersects the region of interest, move parallel to the semi-axis;
- If two semi-axis intersect the region of interest, go to Step 3.

3. End the tracing when a pit or a peak is reached, or when the line hits the extent of the surface model. In the latter case, and whenever the critical line does not end at an existing critical point, insert a pit or a peak at this location.

4.4.5 Post-processing

Specific topographic configurations can lead to a small number of topological inconsistencies that need to be identified and corrected.

Topological consistency of the surface network is impaired by the finite extent of the terrain model. As has been stated above, consistency is ensured if the analysed terrain model is bound by a single contour. Therefore, the exterior of the model is interpreted as being either lower or higher than the rest of the model. As a result, the surface network is complemented by a universal pit or peak. In the case of a universal pit, all pits that were inserted where a valley reached the model's edge are marked as (topologically) belonging to the universal pit.

Point features that are co-located within the same flat region (e.g. plateaux and lakes) must be dealt with as a post-process. To restore topological consistency and, notably, to increase correspondence of the extracted information with the real situation, the separated pits and peaks are topologically merged. Matching pits, for instance, are easily identified because they occupy the same closed topological region of the network graph.

The method for extracting critical points explained above will only find pits, peaks, and passes if they are expressed in the terrain model by a respective topographic form. If, for instance, a pass is located in a large, nearly (or entirely) planar area, it will not be identified because the curvature values are so small that the location is classified as plane. In this particular case, two pits and two peaks are not connected (and separated) by a pass and the set of corresponding critical lines. If such configurations can be identified, the missing element can be inserted at a suitable location, and the topology can be (locally) completed.

As has been discussed already, paths of steepest descent and ascent may intersect in specific morphological configurations. It is justified to interpret such intersections as passes. (Passes themselves may be comprehended as intersection of valleys and ridges.) However, if a pass is inserted at the location of an intersection, the network topology is impaired. Thus, a pit needs to be inserted into the lower part of the intersected valley, and a peak inserted into the upper part of the ridge (Wood and Rana, 2000).

4.5 EXAMPLES

The presented methods have been applied to various synthetic and real surfaces of different characteristics and scales. The results of the extraction method are visually inspected.

The first of the two surfaces shown here is a synthetic surface generated by superimposing several sinus functions:

$$z = -\frac{5}{2}\sin\left(\frac{12\pi x}{200}\right)\sin\left(\frac{8\pi y}{200}\right) + r\sin\left(\frac{2\pi x}{200}\right)\sin\left(\frac{3\pi y}{200}\right) + \frac{10}{3}\sin\left(\sin\left(\frac{2\pi x}{200}\right)\right) + \frac{x}{10} + \frac{y}{20} \tag{4.5}$$

Values z are calculated for the range $0 \leqslant x \leqslant 200$, $0 \leqslant y \leqslant 200$. The cell size is 1 in both the x- and y-direction. Figure 4.8 shows a hill-shaded image of the surface and the derived network. The symbology of Figure 4.1 is used to draw the network elements.

Both surface networks are consistent in the sense that they are connected (i.e. there are no disconnected sub-networks), and that all connections of critical points through critical lines are valid. All critical points are extracted, as far as visual inspection can determine, and for each critical point, an according area object can be discerned. At few locations, the bilinear approach produces short chains of passes and pits or passes and peaks, respectively. Some of the critical lines generated by the bi-quadratic are somewhat jagged because of discontinuities between adjacent surface patches.

The individual elements of the two surface networks correspond well to each other; although the geometric locations may differ, for most elements of one surface network, there exists a matching element in the other. There are a few exceptions, for example, in the upper right part of the area of interest where a valley and a ridge line run very close to each other. In the surface network generated by the bilinear approach, there are two ridge lines to the left of one valley line, whereas in the bi-quadratic network, there is only one ridge line running on the right of the valley line.

The bilinear approach produces a somewhat larger number of critical points, namely of passes. This effect is caused, first, by the detection of passes along the model's edge that are not considered by the bi-quadratic approach, and second, by the detection of spurious passes along crest and in valleys. As a result, the surface network extracted by the bilinear approach is more dense and more complex.

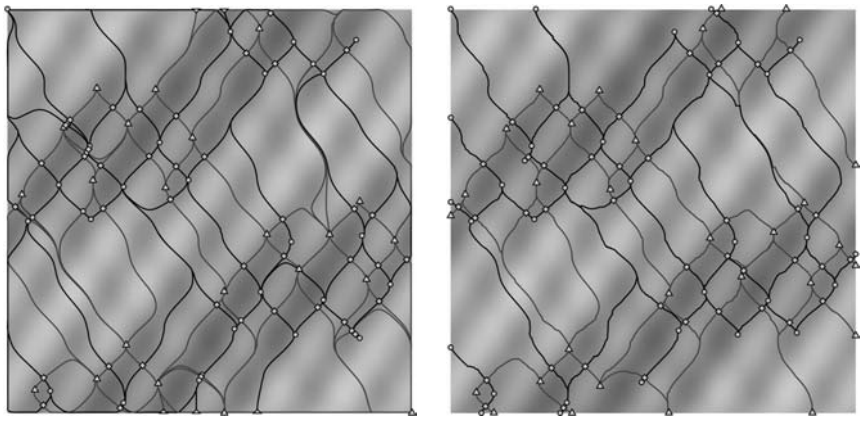

Figure 4.8 *Surface network derived from the synthetic surface of equation (4.4): (a) result of the bilinear approach; and (b) result of the bi-quadratic approach*

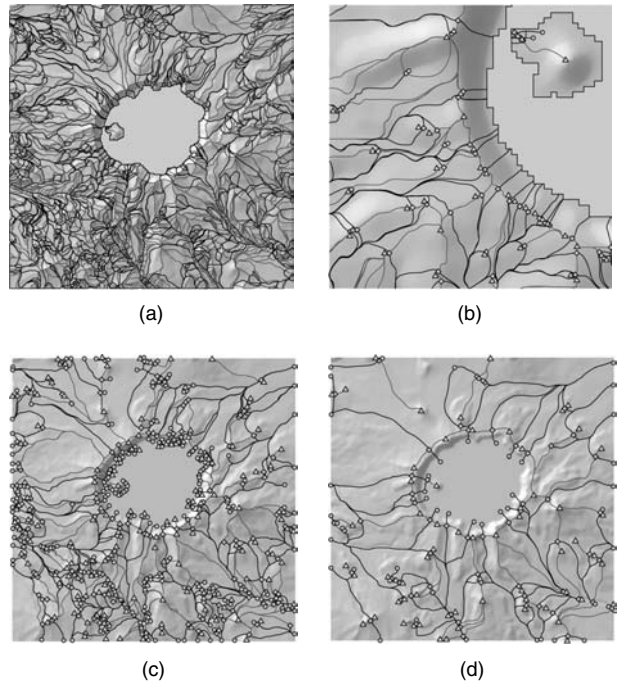

Figure 4.9 *Surface network derived from the Crater lake DEM for varying scales. (a) Result of the bilinear approach (to preserve readability, critical points are not drawn.); (b) enlarged cut out of (a); (c) and (d) results of the bi-quadratic approach with bi-quadratic surface patches specified from* 5×5 *and* 9×9 *windows, respectively*

Figure 4.9 shows a DEM of the Crater Lake area (Oregon, USA) produced by the USGS (URL #1, Gesch et al., 2002). For the presented study, the DEM has been resampled to a cell size of 100 m. The DEM consists of 316 columns and 301 rows and covers an extent of 31.5×30 km. The surface networks have been extracted from this test data set in order to illustrate the possibility offered by the bi-quadratic approach to restrict the analysis to a specific level of scale. The bi-quadratic surface patches have been specified from square windows of 5×5 and 9×9 cells, respectively. As expected, the number of features diminishes and only larger scale features persist.

However, close examination reveals that the surface network is not consistent everywhere. First, not all peaks are separated from each other through valleys. As mentioned above, this indicates that such peak pairs belong to the same superior feature and can be merged topologically. The analogous observation is made for pits. Second, the surface network may consist of a number of disconnected sub-networks. Third, some spurious features are extracted, although they are considerably less frequent than in the surface network extracted with the bilinear approach.

4.6 COMPARISON AND CONCLUSIONS

There is no single best approach to extracting surface networks from DEM data. Although the problem seems well defined in terms of differential algebra, it is clearly

affected by the discrete nature of the data, that is, by the fact that only limited information about the surface is available. Previous research (e.g. Takahashi et al., 1995) has shown that the raster data alone are not sufficient for consistent surface network construction. Naturally, there exist several methods for surface specification, but each method has specific properties affecting the results of the surface network extraction as well as the extraction process itself. Thus, the choice of an interpolation method greatly affects the extracted surface network, as this article clearly illustrates.

The following list summarises the most relevant and apparent differences of the two approaches discussed:

- The bi-quadratic approach needs a number of parameters to be set (scale and morphometric tolerances), while the bilinear approach is deterministic (except for the handling of horizontal areas).
- The bi-quadratic approach allows specifying the scale of analysis *a priori*, while the bilinear approach is limited to the scale induced by the DEM resolution. Introducing a hierarchical structure to the (topological) surface network introduces (an aspect of) scale *a posteriori*.
- The bilinear approach allows and requires dealing with horizontal areas as such. The horizontal areas need to be delineated, typified, interpreted, and accordingly dealt with during network derivation. The bi-quadratic approach makes it unnecessary to deal with horizontal areas. As a consequence, however, critical points belonging to the same horizontal area, for example, to the same lake, need to be merged as part of post-processing.
- Under the assumption that for all continuous surfaces, there exist topologically consistent surface networks, the bilinear surface – being a continuous surface – grants surface network consistency. It remains to be confirmed that the presented approach is able to extract the network accordingly for all DEM data. The bi-quadratic approach, on the other hand, does not start with the specification of a globally continuous surface. (Instead, it composes the global surface by means of discontinuous bi-quadratic surface patches.) Without some post-processing, there is no guarantee of a network consistent with Euler's formula.
- The terrain surface composed of bilinear surface patches is characterised by numerous spurious point features (pits, peaks, and passes). Since they are all extracted and inserted into the surface network, carrying out a post-processing step is advised in order to separate spurious from significant features. This can be achieved by building a hydrological hierarchy from the surface network features (Schneider, 2003). Alternatively, different interpolation schemes (e.g. cubic interpolation over 4×4 windows) may yield globally continuous surfaces exhibiting significantly fewer spurious features. Ongoing research investigates the two approaches.
- The bi-quadratic approach limits the location of critical points to the raster points. This is a benefit if the surface network data is later integrated with other raster data. In terms of expressiveness of the model, it may be considered a drawback. In the bilinear case, the surface model inherently limits pits and peaks to raster points. Passes, however, occur at raster points as well as within cells.
- With the bi-quadratic approach, all passes are 4-valent. This rule does not apply in the bilinear approach. If a pass is represented by a horizontal area, it may be more

than 4-valent. If the pass is located at the edge of the area of interest, it is 3-valent. This observation increases algorithm complexity to some degree.

- In the bi-quadratic approach, the critical points need to have a morphological expression, that is, their occurrence must be accompanied by a topographic form that can be recognised by the extraction algorithm. In other words, creating the network based only on morphometric information is not sufficient (Wood and Rana, 2000). The bilinear approach finds all critical points independent of the surface morphology.

The authors claim that it is not possible to rank the methods presented. However, different users with different applications will develop their own preferences on the basis of the distinct properties of the presented approaches. This article may serve as a guide to recognise the relevant and mandatory properties.

5

Contour Trees and Small Seed Sets for Isosurface Generation

Marc van Kreveld, René van Oostrum, Chandrajit Bajaj, Valerio Pascucci and Dan Schikore

5.1 INTRODUCTION

One of the functionalities of a GIS is to display data by generating tables, charts, and maps, either on paper or on a computer screen. Several kinds of maps are available for displaying the different types of data. Choropleth maps are used to display categorial data, such as different types of vegetation. Network maps, such as railroad maps, show connections (railways) between geographic objects (stations); the regions on a network map are meaningless. Finally, isoline maps are a very effective means of displaying scalar data defined over the plane. Such data can be visualised after interpolation by showing one or more contours: the sets of points having a specified value. For example, scalar data over the plane is used to model elevation in the landscape, and a contour is just an isoline of elevation. Contours can be used for visualising scalar data defined over the three-dimensional space as well. In that case, the contours are two-dimensional isosurfaces. For instance, in atmospheric pressure modelling, a contour is a surface in the atmosphere where the air pressure is constant – an isobar. The use of isolines or isosurfaces for displaying scalar data is not limited to the field of GIS. In medical imaging, for example, isosurfaces are used to show reconstructed data from scans of the brain or parts of the body. The scalar data can be seen as a sample of some real-valued function, which is called a terrain or elevation model in GIS, and a scalar field in imaging.

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

A real-valued function over a two- or three-dimensional domain can be represented using a two- or three-dimensional mesh, which can be regular (all cells have the same size and shape) or irregular. A terrain (mountain landscape) in GIS is commonly represented by a regular square grid or an irregular triangulation. The elements of the grid, or vertices of the triangulation, have a scalar function value associated to them. The function value of non-vertex points in the two-dimensional mesh can be obtained by interpolation. An easy form of interpolation for irregular triangulations is linear interpolation over each triangle. The resulting model is known as the TIN model for terrains (Triangulated Irregular Network) in GIS. In computational geometry, it is known as a polyhedral terrain. More on interpolation of spatial data and references to the literature can be found in the book by Watson (1992).

One can expect that the combinatorial complexity of the contours with a single function value in a mesh with n elements is roughly proportional to $\sqrt{n}$ in the two-dimensional case and to $n^{2/3}$ in the three-dimensional case (Livnat et al., 1996). Therefore, it is worthwhile to have a search structure to find the mesh elements through which the contours pass. This will be more efficient than retrieving the contours of a single function value by inspecting all mesh elements.

There are basically two approaches to find the contours more efficiently. Firstly, one could store the two-dimensional or three-dimensional domain of the mesh in a hierarchical structure and associate the minimum and maximum occurring scalar values at the subdomains to prune the search. For example, octrees have been used this way for regular three-dimensional meshes (Wilhelms and van Gelder, 1992).

The second approach is to store the *scalar range*, also called *span*, of each of the mesh elements in a search structure. Kd-trees (Livnat et al., 1996), segment trees (Bajaj et al., 1996), and interval trees (Cignoni et al., 1996, van Kreveld, 1996) have been suggested as the search structure, leading to a contour retrieval time of $O(\sqrt{n} + k)$ or $O(\log n + k)$, where n is the number of mesh elements and k is the size of the output. A problem with this approach is that the search structure can be a serious storage overhead, even though an interval tree needs only linear storage. It is possible to reduce the storage requirements of the search structures by observing that a whole contour can be traced directly in the mesh if one mesh element through which the contour passes is known. Such a starting element of the mesh is also called a *seed*. Instead of storing the scalar range of all mesh elements, we need only store the scalar range of the seeds as intervals in the tree, and a pointer into the mesh, or an index, if a (two- or three-dimensional) array is used. Of course, the seed set must be such that every possible contour of the function passes through at least one seed. Otherwise, contours could be missed. There are a few papers that take this approach (Bajaj et al., 1996, Itoh and Koyamada, 1995, van Kreveld, 1996). The tracing algorithms to extract a contour from a given seed have been developed before, and they require time linear in the size of the output (Artzy et al., 1981, Howie and Blake, 1994, Itoh and Koyamada, 1995).

The objective of this chapter is to present new methods for seed set computation. To construct a seed set of small size, we use a variation of the *contour tree*, a tree that captures the contour topology of the function represented by the mesh. It has been used before in image processing and GIS research (Freeman and Morse, 1967, Gold and Cormack, 1986, Kweon and Kanade, 1994, Sircar and Cerbrian, 1986, Takahashi et al., 1995). Another name in use is the *topographic change tree*, and it is related

to the *Reeb graph* used in Morse Theory (Reeb, 1946, Shinagawa and Kunii, 1991, Shinagawa et al., 1991, Takahashi et al., 1995). It can be computed in $O(n \log n)$ time for functions over a two-dimensional domain (de Berg and van Kreveld, 1997).

This chapter includes the following results:

- We present a new, simple algorithm that constructs the contour tree. For two-dimensional meshes with n elements, it runs in $O(n \log n)$ time like a previous algorithm (de Berg and van Kreveld, 1997), but the new method is much simpler and needs less additional storage. For meshes with n faces in d-dimensional space, it runs in $O(n^2)$ time. In typical cases, less than linear temporary storage is needed during the construction, which is important in practice. Also, the higher-dimensional algorithm requires subquadratic time in typical cases.
- We show that the contour tree is the appropriate structure to use when selecting seed sets. We give an $O(n^2 \log n)$ time algorithm for seed sets of minimum size by using minimum cost flow in a directed acyclic graph (Ahuja et al., 1993).
- In practice, one would like a close-to-linear-time algorithm when computing seed sets. We sketch a simple algorithm that requires $O(n \log^2 n)$ time and linear storage after construction of the contour tree, and gives seed sets of small size. The approximation algorithm has been implemented, and we supply test results of various kinds.

Previous methods to find seed sets of small size did not give any guarantee on their size (Bajaj et al., 1996, Itoh and Koyamada, 1995, van Kreveld, 1996). After the results of this chapter were published (van Kreveld et al., 1997), Tarasov and Vyalyi (1998) extended our contour tree construction algorithm and obtained an $O(n \log n)$ time algorithm for the three-dimensional case. Their algorithm consists of a pre-processing step with two sweeps, after which our algorithm is used. Later, Carr et al. (2003) gave a contour tree algorithm that is efficient in all dimensions. Its implementation and experiments were given by Kettner and Snoeyink (2001).

5.2 PRELIMINARIES ON SCALAR FUNCTIONS AND THE CONTOUR TREE

In this section, we provide background and definitions of terms used in the following sections. On a continuous function $\mathcal{F}$ from d-space to the reals, the *criticalities* can be identified. These are the local maxima, the local minima, and the saddles (or passes). If we consider all contours of a specified function value, we have a collection of lower-dimensional regions in d-space called a *level set* (typically, $(d-1)$-dimensional surfaces of arbitrary topology). If we let the function value take on the values from $+\infty$ to $-\infty$, a number of things may happen to the contours. Contours deform continuously, with changes in topology only when a criticality is met (i.e. its function value is passed). A new contour starts to form whenever the function value is equivalent to a locally maximal value of $\mathcal{F}$. An existing contour disappears whenever the function value is equivalent to a locally minimal value.

At saddle points, various different things can happen. It may be that two (or more) contours adjoin, or one contour splits into two (or more) components, or that a contour

gets a different topological structure (e.g. from a sphere to a torus in three dimensions). The changes that can occur have been documented in texts on Morse theory and differential topology (Hirsch, 1976, Milnor, 1963). They can be described by a structure called the contour tree, which we describe below.

As an example, consider a function modelled by a two-dimensional triangular mesh with linear interpolation and consider how the contour tree relates to such meshes (see Figure 5.1). For simplicity, we assume that all vertices have a different function value. If we draw the contours of all vertices of the mesh, then we get a subdivision of the two-dimensional domain into regions. All saddle points, local minima, and maxima must be vertices of the mesh in our setting. The contour through a local minimum or maximum is simply the point itself. One can show that every region between contours is bounded by exactly two contours (de Berg and van Kreveld, 1997).

We let every contour in this subdivision correspond to a node in a graph, and two nodes are connected if there is a region bounded by their corresponding contours. This graph is a tree and is called the contour tree (de Berg and van Kreveld, 1997, van Kreveld, 1996). All nodes in the tree have degree 1 (corresponding to local extrema), degree 2 (normal vertices), or at least degree 3 (saddles). In other words, every contour of a saddle vertex splits the domain into at least three regions. For each vertex in the triangulation, one can test locally whether it is a saddle. This is the case if and only if it has neighbouring vertices around it that are higher, lower, higher, and lower, in cyclic order around it. If one would take the approach outlined above to construct the contour tree, $\Omega(n^2)$ time may be necessary in the worst case, because the total combinatorial complexity of all contours through saddles may be quadratic. An $O(n \log n)$ time divide-and-conquer algorithm exists, however (de Berg and van Kreveld, 1997).

In a general framework, we define the contour tree with only few assumptions on the type of mesh, form of interpolation, and dimension of the space over which function $\mathcal{F}$ is defined. The input data is assumed to be

a mesh M of size n embedded in $\mathbb{R}^d$;
a continuous real-valued function $\mathcal{F}$ defined over each cell of M.

A *contour* is defined to be a maximal connected piece of $\mathbb{R}^d$ where the function value is the same. Usually, a contour is a $(d-1)$-dimensional hypersurface, but it

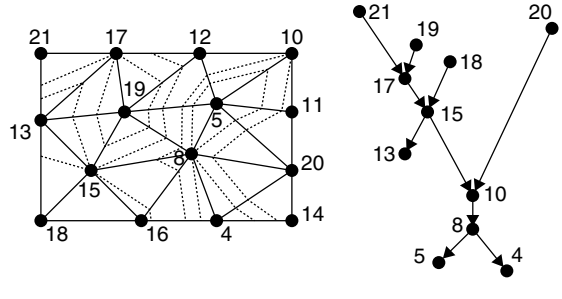

Figure 5.1 *Two-dimensional triangular mesh with the contours of the saddles, and the contour tree*

can also be lower dimensional or d-dimensional. We define the contour tree $\mathcal{T}$ as follows:

Take each contour that contains a criticality.
These contours correspond to the *supernodes* of $\mathcal{T}$ (the tree will be extended later with additional nodes, hence we use the term supernodes here). Each supernode is labelled with the function value of its contour.
For each region bounded by two contours, we add a superarc between the corresponding supernodes in $\mathcal{T}$.

The contour tree is well defined, because each region is bounded by two and only two contours that correspond to supernodes. One can show that the contour tree is indeed a tree: the proof for the two-dimensional case given by de Berg and van Kreveld (1997) can easily be extended to d dimensions.

For two-dimensional meshes, all criticalities correspond to supernodes of degree 1, or degree 3 or higher. For higher-dimensional meshes, there are also criticalities that correspond to a supernode of degree 2. This occurs, for instance, in three dimensions when the genus of a surface changes, for instance, when the surface of a ball changes topologically to a torus. In d-dimensional space (for $d > 2$), a saddle point p is a point such that for any sufficiently small hypersphere around p, the contour of p's value intersects the surface of the hypersphere in at least two separate connected components.

Superarcs are directed from higher scalar values to lower scalar values. Thus, supernodes corresponding to the local maxima are the sources and the supernodes corresponding to the local minima are the sinks.

To be able to compute the contour tree, we make the following assumptions on the mesh M:

Inside any face of any dimension of M, all criticalities and their function values can be determined.
Inside any face of any dimension of M, the range (min, max) of the function values taken inside the face can be determined.
Inside any face of any dimension of M, the (piece of) contour of any value in that face can be determined.

We assume that in facets and edges of two-dimensional meshes, the items listed above can be computed in $O(1)$ time. For vertices, we assume that the first item takes time linear in its degree. Similarly, in three-dimensional meshes we assume that these items take $O(1)$ to compute in cells and on facets, and time linear in the degree on edges and at vertices.

5.3 CONTOUR TREE ALGORITHMS

In this section, we assume, for ease of presentation, that the mesh M is a simplicial decomposition with n cells, and that linear interpolation is used. As a consequence, all critical points are vertices of the mesh M. Instead of computing the contour tree as defined in the previous section, we compute an extension that includes nodes for the contours of all vertices of M including the non-critical ones. So supernodes

correspond to contours of critical vertices and regular nodes correspond to contours of other vertices. Each superarc is now a sequence of arcs and nodes, starting and ending at a supernode. The algorithm we will describe next can easily be adapted to determine the contour tree with only the supernodes. But we will need this extended contour tree for seed selection in the next section. From now on, we refer to the contour tree with nodes for the contours of all vertices as the contour tree $\mathcal{T}$.

The supernodes of $\mathcal{T}$ that have in-degree 1 and out-degree greater than 1 are called *bifurcations*, and the supernodes with in-degree greater than 1 and out-degree 1 are called *junctions*. All normal nodes have in-degree 1 and out-degree 1. We will assume that all bifurcations and junctions have degree exactly 3, that is, out-degree 2 for bifurcations and in-degree 2 for junctions. This assumption can be removed; one can represent all supernodes with higher degrees as clusters of supernodes with degree 3. For example, a supernode with in-degree 2 and out-degree 2 can be treated as a junction and a bifurcation, with a directed arc from the junction to the bifurcation. The assumption that all junctions and bifurcations have degree 3 facilitates the following descriptions considerably.

5.3.1 The general approach

To construct the contour tree $\mathcal{T}$ for a given mesh in d-space, we let the function value take on the values from $+\infty$ to $-\infty$ and we keep track of the contours for these values. In other words, we sweep the scalar value. For two-dimensional meshes, one can image sweeping a polyhedral terrain embedded in a three-dimensional space and moving downward a horizontal plane. The sweep stops at certain event points: the vertices of the mesh. During the sweep, we keep track of the contours in the mesh at the value of the sweep function, and the set of cells of the mesh that cross these contours. The cells that contain a point with value equivalent to the present function value are called *active*. The tree $\mathcal{T}$ under construction during the sweep will be growing at the bottom at several places simultaneously (see Figure 5.2).

Each part of $\mathcal{T}$ that is still growing corresponds to a unique contour at the current sweep value. We group the cells into contours by storing a pointer at each active cell in

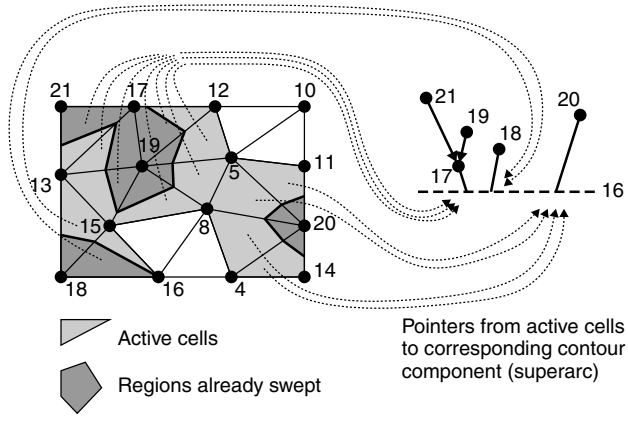

Figure 5.2 *Situation of the sweep of a two-dimensional mesh when the function value is 16*

the mesh to the corresponding superarc in $\mathcal{T}$. The contours can only change structurally at the event points, and the possible changes are the following:

- At a local maximum of the mesh (more precisely, of the function), a new contour appears. This is reflected in $\mathcal{T}$ by creating a new supernode and a new arc incident to it. This arc is also the start of a new superarc, which will be represented. Each cell incident to the maximum becomes active, and we set their pointer to the new superarc of $\mathcal{T}$. At this stage of the algorithm, the new superarc has no lower node attached to it yet.
- At a local minimum of the mesh, a contour disappears; a new supernode of $\mathcal{T}$ is created, and the arc corresponding to the disappearing contour at the current value of the sweep is attached to the new supernode. It is also the end of a superarc. The cells of the mesh incident to the local minimum will no longer be active.
- At a non-critical vertex of the mesh, a new node of $\mathcal{T}$ is created, the arc corresponding to the contour containing the vertex is made incident to the node, and a new arc incident to the node is created. There is no new superarc. Some cells incident to the vertex stop being active, while others become active. The pointers of the latter cells are set to the current superarc of the contour. For the cells that remain active, nothing changes: their pointer keeps pointing to the same superarc.
- At a saddle of the mesh, there is some change in topology in the collection of contours. It may be that two or more contours merge into one, one contour splits into two or more, or one contour changes its topological structure. A combination of these is also possible in general. The first thing to do is to determine what type of saddle we are dealing with. This can be decided by traversing the whole contour on which the saddle lies.
- If two contours merge, a new supernode (junction) is created in $\mathcal{T}$ for the saddle, and the superarcs corresponding to the two merging contours are made incident to this supernode. Furthermore, a new arc and superarc are created for the contour that results from the merge. The new arc is attached to the new supernode. All cells that are active in the contour after the merge set their pointer to the new superarc in $\mathcal{T}$. If a contour splits, then similar actions are taken.
- If the saddle is because of a change in topology of one single contour (i.e. an increase or decrease of its genus by one), a new supernode is made for one existing superarc, and a new arc and superarc are created in $\mathcal{T}$. All active cells of the contour set their pointers to the new superarc.

For the sweep algorithm, we need an event queue and a status structure. The event queue is implemented with a standard heap structure, so insertions and extractions take logarithmic time per operation. The status structure is implicitly present in the mesh with the additional pointers from the cells to the superarcs in the contour tree.

Theorem 1 *Let M be a mesh in d-space with n faces in total, representing a continuous, piecewise linear function over the mesh elements. The contour tree of M can be constructed in $O(n^2)$ time and $O(n)$ storage.*

Proof. The algorithm clearly takes time $O(n \log n)$ for all heap operations. If the mesh is given in an adjacency structure, then the traversal of any contour takes time linear

in the combinatorial complexity of the contour. Any saddle of the function is a vertex, and any contour can pass through any mesh cell only once. Therefore, the total time for all traversals is $O(n^2)$ in the worst case, and the same amount of time is needed for setting the pointers of the active cells.

The quadratic running time shown above is somewhat pessimistic, since it applies only when there is a linear number of saddles for which the contour through them has linear complexity. We can also state that the running time is $O(n \log n + \sum_{i=1}^{m} |C_i|)$, where the m saddles lie on contours $C_1, \ldots, C_m$ with complexities $|C_1|, \ldots, |C_m|$.

We claimed that the additional storage of the algorithm could be made sublinear in practice. With additional storage we mean the storage besides the mesh (input) and the contour tree (output). We will show that $O([\text{no. maxima}] + \max_{1 \leq i \leq m} |C_i|)$ extra storage suffices. We must reduce the storage requirements of both the event queue and the status structure.

Regarding the event queue, we initialise it with the values of the local maxima only. During the sweep, we will insert all vertices incident to active cells as soon as the cell becomes active. This guarantees that the event queue uses no more additional storage than claimed above. Considering the status structure, we cannot afford using additional pointers with every cell of the mesh to superarcs any more. However, we need these pointers only when the cell is active. We will make a copy of the active part of the mesh, and with the cells in this copy, we may use additional pointers to superarcs in $\mathcal{T}$ and to the corresponding cells in the original mesh. When a cell becomes inactive again, we delete it from the copy. With these modifications, the additional storage required is linear in the maximum number of active cells and the number of local maxima. This can be linear in theory, but will be sublinear in typical cases. The asymptotic running time of the algorithm is not influenced by these changes.

5.3.2 The two-dimensional case

In the two-dimensional case, the time bound can be improved to $O(n \log n)$ time in the worst case by a few simple adaptations. First, we give a crucial observation: for two-dimensional meshes representing continuous functions, all saddles correspond to nodes of degree of at least 3 in $\mathcal{T}$. Hence, at any saddle two or more contours merge, or one contour splits into at least two contours, or both. This is different from the situation in three dimensions, where a saddle can cause a change in genus of a contour, without causing a change in connectedness. The main idea is to implement a merge in time linear in the complexity of the *smaller* of the two contours, and similarly, to implement a split in time linear in the complexity of the *smaller resulting contour*.

In the structure, each active cell has a pointer to a *name* of a contour, and the name has a pointer to the corresponding superarc in $\mathcal{T}$. We consider the active cells and names as a union-find-like structure (Cormen et al., 1990) that allows the following operations:

Merge: given two contours about to merge, combine them into a single one by renaming the active cells to have a common name.

Split: given one contour about to split, split it into two separate contours by renaming the active cells for one of the contours to be created to a new name.

Find: given one active cell, report the name of the contour it is in.

Like in the simplest union-find structure, a *Find* takes $O(1)$ time since we have a pointer to the name explicitly. A *Merge* is best implemented by changing the name of the cells in the smaller contour to the name of the larger contour. Let us say that contours C_i and C_j are about to merge. Determining which of them is the smallest takes $O(\min(|C_i|, |C_j|))$ time if we traverse both contours simultaneously. We alternatingly take one "step" in C_i and one "step" in C_j. After a number of steps, twice the combinatorial complexity of the smaller contour, we have traversed the whole smaller contour. This technique is sometimes called *tandem search*. To rename for a *Merge*, we traverse this smaller contour again and rename the cells in it, again taking $O(\min(|C_i|, |C_j|))$ time.

The *Split* operation is analogous: if a contour C_k splits into C_i and C_j, the name of C_k is preserved for the larger of C_i and C_j, and by tandem search starting at the saddle in two opposite directions we find out which of C_i and C_j will be the smaller one. This will take $O(\min(|C_i|, |C_j|))$ time. Note that we cannot keep track of the size in an integer for each contour instead of doing tandem search, because then a *Split* cannot be supported efficiently.

Theorem 2 *Let M be a two-dimensional mesh with n faces in total, representing a continuous, piecewise linear scalar function. The contour tree of this function can be computed in $O(n \log n)$ time and linear storage.*

Proof. We can distinguish the following operations and their costs involved:

- Determining for each vertex what type it is (min, max, saddle, normal) takes $O(n)$ in total.
- The operations on the event queue take $O(n \log n)$ in total.
- Creating the nodes and arcs of $\mathcal{T}$, and setting the incidence relationships takes $O(n)$ time in total.
- When a cell becomes active, the name of the contour it belongs to is stored with it; this can be done in $O(1)$ time, and since there are $O(n)$ such events, it takes $O(n)$ time in total.
- At the saddles of the mesh, contours merge or split. Updating the names of the contours stored with the cells takes $O(\min(|C_i|, |C_j|))$ time, where C_i and C_j are the contours merging into one, or resulting from a split, respectively. It remains to show that summing these costs over all saddles yields a total of $O(n \log n)$ time.

We prove the bound on the summed cost for renaming by transforming $\mathcal{T}$ in two steps into another tree $\mathcal{T}'$ for which the construction is at least as time-expensive as for $\mathcal{T}$, and showing that the cost at the saddles in $\mathcal{T}'$ are $O(n \log n)$ in total.

Consider the cells of the mesh to correspond to additional *segments* in $\mathcal{T}$ as follows. Any cell becomes active when the sweep plane reaches its highest vertex, and stops being active when the sweep plane reaches its lowest vertex. These vertices correspond to nodes in $\mathcal{T}$, and the cell is represented by a segment connecting these nodes. Note that any segment connects two nodes, one of which is an ancestor of the other. A segment can be seen as a shortcut of a directed path in $\mathcal{T}$, where it may pass over several nodes and supernodes.

The number of cells involved in a merge or split at a saddle is equivalent to the number of segments that pass over the corresponding supernode v in $\mathcal{T}$, plus the number

of segments that start or end at v. The set of segments passing v can be subdivided into two subsets as follows: segments corresponding to cells that are intersected by the same contour before the merge or after the split at the saddle corresponding to v are in same subset. The size of the smallest subset of segments passing v determines the costs for processing the saddle since we do tandem search.

The first transformation step is to *stretch* all segments (see Figure 5.3); we simply assume that a segment starts at some source node that is an ancestor of the original start node, and ends at a sink that is a descendant of the original end node. It is easy to see that the number of segments passing any saddle can only increase by the stretch.

The second transformation step is to repeatedly *swap* superarcs, until no supernode arising from a split (bifurcation) is an ancestor of a supernode arising from a merge (junction). Swapping a superarc s from a bifurcation v to a junction u is defined as follows (see Figure 5.4): let $s' \neq s$ be the superarc that has u as its lower supernode, and let $s'' \neq s$ be the superarc that has v as its upper supernode. The number of segments passing the superarcs s', s, and s'' is denoted by a, b, and c, respectively, as is illustrated in Figure 5.4.

These numbers are well defined, since after stretching, any segment passes a superarc either completely or not at all. Now shift s' upward along s, such that v becomes its new lower supernode, and shift s'' downward along s, such that u becomes its new upper supernode. Note that all edges passing s' and all edges passing s'' before the swap now also pass s.

Before the swap, the time spent in the merge at u and the split at v, is $O(\min(a, b) + \min(b, c))$ where a, b, c denote the number of segments passing these superarcs. After the swap, this becomes $O(\min(a, b + c) + \min(a + b, c))$, which is at least as much. No segment ends, because all of them were stretched.

It can easily be verified that a tree $\mathcal{T}'$, with no bifurcation as an ancestor of a junction, can be derived from any tree $\mathcal{T}$ by swaps of this type only. Any segment in

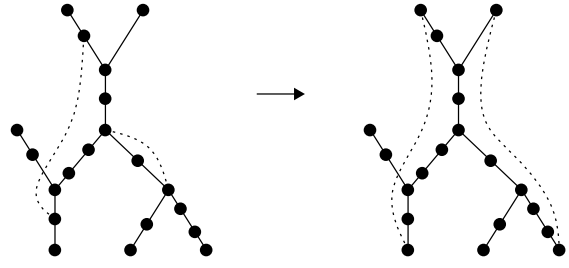

Figure 5.3 *Stretching two segments (dotted) in $\mathcal{T}$*

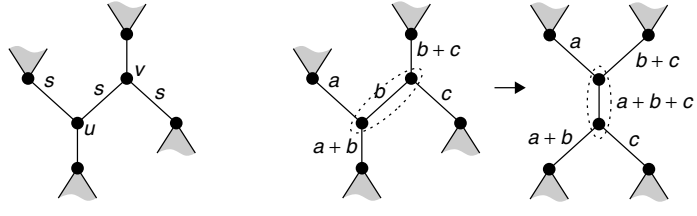

Figure 5.4 *Swapping a superarc*

$\mathcal{T}'$ first passes a sequence of at most $O(n)$ junctions, and then a sequence of at most $O(n)$ bifurcations.

We charge the costs of the merge and split operations to the segments that are in the smallest set just before a merge and just after a split. Now, every segment can pass $O(n)$ junctions and bifurcations, but no segment can be more than $O(\log n)$ times in the smaller set. Each time it is in the smaller set at a junction, it will be in a set of at least twice the size just after the junction. Similarly, each time it is in the smallest set just after a bifurcation, it has come from a set of at least twice the size just before the bifurcation. It follows that any segment is charged at most $O(\log n)$ times. Summing over the $O(n)$ segments, this results in a total of $O(n \log n)$ time for all renamings of cells. This argument is standard in the analysis of union-find structures, see, for instance, Cormen et al. (1990).

The $O(n \log n)$ time bounds for the contour tree construction are tight. An $\Omega(n \log n)$ lower bound construction can be found in (van Oostrum, 1999). The construction uses $\Omega(n)$ critical points, which is not typical for real-world data. Carr et al. (2003) provide a theoretically and practically better algorithm for all dimensions.

5.4 SEED SET SELECTION

A seed set is a subset of the cells of the mesh and serves as a collection of starting points from which contours can be traced, for instance, for visualisation. A seed set is *complete* if every possible contour passes through at least one seed. From now on, we understand seed sets to be complete. Since we assume linear interpolation over the cells, the function values occurring in one cell form exactly the range between the lowest and the highest-valued vertices. Any cell is represented as a *segment* between two nodes of the contour tree $\mathcal{T}$, as in the proof of Theorem 2. Segments can only connect two nodes of which one is an ancestor of the other. Like the arcs of $\mathcal{T}$, the segments are directed from the higher to the lower value. So, each segment is in fact a shortcut of a directed path in $\mathcal{T}$. We say that the segment *passes*, or *covers*, these arcs of $\mathcal{T}$. Let $\mathcal{G}$ denote the directed acyclic graph consisting of the contour tree extended with the segments of all mesh elements. The small seed set problem now is the following graph problem: find a small subset of the segments such that every arc of $\mathcal{T}$ is passed by some segment of the subset. Since the contour tree and the graph $\mathcal{G}$ have the same form regardless of the dimension of the mesh, the algorithms of this section give seed sets in any dimension.

In this section, we give two methods to obtain complete and small seed sets. The first gives a seed set of minimum size, but it requires $O(n^2 \log n)$ time for its computations. The second method requires $O(n \log^2 n)$ time and linear storage (given the contour tree and the segments), and gives a seed set that can be expected to be small, which is evidenced by test results.

5.4.1 Seed sets of minimum size in polynomial time

We can find a seed set of minimum size in polynomial time by reducing the seed set selection problem to a minimum cost flow problem. The flow network $\mathcal{G}'$ derived from $\mathcal{G}$ is defined as follows: we augment $\mathcal{G}$ with two additional nodes, a *source* σ and a *sink*

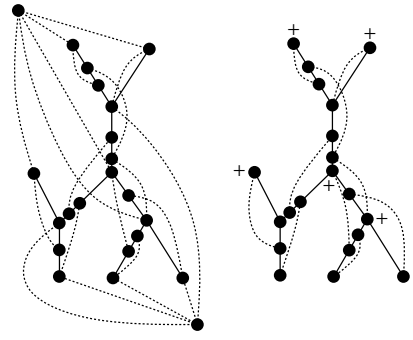

Figure 5.5 *Flow network $\mathcal{G}'$ derived from $\mathcal{G}$, and shorthand for $\mathcal{G}'$*

σ'. The source σ is connected to all maxima and bifurcations by additional segments, and the sink is connected to all minima and junctions with additional segments. This is illustrated in Figure 5.5, left. In the same figure (right) a shorthand for the same flow network has been drawn: for readability, σ and σ' have been omitted, and the additional segments incident to σ and σ' have been replaced by "+" and "−" signs respectively. From now on, we will use this shorthand notation in the figures.

Costs and capacities for the segments and arcs are assigned as follows: nodes in $\mathcal{G}$ are ordered by the height of the corresponding vertices in the mesh, and segments and arcs are considered to be directed: segments (dotted) go downward from higher to lower nodes, arcs (solid) go upward from lower to higher nodes. The source σ is considered to be the highest node, and σ' the lowest. Segments in $\mathcal{G}$ have capacity 1 and cost 1, and arcs have capacity ∞ and cost 0. The additional segments in $\mathcal{G}'$ incident to σ and σ' have capacity 1 and cost 0.

From graph theory we have the following lemma:

Lemma 1 *For any tree, the number of maxima plus the number of bifurcations equals the number of minima plus the number of junctions.*

Hence, the number of pluses in $\mathcal{G}$ balances the number of minuses. Let this number be f.

Consider the following two related problems, the *flow problem* (given the network $\mathcal{G}'$ as defined above and a value f, find a flow of size f from σ to σ'), and the *minimum cost flow* problem (find such a flow f with minimum cost). For both the problems, a solution consists of an assignment of flow for each segment and arc in $\mathcal{G}'$. For such a solution, let the *corresponding segment set* $\mathcal{S}$ be the set of segments in $\mathcal{G}$ that have a non-zero flow assigned to them (the additional segments in $\mathcal{G}'$ from σ to the maxima and bifurcations and from the minima and junctions to σ' are not in $\mathcal{S}$). Hence, the cost of an *integral solution*, where all flow values are integers, equals the number of segments in $\mathcal{S}$. It follows from the lemmas below that for any integral solution to the minimum cost flow problem on $\mathcal{G}'$, the corresponding segment set $\mathcal{S}$ is a minimum size seed set for $\mathcal{G}$. The proof is given by van Oostrum (1999).

Lemma 2 *For any integral solution to the flow problem on $\mathcal{G}'$, the corresponding segment set $\mathcal{S}$ is a seed set for $\mathcal{G}$.*

A seed set is *minimal* if the removal of any segment yields a set that is not a seed set. A *minimum* seed set is a seed set of smallest size.

Lemma 3 *For any minimal seed set S for $\mathcal{G}$, there is a solution to the flow problem on $\mathcal{G}'$ such that the corresponding segment set $\mathcal{S}$ for that solution equals S.*

The proof of the lemma above can also be found in (van Oostrum, 1999). Combining Lemmas 2 and 3 gives the following result:

Theorem 3 *The minimum seed set selection problem for $\mathcal{G}$ can be solved by applying a minimum cost flow algorithm to $\mathcal{G}'$ that gives an integral solution. Such a solution is guaranteed to exist, and the corresponding segment set for that solution is an optimal seed set for $\mathcal{G}$.*

The minimum cost flow problem can be solved with a successive shortest path algorithm (Ahuja et al., 1993). Starting with a zero flow, this algorithm determines at every step the shortest path π from σ to σ', where the length of an arc or segment is derived from its cost. The arc or segment with the lowest capacity c on this shortest path π determines the flow that is sent from σ to σ' along π. Then the *residual network* is calculated (costs and capacities along π are updated), and the algorithm iterates until the desired flow from σ to σ' is reached, or no additional flow can be sent from σ to σ' along any path.

In our case, c is always *1* and the algorithm terminates after f iterations. If we use Dijkstra's algorithm to find the shortest path in each iteration, the minimum cost flow algorithm runs in $O(n^2 \log n)$ time on our graph $\mathcal{G}'$, and uses $O(n)$ memory.

Theorem 4 *An optimal seed set for $\mathcal{G}$ can be found in $O(n^2 \log n)$ time, using $O(n)$ memory.*

5.4.2 Efficient computation of small seed sets

The roughly quadratic time requirements for computing optimal seed sets makes it rather time consuming in practical applications. We therefore developed an algorithm to compute a seed set that, after constructing the contour tree $\mathcal{T}$, uses linear storage and $O(n \log^2 n)$ time in any dimension. The seed sets resulting from this algorithm can be expected to be small, which is supported by test results. We will only sketch the algorithm here; details are given by van Oostrum (1999).

As before, we will describe the algorithm in the simplified situation that each critical vertex of the mesh is a minimum, a maximum, a junction, or a bifurcation. In the case of a junction or bifurcation, we assume that the degree is exactly three. These simplifying assumptions make it easier to explain the algorithm, but they can be removed as before.

Our algorithm is a simple greedy method that operates quite similar to the contour tree construction algorithm. We first construct the contour tree $\mathcal{T}$ as before. We store with each node of $\mathcal{T}$ two integers that will help determine fast whether any two nodes of $\mathcal{T}$ have an ancestor/descendant relation. The two integers are assigned as follows. Give $\mathcal{T}$ some fixed, left-to-right order of the children and parents of each supernode. Then perform a left-to-right topological sort to number all nodes once. Then perform a

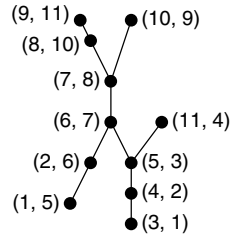

Figure 5.6 *The numbering of* $\mathcal{T}$

right-to-left topological sort to give each node a second number. The numbers are such that one node u is an ancestor of another node v if and only if the first number and the second number of u are smaller than the corresponding numbers of v (see Figure 5.6).

The pre-processing of the contour tree takes $O(n)$ time, and afterwards, we can determine in $O(1)$ time for any two nodes whether one is a descendant or ancestor of the other.

Next we add the segments, one for each cell of the mesh, to the contour tree $\mathcal{T}$ to form the graph $\mathcal{G}$. Then we sweep again, in the mesh and in the graph $\mathcal{G}$ simultaneously. During this sweep, the seeds are selected. At each event point of the sweep algorithm (the nodes of $\mathcal{G}$), we test whether the arc incident to and below the current node is covered by at least one of the already-selected seeds. If this is not the case, we select a new seed. The new seed will always be the greedy choice, that is, the segment (or cell) for which the function value of the lower end point is minimal. Using the numbering of $\mathcal{T}$ and some additional data structures during the sweep, we can decide efficiently whether an arc is covered and what would be the greedy choice if we need to select another seed. Details are given by van Oostrum (1999).

5.5 TEST RESULTS

In this section, we present empirical results for generation of seed sets using the method of Section 5.4.2 (the method of Section 5.4.1 has not been implemented). In Table 5.1, results are given for seven data sets from various domains, both two-dimensional and three-dimensional. The data used for testing include the following:

Heart: a two-dimensional regular grid of MRI data from a human chest;
Function: a smooth synthetic function sampled over a two-dimensional domain;
Bullet: a three-dimensional regular grid from a structural dynamics simulation;
HIPIP: a three-dimensional regular grid of the wave function for the high potential iron protein;
LAMP: a three-dimensional regular grid of pressure from a climate simulation;
LAMP 2d: a two-dimensional slice of the three-dimensional data, which has been coarsened by an adaptive triangulation method;
Terrain: a two-dimensional triangle mesh of a height field.

The tests were performed on a Silicon Graphics Indigo^2 IMPACT with 128 Mb memory and a single 250 MHz R4400 processor. Presented are the total number of

Table 5.1 *Test results and comparison with previous techniques*

Data	Total cells	# Seeds	Storage	Time (s)	# Seeds of (Bajaj'96)	Storage of (Bajaj'96)	Time (s)
			Structured data sets				
Heart	256×256	5631	30651	32.68	12214	255	0.87
Function	64×64	80	664	1.23	230	63	0.15
Bullet	$21 \times 21 \times 51$	8	964	2.74	47	1000	0.30
HIPIP	$64 \times 64 \times 64$	529	8729	121.58	2212	3969	3.24
LAMP 3d	$35 \times 40 \times 15$	172	9267	6.82	576	1360	0.33
			Simplicial data sets				
LAMP 2d	2720	73	473	0.69	–	–	–
Terrain	95911	188	2078	13.67	–	–	–

cells in the mesh, in addition to seed extraction statistics and comparisons to a previously known efficient seed set generation method. The method presented in Section 4.2 represents an improvement of two to six times over the method of Bajaj et al. (1996). The presented storage statistics account only for the number of items, and not the size of each storage item (a constant).

5.6 CONCLUSIONS AND FURTHER RESEARCH

This chapter presented the first method to obtain seed sets for contour retrieval that are provably small in size. We gave an $O(n^2 \log n)$ time algorithm to determine the smallest seed set, and we also gave an algorithm that yields small seed sets and takes $O(n \log^2 n)$ time for functions over a two-dimensional domain and $O(n^2)$ time for functions over higher-dimensional domains. The latter time bound can be improved using more recent results on constructing contour trees (Carr et al., 2003).

Test results indicate that seed sets resulting from the methods described here improve on previous methods by a significant factor. Storage requirements in the seed set computation remain sublinear, as follows from the test results.

ACKNOWLEDGEMENT

This work was performed under the auspices of the US Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

6

Surface Shape Understanding Based on Extended Reeb Graphs

Silvia Biasotti, Bianca Falcidieno and Michela Spagnuolo

6.1 INTRODUCTION

Knowledge about the global properties of a shape and its main features is very useful for the comprehension and intelligent analysis of large data sets: the main features and their configuration are important to devise a surface understanding mechanism that discards irrelevant details without loosing the overall surface structure. As far as terrain surfaces are concerned, it is also important that a description captures important topographic elements, such as peaks, pits, and passes, which have a relevant semantic content and, at the same time, are formally well defined. Critical points and their configuration, indeed, and the related theory of differential topology give a suitable framework for formalising and solving several problems related to shape understanding. Computational topology techniques provide several tools and measures for surface analysis and coding (Dey et al., 1999): Euler's equation, Morse theory, surface networks or Reeb graphs, for example, provide highly abstract shape descriptions, with several applications to the understanding, simplification, and minimal rendering of large data sets.

Obviously, the best shape descriptor does not exist, and each gives a specific view of a shape. For example, surface networks give a region-oriented description of a terrain, which can be seen as decomposed in patches having their vertices at critical points; Reeb graphs, conversely, give a volume-oriented description in which hills and dales are represented explicitly together with their elevation-based adjacency relationships.

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

To use topological approaches in a computational context and for discrete surfaces, it is necessary to adapt to discrete surface model concepts developed for smooth manifolds, such as piecewise linear approximations. In this chapter, the notion of extended Reeb graph (ERG) is introduced; it is based on a characterisation strategy, which defines critical points and areas by analysing the evolution of the contour levels on a shape, including the so-called *degenerate configurations* also. An algorithm for the construction of the ERG extraction is also proposed.

The remainder of this chapter is organised as follows: first, an overview of the definition of critical points and Morse complexes for smooth manifolds is given; then, topological structures used for the analysis and simplification of triangular meshes are described, focusing on surface networks and Reeb graphs; the characterisation, based on a surface slicing approach, and the ERG representation are presented in Section 6.3; finally, in Section 6.4, an algorithm for implementing the characterisation and the ERG extraction from triangular meshes is presented together with several examples; discussions and conclusions end the chapter.

6.2 BACKGROUND: DIFFERENTIAL TOPOLOGY FOR SURFACE CHARACTERISATION

Theoretical approaches based on differential topology and geometry give complete answers to the problem of understanding and coding the shape of scalar fields. In general, the configuration of the critical points gives sufficient information to fully characterise the surface shape with diverse formal codings, which highlight slightly different properties of the surface. The best example is the Morse theory, which sets the background for surface networks and Reeb graphs, both being effective tools for coding the surface shape. In this section, some topological techniques for surface shape descriptions are introduced, which propose different organisation and coding of the relationships among the surface features, focusing on the Reeb graph representation (Reeb, 1946, Shinagawa et al., 1991).

6.2.1 Morse theory

Morse theory is a powerful tool for capturing the topological structure of a shape. In fact, Morse theory states that it is possible to construct topological spaces equivalent to a given differential manifold describing the surface as a decomposition into primitive topological cells, through a limited amount of information (Guillemin and Pollack, 1974, Milnor, 1963).

Formally, let M be a smooth manifold, that is, a space for which each point has a neighbourhood locally homeomorphic to the open unit ball B^n in $\Re^n$, and let $f: M \rightarrow \Re$ be a real smooth function defined on the manifold M, whose critical points are those where the gradient is zero. Then, the following definition is given:

Definition 1 *(Morse function) The function f is called a Morse function if all of its critical points are non-degenerate, where a critical point is non-degenerate if the Hessian matrix $\boldsymbol{H}$ of the second derivatives of f is non-singular at that point.*

It follows that a Morse function has to be at least C^2. Non-degenerate critical points are isolated, and, in a neighbourhood of each critical point P, the function f can be expressed in a local coordinate system as $f = f(P) - (y_1)^2 - \cdots - (y_\lambda)^2 + (y_{\lambda+1})^2 + \cdots + (y_n)^2$, where λ is called the index of f in P and it represents the number of negative eigenvalues of the Hessian matrix in P. Additional details can be found in (Griffiths, 1976, Guillemin and Pollack, 1974, and Milnor, 1963).

In the case of terrain surfaces, which are modelled by single-valued functions, the reference manifold M is a two-manifold with boundary, where all points, except those along the boundary, have a neighbourhood homeomorphic to a sphere of dimension 2, that is, to a disk. Points on the boundary have a neighbourhood homeomorphic to a half-disk.

Isolevels, that is, subsets of M having the same value of f, can also be used to describe the surface shape. Isolevels are also called *contours* or *level sets*. The topological changes in the isolevel configuration, that is, contour splitting or merging, only occur in correspondence of critical points of f. In Figure 6.1, examples of critical points are shown together with the projection of the surface isolevels in their neighbourhood. This property can be easily extended to degenerate critical points such as the monkey saddles and, in a broader sense, to flat regions; in particular, Figure 6.1(c) and Figure 6.1(d) highlight two degenerate situations, a monkey saddle and a volcano rim respectively. In Section 6.3, we will see how the evolution of isolevels on a manifold M is used to define the Reeb graph of the manifold.

Critical points are classified as maxima, minima, and saddles, according to the behaviour of the function f around them: all the outgoing directions from a maximum (resp. minimum) point are descending (resp. ascending), while a saddle alternates at least two ascending and two descending directions.

In addition, given a Morse function f, a smooth manifold without boundary satisfies the so-called Euler formula, which states that the number of non-degenerate maximum (M), saddle (p), and minimum (m) points verifies the relation $M - p + m = 2(1 - g) = \chi$, where $\mathbf{g}$ represents the genus of the surface and $\boldsymbol{\chi}$ is called the Euler characteristics of the surface. However, considering the right contribution of each critical point, this relation can be extended to the degenerate ones, as shown in (Attene et al., 2003, and Biasotti et al., 2002).

Among all the possible Morse functions, the height function, which associates to each surface point its elevation, may be effectively used to study the surface shape in the Euclidean space. In particular, the level sets of a height function associated to a surface are the intersections of the surface with planes orthogonal to a given direction.

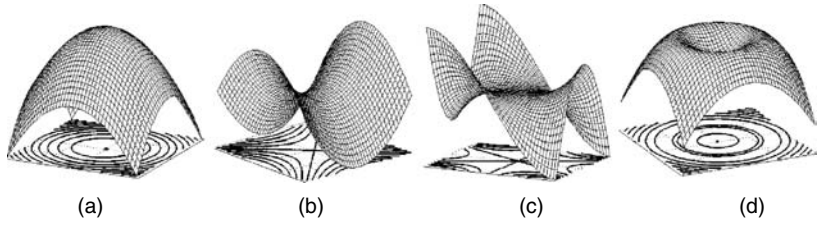

Figure 6.1 *The behaviour of the contour levels around (a) a maximum; (b) a saddle; (c) a monkey saddle; and (d) a volcano rim*

In (Banchoff, 1970), Banchoff presented a full framework that may be regarded as the discrete counterpart of the Morse theory, where critical points and their relationships are formally defined for triangle meshes. A basic assumption of this approach and its derived applications (Bajaj and Schikore, 1998), concerns the behaviour of the scalar field at the vertices of the triangle mesh, since adjacent vertices, that is, vertices joined by an edge, are required to have different field values. This hypothesis is needed to avoid the typical problem represented by degenerate critical points, that is, non-isolated critical points such as plateaux and flat areas of the surface. Methods proposed in the literature usually do not consider the problem, delegating the solution of problematic cases to local adjustments or perturbations. This strategy, however, while solving the problem theoretically, can lead to a wrong interpretation of the shape by introducing artefacts, which do not correspond to any shape feature. Also, many of the proposed computational approaches suffer from numerical instability since a lot of degeneracies occur in real situations.

6.2.2 Surface networks

As Maxwell already guessed, critical points play a fundamental role for fully understanding the global topology of a shape. Topological networks, which code the relationships among the critical points, have been extensively studied; in particular, surface networks have been proposed by Pfaltz (Pfaltz, 1976) for the analysis of geographical surfaces. Such structures code in a graph the relation among the critical points of a surface, which are joined in the structure if there is an *integral curve* connecting them, that is, a curve everywhere tangent to the gradient vector field. Integral curves originate from a critical point and flow to another critical point, or boundary component, and follow the maximum increasing growth of the height function; hence, they cannot be closed (nor infinite) and do not intersect each other except at the critical points. In practice, integral curves originate from each minimum in every direction and converge either to a saddle or a maximum, while only a finite number of integral curves can start from a saddle point.

Nackman in (Nackman, 1984) introduced the idea of *critical point configuration graph*. Under this hypothesis, the height function is Morse. He demonstrated that a surface network can assume only a finite number of configurations on the surface, which induce a surface subdivision into zones of constant first derivative behaviour, the so-called *slope districts*. In particular, the slope districts are classified into four classes only. Then, the surface networks can be represented through a limited number of primitives, whose nodes are the critical points and whose arcs are detected through the steepest ascending directions on the surface.

For applications of the surface network framework to the GIS context, see this book, Part II.

6.2.3 The Reeb graph

In this chapter, we are focusing on the approach proposed by Reeb to code the evolution and the arrangement of isolevel curves (Reeb, 1946). In the general case, the Reeb graph of a manifold M under a mapping function f is defined as follows:

Definition 2 *(Reeb graph) Let* $f: M \to \mathbb{R}$ *be a real-valued function on a compact manifold* M*. The Reeb graph of* M *with respect to* f *is the quotient space of* $M \times \mathbb{R}$ *defined by the equivalence relation '*$\sim$*' given by*

$$(X_1, f(X_1)) \sim (X_2, f(X_2)) \Leftrightarrow f(X_1) = f(X_2) \text{ and } X_1 \text{ and } X_2 \text{ are in the same connected component of } f^{-1}(f(X_1)).$$

Therefore, the Reeb graph of M collapses into one element all points having the same value under the real function f and being in the same connected component. Moreover, since the topological changes of the level sets occur only in correspondence to critical points, the Reeb quotient space can be effectively represented as a graph structure: a node is defined for each critical level of f, which corresponds to the creation, merging, split, or deletion of a contour, that is, to topological changes affecting the number of connected components in the counter-image of f; at each node, a number of arcs is defined corresponding to the number of connected components of the counter-image of f, each joining two successive critical levels in their own component. If an arc joins two nodes n_1 and n_2, then the topology of isolevels on M between the height levels n_1 and n_2 does not change along the connected component of M joining the corresponding critical points.

Therefore, the Reeb graph of M under the height function f can be defined as $RG_f(M) = (P_f(M), A_f(M))$, where the node set is defined by $P_f(M) = \{P_i \in M, P_i \text{ is a critical point of } f(M)\}$ and the arc set $A_f(M)$ is defined as stated before.

The arcs of $RG_f(M)$ can be oriented according to the increasing value of the height function f, that is, if $a = (n_1, n_2)$ is an arc of the graph, then $f(n_1) < f(n_2)$. Since the arcs of $RG_f(M)$ are oriented, no oriented path of $RG_f(M)$ can start and end at the same node; hence the Reeb graph is acyclic. Moreover, if f is Morse, the nodes have at most degree three.

With regard to terrain surfaces, these are mathematically modelled as scalar fields $h: D \subseteq \mathbb{R}^2 \to \mathbb{R}$ such that $h: (x, y) \to z = height(x, y)$. In this case, the manifold is defined by the points in $M = \{P \in \mathbb{R}^3 / P = (x, y, h(x, y))\}$ and the height function f is naturally defined over M as $f(P): M \to \mathbb{R}$ such that $f(P) = f((x, y, h(x, y)) = h(x, y)$. Terrain surfaces are therefore represented by scalar fields with boundary, but the Reeb graph can be always defined by adding a *minimum* to the set of critical points, which virtually closes the surface and makes it homeomorphic to a sphere, as shown in (Biasotti et al., 2000, Takahashi et al., 1995, and Wood and Rana, 2000). Reeb graphs of terrain surfaces can be always represented as trees, where the root is given by this virtual closure of the surface.

The Reeb graph of a terrain surface M, under its natural height function, codes the shape of M in terms of the critical points of f, which are associated to meaningful topographic features, that is, peaks, pits, or passes, structured into a topologically consistent framework.

In Figure 6.2(a), the points drawn on the manifold represent the equivalence classes of an elementary terrain surface with respect to the height function. In Figure 6.2(b), the Reeb's quotient space is represented as a *traditional* graph where the equivalence classes are grouped into arcs.

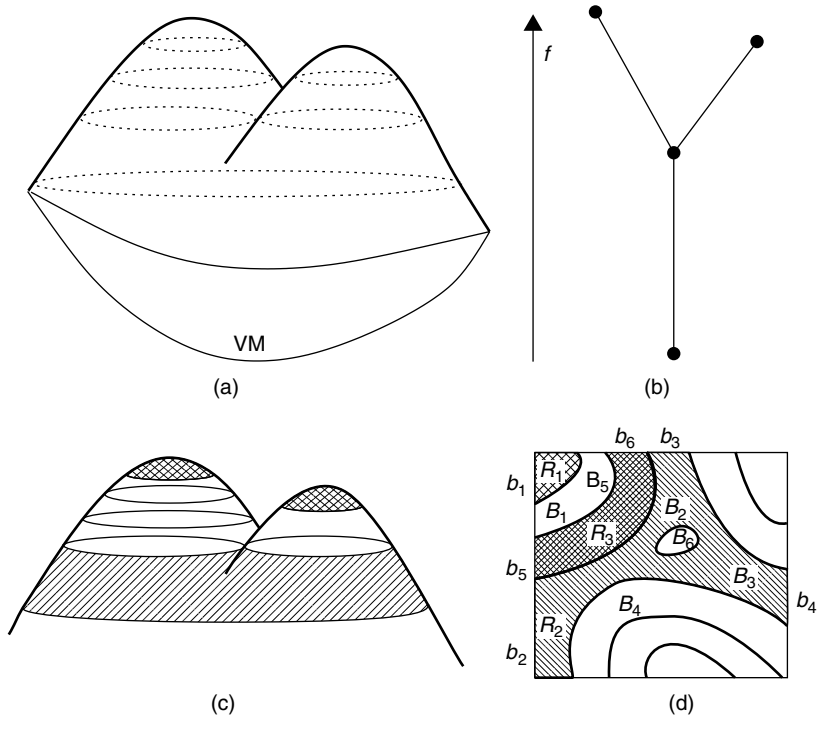

Figure 6.2 *(a) Reeb equivalence classes (dotted lines) and (b) Reeb graph of a simple surface. The introduction of a virtual minimum makes the surface topologically equivalent to a sphere. The dark regions in (c) are critical areas, the white are the "regular" ones. In (d), the regions R_1, R_2, and R_3 and their boundary components are highlighted; the capital labels indicate the contour lines and the small ones are portions of the surface boundary*

Since the choice of the height function depends on the surface embedding, a manifold admits different Reeb graphs; however, this is not a problem for terrain surfaces, which have a natural privileged direction.

Since the Reeb graph is not limited to scalar fields but is really useful for analysing surfaces of arbitrary topology, it might also be extended to represent more general terrain surfaces that also have vertical walls or cavities.

6.3 GENERALISED SURFACE CHARACTERISATION

As shown in Section 6.2, knowledge about critical points is crucial for understanding and organising the topological structure of a surface. Unfortunately, the hypothesis that a surface is only continuous does not guarantee that the associated height function is Morse, or derivable. Moreover, it would be desirable to distinguish between small details and relevant features of the surface, especially when dealing with rough surfaces as terrains. Many of the existing approaches to the characterisation of discrete surfaces use local point-wise criteria to detect and classify critical points: for example, triangle meshes are analysed in (Banchoff, 1970, De Floriani et al., 2002, and Takahashi et al., 1995) by checking the height difference between a vertex and the adjacent ones in

its star-neighbourhood, and by producing a topological coding, which is an adaptation of the surface network structure to piecewise linear surfaces. Two drawbacks can be identified: first, these methods rely on the hypothesis that all edge-adjacent vertices have different heights; second, the number of the detected critical points is usually very high and pruning or simplification steps are necessary to make the resulting structures understandable.

Our aim is to faithfully represent the surface topology and shape, without any height shift at surface vertices, by using an extended characterisation, which can handle degenerate as well as non-simple critical points and can be tuned to filter *small* features. Our approach is based on the use of contours for characterising the surface shape and constructing a topological structure, the ERG, which represents the configuration of the critical areas of the surface. This extended characterisation is a generalisation of our previous work (see Biasotti et al., 2000, 2002), in terms of both characterisation definition and algorithm for the extraction of the Reeb graph. Our approach is also similar to the method proposed in (Jun et al., 2001) for supporting the computation of intersections between parametric surfaces.

6.3.1 Definition of critical areas

A terrain surface M is characterised by sweeping slicing planes along the height direction and analysing the configuration and topological changes of the resulting isolevels, or contours. These contours decompose M into a set of regions whose boundaries contain complete information for detecting critical areas and for classifying them as maximum, minimum, and saddle areas. For example, if a contour does not contain any other contours and its elevation is higher than the successive one, then it identifies a maximum area. Our generalised characterisation corresponds to the localisation of these critical areas on M, aimed at region-oriented rather than point-oriented classification of the behaviour of M. All subsets of M defined by counter-images of critical values of f will be considered *critical area*s of M and they can be points, lines, and regions.

Since terrain surfaces are surfaces with one boundary, it is also necessary to give a unique interpretation of the critical points on the boundary. This is achieved by the insertion of a global virtual minimum point, so that the outgoing directions from the surface boundary are only descending and M is virtually closed.

As shown in Section 6.2, there is a close correspondence between the existence of critical points, or areas, and the evolution of the height contours on the surface. The use of height contours has also an inherent and efficient filtering effect, which is related to the frequency or distribution of the slicing planes.

While the filtering effect will be discussed later in this section, we will assume for now that the variation interval $[f_{\min}, f_{\max}]$ of the height function is uniformly sliced with np planes, at a distance dp between them. The relationship between np and dp is: $np = (f_{\max} - f_{\min})/dp$, and the first plane is located at the height value $f_{\min} + dp/2$. Moreover, we will consider that all contours are non-degenerate, that is, the slicing planes are never tangent to M. Details on the implementation aspects are given in Section 6.4. Let $C(M)$ be the set of the resulting contour levels of the surface M, without any specific ordering. Each contour is either a simple closed line or an open line with the end points on the surface boundary B_M.

The contours in $C(M)$ fully decompose the surface M into sub-regions, which correspond either to *critical* or *regular* areas. Let $B_M(R)$ be the boundary of a region R and bb the number of its connected components; in general, a connected component of $B_M(R)$ may be either a closed contour, or it may be composed by a connected and closed sequence of open contour lines and B_M parts. Note that in this latter case, if this type of component exists, then it is only one corresponding to the external boundary component of the region R. Therefore, the boundary of a region R on M is defined by $B_M(R) = B_1 \cup B_2 \cup \cdots \cup B_n \cup b_1 \cup \cdots \cup b_k$ where $B_i \in C(M)$ and each b_j is a portion of the surface boundary, B_M. Obviously, the boundary components $b_1 \cup b_2 \cup \cdots \cup b_k$ are missing when the region does not intersect B_M, that is, the sub-region R is fully contained within the surface domain.

According to the definition of contours, if an element of $C(M)$ intersects a region R, then it has to be completely part of its boundary $B_R(M)$. If the region R intersects the surface boundary B_M, then the external component of $B_R(M)$ is a closed sequence of open contours connected among them through b_j components, as shown in Figure 6.2(d). With reference to Figure 6.2(d), the boundary components of $R2$ are made of the ordered sequence union b_2, B_4, b_4, B_3, b_3, B_2 and the boundary component B_6; in this case, bb is equal to two. In particular, with reference to the region $R2$, the B_i components correspond to B_2, B_3, B_4, and B_6, while the b_j components are given by b_2, b_3, and b_4.

A generic region R of M is classified according to the number and behaviour of its boundary components. Since the interior of any region R is well defined, it is possible to associate the so-called *outgoing directions* to each component of $B_R(M)$, which are needed to classify the region type. In particular, to all closed components of $B_R(M)$, only one outgoing direction is associated, while to the component intersecting B_M, if any, one outgoing direction is associated to each composing part. Each outgoing direction is classified as ascending or descending according to the behaviour of f across the corresponding boundary component. If the f value decreases (increases) walking from the inside towards the outside of the region through the boundary component B_i, then the associated outgoing direction is descending (ascending). The existence of the virtual minimum, indeed, does not alter the surface characterisation but implies that during the classification process, each boundary component b_j has to be considered as a descending direction.

Given a region R and its boundary $B_R(M)$, the following classification scheme is adopted:

- R is a *maximum* area iff all the outgoing directions from $B_R(M)$ are descending (see Figure 6.3);
- R is a *minimum* area iff all the outgoing directions from $B_R(M)$ are ascending and $B_R(M)$ does not intersect the surface boundary, that is, $k = 0$ (see Figure 6.3(c));
- R is a *saddle* area iff either $k = 0$, $bb > 2$ and there are both ascending and descending outgoing directions from $B_R(M)$, or $k > 0$ and $B_R(M)$ verifies at least one of the following conditions (see Figure 6.3(a, b)):
  - (a) $bb = 1$ and there are at least two ascending outgoing directions;
  - (b) $bb > 1$ and at least one of the open boundary components $B_i \in B_R(M)$ has an outgoing ascending direction;

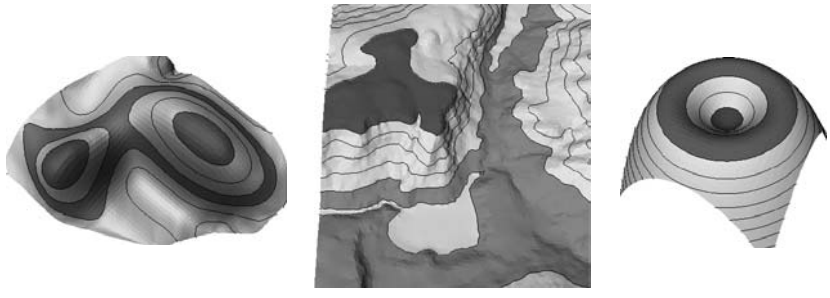

Figure 6.3 *Maximum and saddle characterisation for regions (a) non-intersecting and (b) intersecting the surface boundary. In (c) a minimum and a non-simply connected maximum are presented*

– finally, R is called *regular* iff it does not belong to the previous categories, (see Figure 6.2(c)).

With reference to Figure 6.2(c), the dark regions represent three critical areas, while the white ones correspond to regular areas. In addition to the previous classification scheme, a further distinction between simple and multi-connected minimum and maximum areas is done: *simple* critical areas are minima (maxima) that correspond to a simply connected region and *complex* critical areas are minima (maxima) that correspond to multi-connected regions. Moreover, due to the assumption that all the outgoing directions across the surface boundary B_M are descending, minima cannot be adjacent to B_M and in this sense, the classification of minima and maxima is not fully symmetrical. In particular, the dark regions of the image in Figure 6.3(a) represent critical areas that do not belong to the boundary surface, while the regions in Figure 6.3(b) do.

Let us now discuss the relation between the distribution of slicing planes and the size of the features detected. First of all, for terrain surfaces, the notion of *size* can be associated only to maximum and minimum areas, either simple or complex, and it corresponds to the height difference between the critical level and the closest adjacent saddle level. The adopted uniform slicing guarantees that all features having size greater than dp are detected. Features whose size is less than dp are discarded, except those that extend across a slicing plane. To make the filtering effect homogeneous, the contour behaviour is re-computed at a distance dp from the point q in the critical area, which has the maximum height variation within the region. In Figure 6.4, an example is given: the size of the feature h is smaller than dp and the maximum q disappears when the contour level c_1 is replaced by c_2. In this way, all the features having size greater than dp are recognised and the smaller ones are discarded.

6.3.2 From critical areas to the extended Reeb graph

The generalised characterisation just described can be coded as an ERG by simply extending the equivalence relation used in the Reeb graph. Let $f: M^* \to \mathbb{R}$ be the height function defined on the virtual closing M^* of the surface M, and let $[f_{\min}, f_{\max}]$ be an interval containing the variation interval of f on the surface M, and $f_{\min} <$

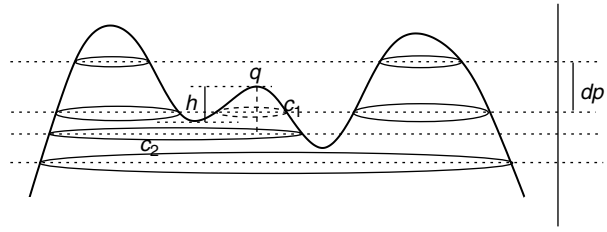

Figure 6.4 *The feature in the middle has size h, which is less than the slicing step dp, hence it is discarded during the characterisation process*

$f_1 < \cdots < f_h < f_{\max}$ the height distribution of the contour levels $C(M)$, which are supposed to be all non-degenerate contours. We observe that the relations $f_{\min} < f_1$ and $f_h < f_{\max}$ holds, because if $f_{\min} = f_1$ and $f_h = f_{\max}$, the horizontal planes would be somewhere tangent to M and some contours would be degenerate. In addition, let $I = \{(f_{\min}, f_1), (f_i, f_{i+1}), i = 1 \ldots h-1, \textit{and } (f_h, f_{\max})\} \cup \{f_{\min}, f_1, \ldots, f_h, f_{\max}\}$ be the partition of the interval $[f_{\min}, f_{\max}]$ provided by the set of the $h+1$ interior parts of $[f_{\min}, f_1, \ldots, f_h, f_{\max}]$ and the height values of the contour levels.

Definition 3 *An extended Reeb equivalence between two points P, $Q \in M^*$ is given by the following conditions:*

- $f(P)$, $f(Q)$ belong to the same element of $t \in I$;
- $f^{-1}(f(Q))$ and $f^{-1}(f(Q))$ belong to the same connected component of $f^{-1}(f(t)), t \in I$.

Therefore, by applying the notion of the quotient relation in Definition 3, it follows that all the points belonging to a region R are Reeb-equivalent in the extended sense and they may therefore collapse into the same point of the quotient space. The quotient space obtained from such a relation is called *extended Reeb* (*ER*) *quotient space*. Moreover, the ER quotient space, which is an abstract sub-space of M^* and is independent from the geometry, may be represented as a traditional graph, which is called the *extended Reeb graph* (*ERG*).

To represent the ER quotient space as a graph, the classes that are defined by points on contours are represented by *connecting points*, while all other classes are represented by normal points, simply called *points*. Connecting points are representative of contours and normal points are representative of regions. A point p representing a region $\mathbb{R}$ is adjacent through a *connecting point* to another point q representing another region $\mathbb{R}'$ in the quotient space, and a normal point is adjacent to as many connecting points as the number of connected components of the boundary of the associated region. From this point of view, the image of a regular region of M^* in the ER quotient space is adjacent only to two connecting points. Therefore, the connectivity changes of the graph representation are concentrated in the image of the critical areas, and they are equivalent to the standard Reeb graph representation that can be easily derived by merging the intermediate nodes representing regular areas into a single arc. After this merging step, the ERG simply consists of nodes representing critical areas and the associated connecting arcs.

Finally, in the Reeb representation, complex areas are distinguished from simple ones by labelling the graph nodes *macro-nodes* in the former case, and *nodes* in the latter one; that is, the macro-nodes are those particular leaf nodes with only ingoing or, respectively, outgoing arcs and whose degree is at least two.

Starting from the surface characterisation previously defined and considering the introduction of the global virtual minimum, V_M, the relationship among the critical points expressed in the Euler formula may be recovered also for the critical areas, as shown in (Attene et al., 2003, and Biasotti et al., 2000). The generalised Euler formula has to take into account the number of simple as well as complex critical areas. For each complex critical area, c_a, we consider the number $mc_a = i_b - 1$, where i_b represents the number of inner boundary components of c_a. Then, if P_{mc} is the sum of all the contributions of the complex areas, the Euler formula in Section 6.2 becomes $M - p + m - P_{mc} + VM = \chi$. The contribution of the ith critical area is provided by $2 - bb_i$, where bb_i is the number of its boundary components and the Euler relation: $V_M + \Sigma(2 - bb_i) = \chi$. Because the number of boundary components of such a critical area corresponds to the degree of the node in the Reeb graph G, the previous relation can be re-written as $\Sigma(2 - \delta_i) = \chi - V_M$, where δ_i is the degree of the ith node of G. Considering that the sum of all the node degrees is twice the number of arcs E of G (as each arc is computed in the sum for two nodes) and the contribution of V_M is one, the previous relation can be further expressed by: $2(N - E) = \chi - 1$, where N represents the number of critical areas of M.

6.4 ERG EXTRACTION

As shown in Section 6.3, the quotient space defined by the ER equivalence relation can be represented in terms of a graph. Through the extended definition of critical areas proposed in Section 6.3.1, the application domain can be extended to generic continuous surfaces, without any artefacts (Biasotti et al., 2000). Then, the approach proposed in this chapter is actually not an extension of the Reeb graph itself, but rather a full application of its definition in the discrete domain, which does not require the height function to be Morse.

In this section, a short description of the algorithm for characterising a triangle mesh is given. The extraction and classification of critical areas is done first by computing and inserting a suitable number of contours into the triangle mesh, and second, by reconstructing and classifying the boundaries of the regions delimited by the inserted contours, according to the scheme proposed in Section 6.3.1.

The computation and the insertion of the contours into the mesh is done in a single step. The contour levels $C(M)$ inserted into the mesh model are used as constraints for the region-detection process, which uses a region-growing strategy. The insertion of a contour C into M is computed as follows: given a slicing plane π, a seed point $p \in C$ is computed by selecting an edge e, which properly intersects π, that is, e does not belong to π nor does it intersect π in a vertex. C is extracted by starting from p and moving horizontally by adjacency on the mesh until either p or the surface boundary is reached. If the surface boundary has been reached, C is an open contour and the algorithm restarts from p in the opposite direction until the surface boundary is reached again. If the points of C are not vertices of the mesh, they are inserted into

the mesh. The mesh is locally re-triangulated in order to obtain a valid mesh, and the contours are inserted into the mesh as constrained edges. This process stops when all the planes have been considered. This procedure guarantees that degenerated contours such as points, lines, and so on are not taken into account. Then, the intersections of the model with the slicing planes are computed and stored as a set of connected components, which can also be open, in correspondence with the surface boundary intersection.

The insertion of $C(M)$ decomposes the triangle mesh into a set of regions, each bounded by $C(M)$ elements and mesh boundary edges. These regions are detected by labelling all triangles in the mesh with a region-growing process that propagates the label from a triangle to its adjacent ones without crossing any constraint. At the end of this labelling phase, all triangles having the same label identify a region. Then, the boundary of each region is detected and the associated outgoing directions are classified. Starting from any edge of the region boundary, the associated connected component is fully traced using edge–vertex adjacency. If the component is closed, then there is only one outgoing direction, which can be easily classified by checking the elevation of any vertex inside the adjacent region. If the traced component is open, then the tracing has to continue along the mesh boundary as well, and the whole component will consist of a sequence of open contours and boundary parts. The tracing can be done since all triangles are labelled with the region label. In this case, each part of the boundary component defines an outgoing direction that has to be classified. Finally, the number of boundary components *bb* and their classification allow distinguishing between simple and complex critical areas.

According to the graph representation of the ER's quotient space, each node of the graph corresponds to a critical area; in particular, when the critical region recognised as a maximum/minimum area is complex, a macro-node is defined with as many arcs as the inner components of the critical region. Since each arc corresponds to a connected component of the manifold between two critical areas, the Reeb graph extraction is based on tracking the evolution of contour lines.

When the critical areas have been recognised, the *ERG* is initialised by creating the node corresponding to the virtual minimum, V_M. The V_M is connected to the saddle having the minimum elevation and external to each macro-node. If such a saddle does not exist, the V_M is connected to the nearest (in terms of geodesic distance) complex maximum area; otherwise, if there are not complex maxima, the *ERG* is a trivial graph connecting the V_M to the only simple maximum existing and the surface is topologically equivalent to a sphere (Milnor, 1963).

Our algorithm for the extraction of the ERG runs in two steps: first, the arcs between minima (maxima) and saddles are inserted; then the other arcs are detected. In the following, a construction algorithm is described using a C pseudo-code:

```
/*The ERG is defined by the set of nodes, N, and of arcs,
   A*/
ERG_Construction(N,A) {
  /* Identify critical areas and initialise the virtual
   minimum */
 N=CriticalAreasRecognition(tin, contours);
 /* Order the Critical Areas by elevation      */
```

```
OrderAreas(N);
/*Create a virtual minimum and connect it to the node the
  most appropriate */
ConnectVirtualMinimum(N);
/*Leaf arc extraction */
ExpandMaxima&Minima(N,A);
for (each node in N) {
  if (IsGrowingArea(node)) {
   for ( each non visited growing direction node) {
     while ((not(findBoundarySurface)) or
  (not(findOtherCriticalArea)))
       ExpandToUpperLevel(node);
       if (R=OtherAreaReached)
           ConnectWithArc(node, R);
     }                      /* end for */
   }                        /* end if */
}                          /* end for */
```

The function `ExpandMaxima&Minima(N)` connects all the maxima and minima to their nearest (in terms of region expansion) critical area and extracts a subset of Reeb arcs, while the function `IsGrowingArea(node)` returns a Boolean value, which is `TRUE` if the critical area has at least one growing direction that has not been visited yet. In Figure 6.5, the main steps of the *ERG* extraction process are depicted; Figure 6.5(a) represents how the maxima (minima) are expanded until other critical areas are reached and the corresponding graph representation, while Figure 6.5(b) shows how the algorithm works for completing the area expansion process.

Some results of our *ERG* extraction for real terrains are provided in Plate 2. The nodes of the *ERG* representation are coloured according to the meaning of the corresponding critical areas in the models. In particular, the maxima are depicted in red, the minima in blue, and the saddles in green, while the virtual minimum is represented in yellow. Moreover, we show the simplified models obtained considering only the mesh vertices, which form the boundary of all the critical areas of the models. The original models of Plate 2(a) and 2(c) have 160,000 and 129,600 vertices, respectively while the simplified ones in Figure 6.6 have 19,200 and 26,200 vertices respectively; it is important to point out that the simplification provided by the *ERG* mainly depends on the *topological* complexity of the models rather than on the number of the original vertices.

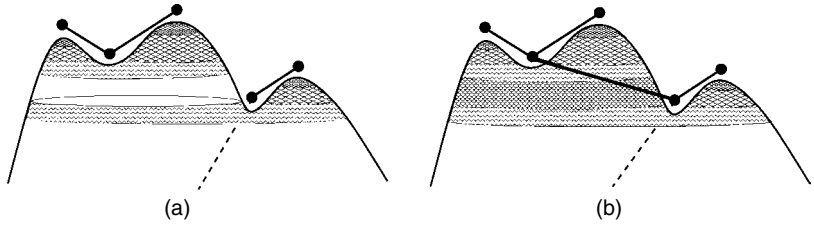

Figure 6.5 *Two steps in the pipeline of the ERG extraction*

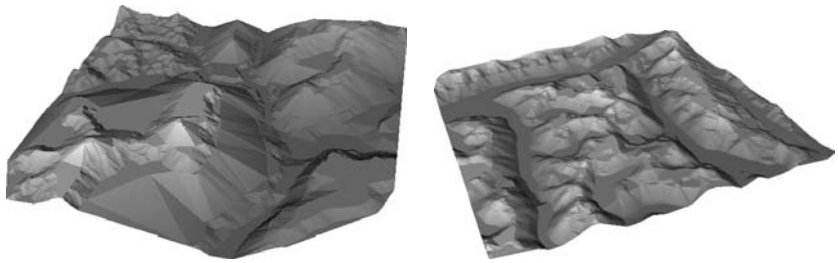

Figure 6.6 *Examples of simplification obtained by considering only the boundaries of critical areas: in (a) the simplified model of the terrain given in Plate 2(a), and in (b) that of Plate 2(c)*

6.4.1 Computational complexity

The computational cost of the whole algorithm for the ERG extraction is given by the sum of the cost of its single subparts, that is, the insertion of contour levels into the mesh, the extraction of the critical areas, and the final expansion process. Given the surface mesh, the insertion of the contour levels $C(M)$ depends on both the number of vertices of the original triangulation, n, and the number m of the vertices of $C(M)$. Because the number of edges and triangles has the same order as the number of vertices, checking the edge-to-plane intersection requires $O(\max(m, n\log(n)))$ operations. In fact, the edges of the mesh are sorted in $O(n\log(n))$ operations, while $O(\max(m, n))$ is the number of intersection tests. Finally, the insertion of the whole set of constraints requires $O(m)$ edge splits.

With regard to the computational complexity of the characterisation process, if the recognition of critical areas is linear in the number of mesh triangles, then it requires $O(n + m)$ operations, because the number of triangles in the constrained mesh has the same order as the sum of original vertices and the constrained ones. Also during the arc completion step, the triangles are processed once and the complexity still is $O(n + m)$, so that the total computational cost of the *ERG* extraction mainly depends on the insertion of contours into the mesh. Therefore, the whole process, starting from a generic triangulation, requires $O(\max(m + n,\ n\log(n)))$ operations. Finally, we observe that, if we consider a generic triangle mesh, the average size of m is $O(n\log(n))$, even if in the worst case, m is $O(n^2)$.

6.5 DISCUSSION AND FINAL REMARKS

The generalised characterisation and the ERG coding provide a compact representation of the main features of a terrain surface, which is effectively represented as a configuration of hills and dales.

With regard to the feature extraction step, the mesh characterisation based on the classical height comparison at mesh vertices, as classically proposed in (Banchoff, 1970, De Floriani et al., 2002, and Takahashi et al., 1995), can be recovered also through our method. It is sufficient, indeed, to slice the mesh in correspondence to the midpoint of each edge; in this way, all the original mesh vertices would lie in a

separate region and the characterisation obtained through the mesh contouring would be equivalent to considering the star region of each vertex.

Finding the best compromise between the effectiveness of feature extraction and the number of slicing planes is the most critical point of the method. The first solution is to characterise the mesh as proposed in (De Floriani et al., 2002 and by slicing the mesh with planes placed at optimal positions: one plane directly below (above) maxima (minima), and two planes for saddles, one above and one below. In this case, the number of slicing planes considerably decreases but the number of features does not, and the results would still be sensitive to small variations of the vertex elevation.

Using the uniform slicing, the surface shape is described by the topological coding of its features at a fixed resolution dp. In many cases, however, a description at different scales could be more effective. This could be achieved by adopting a multi-resolution slicing process of the mesh as proposed in (Hilaga et al., 2001): a sequence of Reeb graphs can be extracted by halving the distance interval between the slicing planes until a threshold defined by the user is reached. At each step, new nodes and arcs might be inserted into the graph as shown in Figure 6.7, but there is a hierarchical relation between the nodes of the current graph and the previous one (Attene et al., 2003).

In our setting, a multi-resolution ERG extraction can be implemented by iteratively halving the height interval $[f_{\min}, f_{\max}]$; for example, the graphs proposed in Plate 2 have been obtained with 32 subdivisions of the interval $[f_{\min}, f_{\max}]$. The power of this approach is clear: the surface shape can be processed at different levels of detail and the estimation of its features is automatically provided.

In addition, we notice that by adopting the mesh characterisation approach based on the neighbours of each vertex, the Reeb graph is equivalent to that provided by the contour tree, as proposed in (Carr et al., 2003, and van Kreveld et al., 1997). In fact, both structures have a common root in Maxwell's paper and pursue the aim of organising the contour levels of a two-dimensional surface in a systematic and topologically correct way. However, the contour trees have been proposed only for scalar fields, while the Reeb graphs have been studied for the generic two-manifold and successfully applied to arbitrary complex surfaces; as an example, our approach also works on terrain surfaces with vertical walls and cavities.

Considering simple Morse functions, that is, functions whose critical points are non-degenerate and not at the same level, Reeb graphs and surface networks may be easily compared: the Reeb graph is a subgraph of the surface network, at least for the arcs not involving the boundary. An algorithm for the extraction of Reeb graphs from surface networks has been, for example, proposed in (Takahashi et al., 1995). Both graphs code the topological structure of a surface, with surface networks giving a surface-oriented view, while Reeb graphs give a skeleton-like and volume-oriented description.

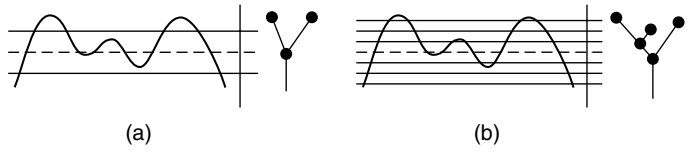

Figure 6.7 *Reeb graph variation when halving the distance among the sections*

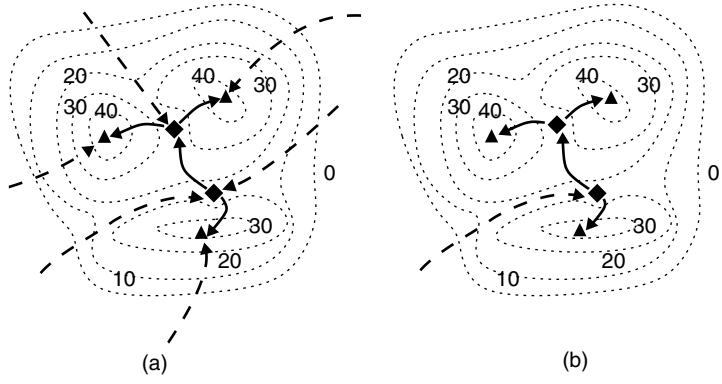

Figure 6.8 *(a) The surface network structure and (b) the Reeb graph of the same terrain model*

In Figure 6.8, the surface network of a terrain represented by contours is compared with the corresponding Reeb graph; all the arcs of the surface network coming from the outside of the surface boundary originate from a virtual minimum, which is depicted for the Reeb graph structure.

In the generalised version presented in this chapter, surface networks and ERG cannot be directly compared. Surface networks obviously fail if degenerate critical points exist, and, to our knowledge, there is no way to automatically filter the resulting features during the network delineation process. Conversely, the ERG construction process is stable and provides a simplified configuration of the terrain features, which easily and efficiently supports the minimal rendering of large terrain data.

ACKNOWLEDGEMENTS

The authors would like to thank all the people of the Shape Modelling Group at IMATI-CNR. This work has been partially supported by the National MIUR Project 'MACROGeo: Metodi Algoritmici e Computazionali per la Rappresentazione di Oggetti Geometrici', FIRB grant.

Part II

Applications

7

A Method for Measuring Structural Similarity among Activity Surfaces and its Application to the Analysis of Urban Population Surfaces in Japan

Atsuyuki Okabe and Atsushi Masuyama

7.1 INTRODUCTION

In urban and regional studies, we often study the differences in regional characteristics in comparison with the distributions of a common attribute value (say, population density) of different regions. When the distribution is continuous in each city, we represent it by a surface in three-dimensional space, for example, the surface representing a population density as in Plate 3. Since such representation is commonly employed in urban and regional studies, we consider it important to develop a method for measuring similarity or dissimilarity among the surfaces. In particular, we are interested in "qualitative" or structural similarity rather than "quantitative" or detailed similarity. This is because in the humanities and the social sciences, the quality of geographical data is often poor owing to spatial aggregation, and measuring quantitative difference in a very precise manner (as in natural sciences) is not always meaningful. In this chapter, we attempt to formulate a general method for measuring structural or skeletal similarity by use of surface networks, and we apply the method to a comparative study on urban population densities in Japan. Although this application is specific, the method itself is so general that it may be applied to the analysis of a broad class of two-dimensional continuous distributions treated in humanities and social sciences.

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

In urban studies, much effort has been devoted to the analysis of urban population densities since the 1950s. A pioneering work was done by Clark (1951), who shows that the population density, $z = f(x)$, in a city tends to decrease negative-exponentially as the distance, x, from the centre of the city increases, that is,

$$f(x) = a \exp(-bx). \tag{7.1}$$

This finding, called the *Clark law*, has been examined by many researchers, for example, Berry et al. (1963), Brush (1968), Casetti (1967, 1969), Clark (1958, 1967), Duncan et al. (1961–1962), Kramer (1958), Latham and Yeats (1970), Mills (1972), Muth (1965, 1969), Newling (1969), Ohtomo (1979), and Sherratt (1960). In their studies, comparative analysis is achieved by comparing coefficients a and b estimated by data in each city. This comparison is possible because the implications of the coefficients a and b are common to any city (i.e. the coefficient a implies the population density at the centre and the coefficient b implies the degree of decrease in population density with respect to distance).

A problem with these studies, as is noticed from the form of $z = f(x)$, is that they assume one-dimensional space. Obviously, the actual geographical space is two-dimensional, $z = f(x, y)$. To take this fact into account, the trend surface analysis developed in geology (e.g. Krumbein, 1956) was found to be useful in urban studies. This analysis treats a surface in terms of a polynomial function, that is,

$$f(x, y) = a_1 x^n + a_2 y^n + a_3 x^{n-1} y + \cdots + a_{m-2} x + a_{m-1} y + a_m. \tag{7.2}$$

As is seen in (Johnson and Vance, 1967, Norcliffe, 1969, Watson, 1972, Bassett, 1972, and Whitten 1974), this method is useful for analysis of one city, but it is not appropriate for the comparative analysis of many cities. A reason for this is that the estimated values of coefficients $a_1, \ldots, a_m$ vary according to the location of $x - y$ axes, and so the coefficients in different regions are not comparable. To overcome this difficulty, a few methods are proposed in the literature. For example, Haggett and Bassett (1970) proposed a measure of similarity that was invariant with respect to the location of the $x - y$ axes. The meaning of their measure, however, is not straightforward.

A few methods that are different from the trend surface analysis are proposed in the literature. Okabe and Sadahiro (1994) use the Kullback–Leibler information index to examine the relationship between a population distribution and a retail potential distribution in a region. Although this index works well in statistical contexts, sometimes it does not work well in geographical contexts. King (1969) proposes a method that extends spectrum analysis of time (one dimension) to that of two-dimensional space. His method assumes that the data units are squares in a rectangular area, but this assumption is not always satisfied in practice.

All the methods mentioned above examine quantitative similarity but, as mentioned above, it is often more important in the humanities and social sciences to examine structural or skeletal similarity rather than quantitative or detailed similarity. To deal with skeletal similarity, the surface network method proposed by Pfaltz (1976) and the activity contour method proposed by Fujii (1978) are useful. In fact, applying a similar method to the analysis of urban population densities in Japan, Okabe and Masuda

(1984) clearly show skeletal difference among population densities in Japanese cities. A similar method was also employed by Sadahiro (2001), who showed a structure in the distribution of retail stores in Tokyo. Their methods, however, have ambiguity in choosing a parameter value, which causes some arbitrariness. This chapter proposes a method to overcome this shortcoming.

7.2 INTRODUCTION TO SURFACE NETWORKS

To understand the key concept of this chapter on surface networks, readers need to be familiar with what surface networks are. We feel, however, that concepts related to surface networks are not always well known to the humanities and social scientists. This book is useful for learning these concepts, but chapters in this book may seem to be slightly too technical for researchers in the humanities and social sciences who are not familiar with surface networks. Since we expect our method to be applied to the analysis of phenomena studied in the humanities and social sciences, this chapter first presents an intuitively understandable introduction to surface networks. If readers are familiar with surface networks, they can skip this section and go to Section 7.3. If readers wish to understand the theories and computational methods of surface networks in depth, they should consult the chapters in this book.

We consider a distribution of an attribute value (say, population density) over a region S. Mathematically, this distribution is represented by a function, $z = f(x, y)$, where z is an attribute value at (x, y) in S. If we indicate the value of z with the height at (x, y), the function is depicted in the three-dimensional space as in Plate 3, which looks like the surface of a mountain. We call this surface an activity surface (in the case when the attribute value is population, it is called a *population surface*). We denote an activity surface in S by T. The activity surface T(Plate 3) can alternatively be represented by contour lines as in Figure 7.1(a). This representation is often used in topographical maps.

An actual mountain surface may have cliffs and overhangs, but to gain analytical tractability, we assume that the activity surface T is very smooth (mathematically, second-order differentiable), and proper (mathematically, non-degenerate). We also assume that $f(x, y) \geq 0$ in S and $f(x, y) = 0$ on the boundary of S. The former assumption does not lose generality, because if $f(x, y) \geq -a$ $(a > 0)$, we use $f(x, y) + a \geq 0$, and this modification does not affect the derivations in Section 7.3. The latter assumption is satisfied with the following modification. We spread a dummy downward surface around the periphery of S in such a way that the surface smoothly joins the edge of the surface T, and the bottom edge of the surface touches the ground $(z = 0)$ (this modification is easily made if we use software for generating contour lines). This modification does not affect the derivations in Section 7.3. We make these assumptions only for gaining analytical tractability.

Under the above assumptions, we consider part of an activity surface T in a sufficiently small area around a point in S, and call it the *local (activity) surface* at that point. Local surfaces have various forms at points in S, but they are classified into four categories: peaks, bottoms, cols, and slopes (see Figure 7.2). Peaks, bottoms, and cols are subsumed under critical points. The common property of the critical points

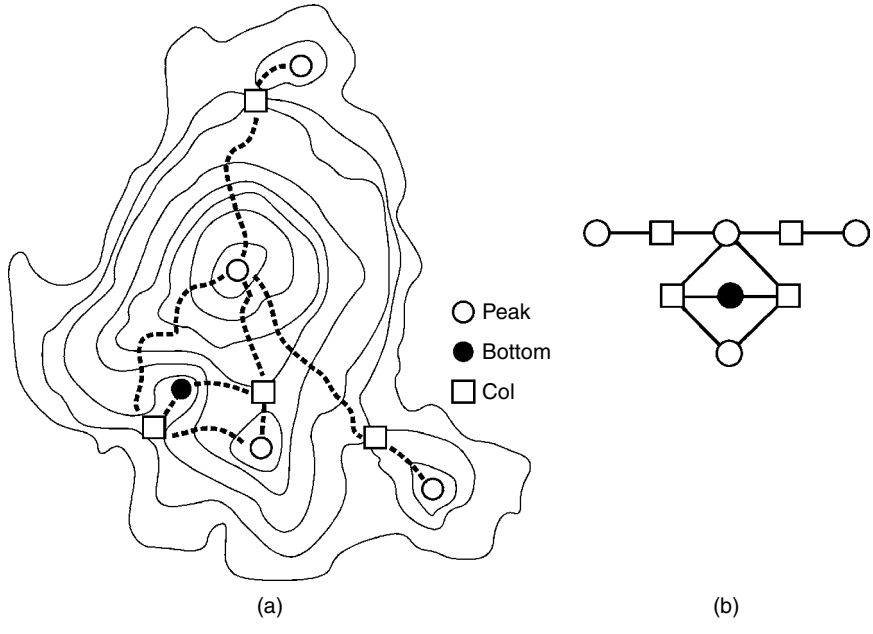

Figure 7.1 *(a) The population surface in Plate 3 represented by contour lines, its surface network (the broken lines) and (b) its surface graph*

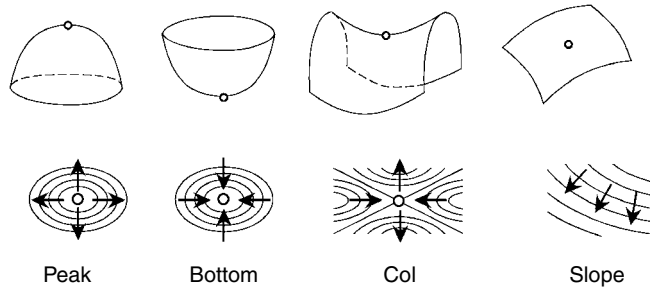

Figure 7.2 *A peak, a bottom, a col, and a slope*

is that the gradient at these points is flat. In Figure 7.1, peaks, bottoms and, cols are indicated by white circles, black circles, and white squares respectively. Since almost all local surfaces at points in S are slopes, critical points are very distinctive points. In the context of a mountain surface, we may regard the critical points as landmarks of mountain landscape. The configuration of these landmarks characterises the structure of the landscape.

To represent the configuration of critical points explicitly, we introduce a "surface network". To give an intuitive image of the "surface network", we use the analogy of water drops that flow on a mountain surface T. Suppose that infinitely many water drops fall at peaks of the surface T in S. The water drops flow in all directions from

the peaks and their trajectories cover the surface T. The trajectories of water drops have the following properties.

(i) The direction of the trajectory at a point is the steepest one among all possible directions at that point (this property corresponds to the fact that a water drop flows in the steepest direction).
(ii) The direction of the trajectory at a point is perpendicular to the contour line passing through the point (except at critical points).
(iii) At every col, exactly two trajectories go in and go out (Figure 7.2).

Among many trajectories on the surface T, we consider a set of trajectories each of which goes through a col and a critical point on T (the broken lines in Figure 7.1) except for trajectories each of which goes through a col and a point on the boundary of S. We call this set the *surface network* of T, and denote it by $N(T)$. As is noticed from Figure 7.1, the surface network $N(T)$ looks like a skeleton of the surface T. Actually, as will be discussed in the subsequent section, it is a skeleton or structure of the surface T.

The method that we propose in the next section uses surface networks extensively and so construction of surface networks is important in practice. Fortunately, nowadays, efficient computational methods have been developed and they are easily available. These methods are reviewed in Chapters 3 and 4 of this book.

7.3 STRUCTURAL SIMILARITY

To introduce the concept of structural or skeletal similarity, let us imagine a miniature model of a mountain surface T that is made of elastic materials (such as clay) placed on a rubber sheet S. We deform the surface T by stretching and shrinking the rubber sheet S without breaking it (mathematically, a homotopic deformation). Then the surface T in Figure 7.3(a) may be deformed into the surface T' in Figure 7.3(b). As a result, the surface network $N(T)$ of T (the broken lines in Figure 7.3(a)) changes to $N(T')$ (the broken lines in Figure 7.3(b)). Comparing these two surface networks, we feel that they are "structurally" similar, although they are different in details.

Furthermore, we deform the surface T', and then it may become the surface T'' in Figure 7.3(c). As a result, the surface network $N(T')$ (the broken lines in Figure 7.3(b)) changes to $N(T'')$ (the broken lines in Figure 7.3(c)). Comparing these two surface network, we feel that they are no more "structurally" similar.

To state the "structural" similarity (dissimilarity) explicitly, we introduce the concept of the "surface graph". Roughly speaking, a surface graph treats adjacency relations between critical points in a surface network. To be precise, consider a set, P, of nodes given by the critical points of an activity surface T, and a set, L, of links that correspond to line segments in the surface network $N(T)$ that directly joins two critical points. We call the paired sets, $\{L, P\}$, with their adjacency relations the *surface graph* of T, and denote it by $G(T)$. The surface graph of the activity surface T in Figure 7.1(a) is shown in Figure 7.1(b). The surface network $N(T)$ in Figure 7.1(a) can also be regarded as the surface graph of T. As is seen in these two examples, links

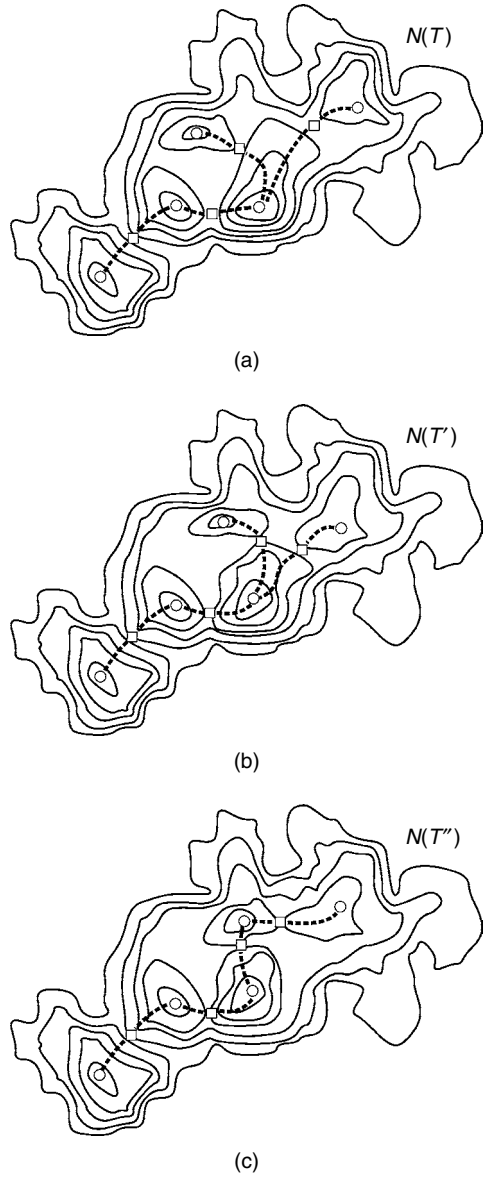

Figure 7.3 *Deformation of a surface*

in a surface graph are symbolic in the sense that they only show adjacency relations. Formally, two surface graphs,$\{L_1, P_1\}$ and $\{L_2, P_2\}$, of activity surfaces T_1 and T_2 are the same or isomorphic if, and only if, there is one-to-one correspondence between L_1 and L_2; between P_1 and P_2; and between adjacency relations of $\{L_1, P_1\}$ and those of $\{L_2, P_2\}$.

In terms of surface graphs, we can now define structural similarity explicitly. We say that two activity surfaces T and T' are *structurally similar* if, and only if, their surface

graphs $G(T)$ and $G(T')$ are isomorphic (the same); and the surfaces are *structurally dissimilar* if, and only if, their surface graphs are not isomorphic. For example, consider activity surfaces T, T' and T'' in Figure 7.3. Their surface graphs are shown in Figures 7.4(a), (b) and (c), respectively. As is noticed from these figures, the activity surfaces T and T' are structurally similar, but the activity surface T'' is structurally dissimilar to the activity surfaces T and T'.

The definition of structural similarity given above becomes contentious when we compare the activity surfaces shown in Figure 7.5. By definition, the activity surface T_1 is structurally dissimilar to the activity surface T_2. However, a peak in T_1 is so small that we may neglect it. Then the activity surface T_1 becomes structurally similar to the activity surface T_2.

To take this argument into account, we consider the relative height of a peak, p_i. A peak p_i is always adjacent to cols, $c_{i1}, c_{i2}, \ldots$ via links. Let $h(p_i)$ and $h(c_{ij})$ be the height of a peak p_i and that of a col c_{ij}, respectively. We define the relative height, $r(p_i)$, of a peak p_i by the minimum difference in height between the peak and the

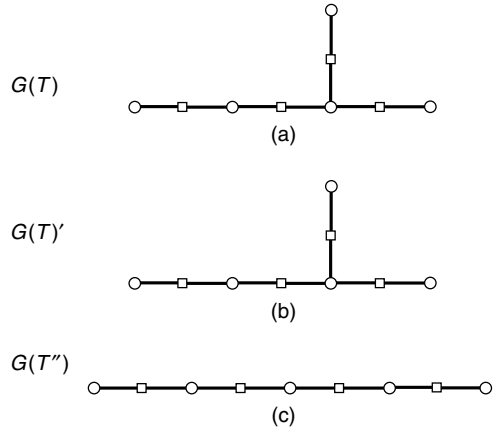

Figure 7.4 *The surface graphs of the activity surfaces T, T', T'' in Figure 7.3*

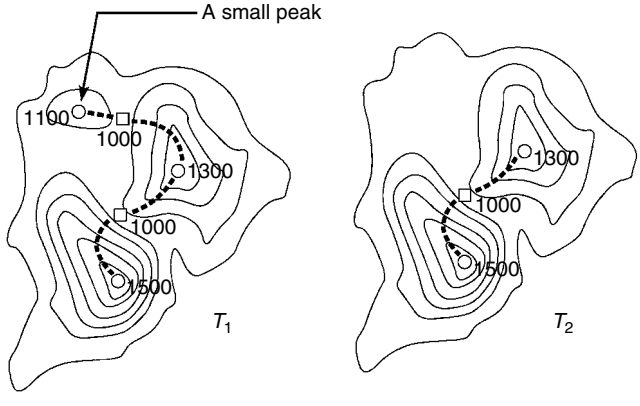

Figure 7.5 *Activity surfaces that are not exactly structurally similar but almost structurally similar*

adjacent cols, that is,

$$r(p_i) = \min_{j=1,2,\ldots} \{h(p_i) - h(c_{ij})\}. \tag{7.3}$$

For example, in Figure 7.6(a), the relative height of the peak p_2 is obtained from

$$\begin{aligned} r(p_2) &= \min\{h(p_2) - h(c_1), h(p_2) - h(c_2), h(p_2) - h(c_3), h(p_2) - h(c_4)\} \\ &= \min\{6337 - 1002, 6337 - 1609, 6337 - 680, 6337 - 355\} \\ &= 4728. \end{aligned}$$

Similarly, we define the depth (or height) of a bottom b_i by the minimum difference in height between the bottom and the adjacent cols, that is,

$$r(b_i) = \min_{j=1,2,\ldots} \{h(c_{ij}) - h(b_i)\}. \tag{7.4}$$

For example, in Figure 7.6(a), the relative depth (height) of the bottom b_1 is obtained from

$$\begin{aligned} r(b_1) &= \min\{h(c_3) - h(b_1), h(c_4) - h(b_1)\} \\ &= \min\{680 - 327, 355 - 327\} \\ &= 28. \end{aligned}$$

Let c_i^* be the chosen adjacent col, which we call a *base col*. For example, in Figure 7.6(a), the base col for the peak p_2 is c_2, and the base col for the bottom b_1 is c_4. As mentioned in Section 7.2, exactly four trajectories go through a base col in a surface network, and so at least two links but at most four links are incident to the base col (recall that a trajectory going to a point on the boundary of S is omitted in a surface graph; see the definition below properties (i) to (iii)). There are five possible combinations of critical points adjacent to a base col with respect to a critical point (a peak or a bottom) that is deleted. These combinations are shown in the left-hand side in Figure 7.7. Note that in Figure 7.7, a broken line may indicate one link or more than one link; and that two peaks adjacent to a base col may be the same. If we delete a peak or a bottom, the base col and links incident to the base col disappear and the adjacency relation changes. This change depends on the configuration of critical points. The results are shown in the right-hand side of Figure 7.7.

With the deletion rules shown in Figure 7.7, we delete peaks and bottoms whose relative heights are negligibly small in the following manner. As a first step, peaks and bottoms are sorted from the lowest relative height to the highest relative height, and the lowest peak (or bottom) is chosen. In the case of Toyama shown in Figure 7.7(a),

$$h(b_1) = 28 < h(p_1) = 43 < h(p_3) = 170 < h(p_4) = 1784 < h(p_2) = 4728.$$

Let h_1 be the lowest relative height ($h_1 = 28$ in Figure 7.6(a)). We delete the lowest peak (or bottom) with the rule in Figure 7.7 (in the case of Figure 7.6(a), the lowest relative height is the height (depth) of the bottom b_1, and the bottom is deleted with the rule in Figure 7.7(d)). As a result, we obtain a new surface graph and denote it by $G(T|h_1)$ (see Figure 7.6(b)).

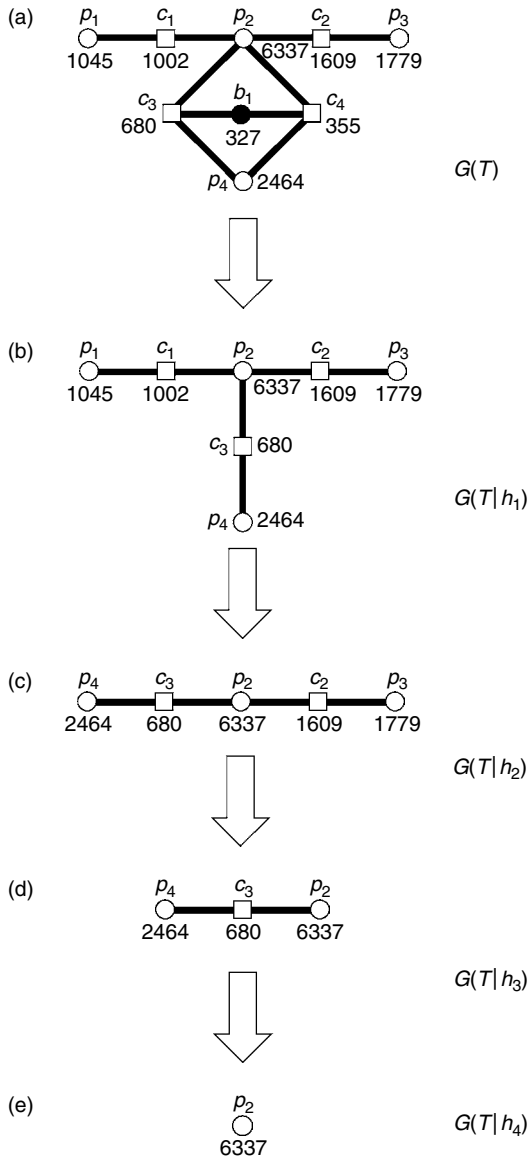

Figure 7.6 *Deleting peaks (bottoms) from the relatively lowest one, the second relatively lowest one and so forth*

In the second step, for the surface graph $G(T|h_1)$, peaks and bottoms are sorted from the lowest relative height to the highest relative height, and choose the lowest peak (bottom). In the example of Figure 7.6(b),

$$h(p_1) = 43 < h(p_3) = 170 < h(p_4) = 1784 < h(p_2) = 4728.$$

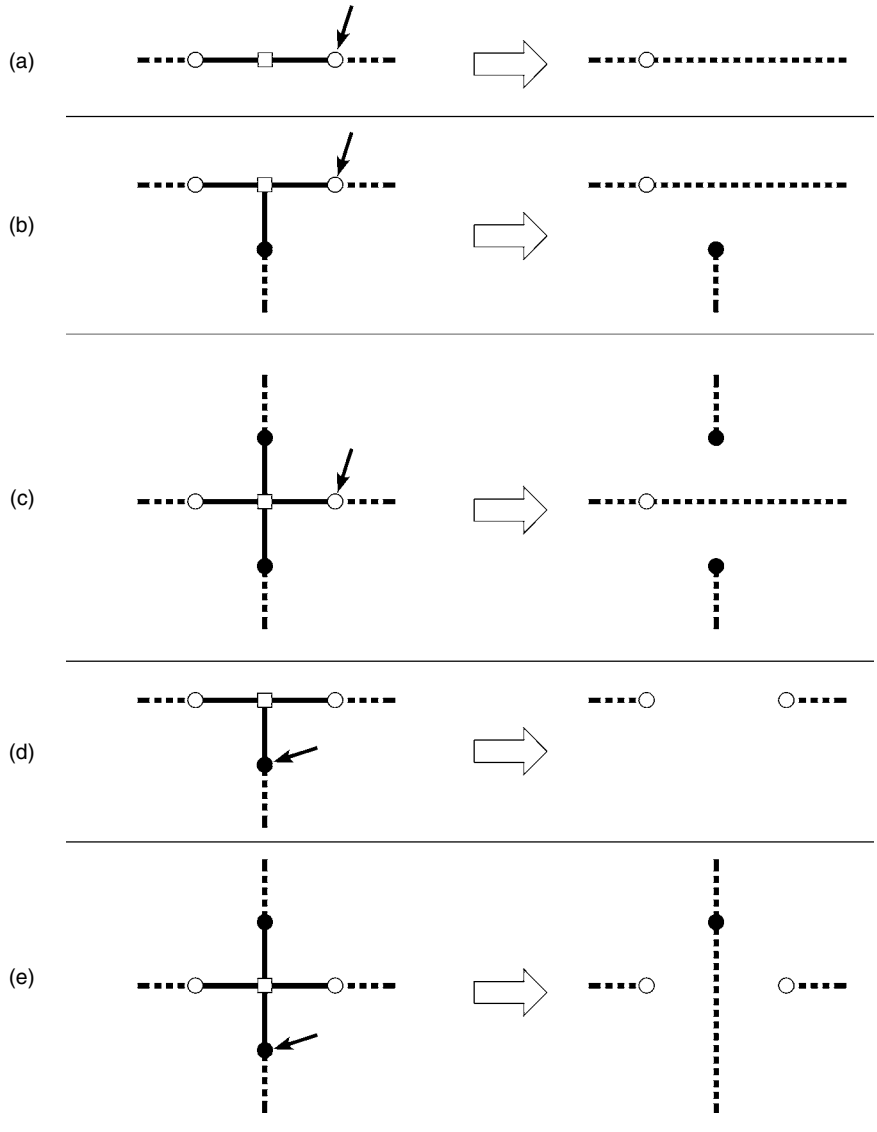

Figure 7.7 *Five types of adjacency relations at a base col and a critical point to be deleted (the left-hand side) and adjacency relations after the critical point is deleted (the right-hand side)*

Let h_2 be the lowest relative height ($h_2 = 43$ in Figure 7.6(b)). We delete the relatively lowest peak (or bottom) with the rule in Figure 7.7 (in the case of Figure 7.6(b), the lowest relative height is the height of the peak p_1, and the peak is deleted with the rule in Figure 7.7(a)). As a result, we obtain the surface graph $G(T|h_2)$ (Figure 7.6(c)).

In the third step, for the surface graph $G(T|h_2)$, we do the same tasks as in the first and second steps, and so forth. As this procedure continues, the number of peaks and bottoms decreases, and eventually there is only one peak left (Figure 7.6(e)).

Obviously, if we choose h, which is higher than the height of the last remaining peak, we have no peak.

We now regard the values $h_1, h_2, \ldots, h_m$ as specific values of a parameter h, and define $G(T|h)$ as

$$G(T|h) = G(T|h_i) \text{ for } h_i \leq h < h_{i+1} \quad i = 0, 1, \ldots m-1. \tag{7.5}$$

Note that $h_0 = 0$, and $G(T|0) = G(T)$. For two activity surfaces T_1 and T_2, we say that these surfaces are *structurally similar at level* h if and only if $G(T_1|h)$ is isomorphic to $G(T_2|h)$. Suppose that peaks and bottoms whose height is less than h^* are negligible. Then we examine structural similarity among activity surfaces at level h^*. In the example of Figure 7.5, two activity surfaces are structurally similar at level $h^* = 100$.

Note that the equivalence relation "T_i is structurally similar to T_j at h" can be used for classifying activity surfaces into a set of categorical groups, $G_1^h, \ldots, G_{n(h)}^h$, where any two activity surfaces in G_i^h are structurally similar at level h, and an activity surface in G_i^h and an activity surface in G_j^h $(i \neq j)$ are structurally dissimilar at level h. An actual example will be shown in Section 7.5.

7.4 OVERALL STRUCTURAL SIMILARITY INDEX

Although we can avoid trifling peaks and bottoms using structural similarity at level h^*, choice of level h^* is arbitrary. To avoid this arbitrariness, we propose the following index. Let

$$J(T_1, T_2|h) = \begin{cases} 1 \text{ if } G(T_1|h) & \text{is isomoprhic to } G(T_2|h), \\ 0 & \text{otherwise.} \end{cases} \tag{7.6}$$

This means that $J(T_1, T_2|h)$ becomes 1 if the activity surfaces T_1 and T_2 are structurally similar at level h; otherwise, it becomes 0. An actual example is shown in Figure 7.8 (Hiratsuka and Kawagoe), where the values of 1 and 0 are indicated by heavy-line segments and hairline segments, respectively, in the range of $0 \leq h \leq h^*$. Obviously, if the length of the heavy-line segments is long, the two activity surfaces are structurally similar.

In terms of $J(T_1, T_2|h)$, we define an index I as

$$I = \frac{1}{h_{\max}} \int_0^{h_{\max}} J(T_1, T_2|h) \mathrm{d}h. \tag{7.7}$$

This index refers to the standardised length of the interval of h where two activity surfaces T_1 and T_2 are structurally similar. When two activity surfaces T_1 and T_2 are structurally similar at any level of h, then $I = 1$; when two activity surfaces T_1 and T_2 are not structurally similar at any level at all, then $I = 0$. We call this index the *overall structural similarity index*.

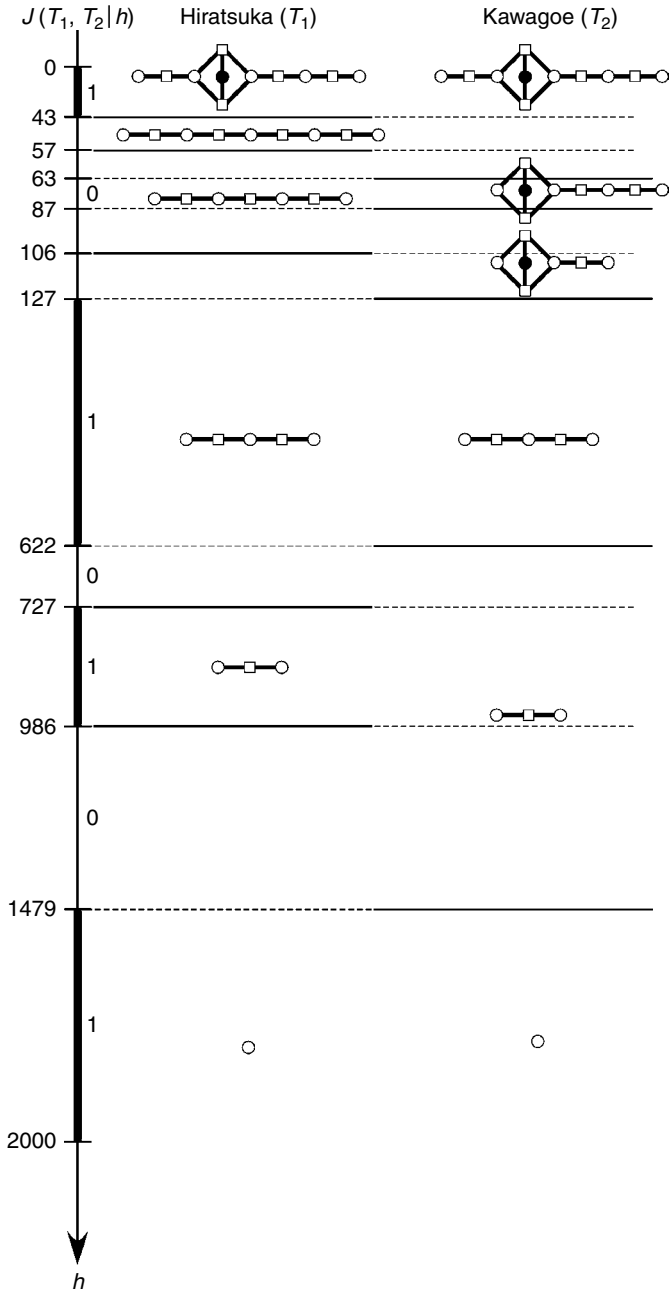

Figure 7.8 *The function $J(T_1, T_2|h)$ for the case of Hiratsuka (T_1) and Kawagoe (T_2)*

7.5 COMPARATIVE STUDY ON URBAN POPULATION DISTRIBUTIONS IN JAPAN

Having established the method for structural comparison of activity surfaces in the preceding sections, we now apply it to a comparative study of population densities in Japan. We choose 20 Japanese cities whose population sizes are in between 200,000 and 400,000. The data set used for this analysis is the 1-km by 1-km "mesh-data" (grid-data) published from the Japanese Census Bureau in 1998.

First, using the algorithms proposed by Takahashi et al. (1995) (also see Chapter 3 in this book), we construct surface networks $N(T_1), \ldots, N(T_{20})$. Second, we obtain surface graphs $G(T_1|0), \ldots, G(T_{20}|0)$ from these surface networks. The results are depicted in the second column of the table in Figure 7.9. Third, following the rules in Figure 7.7, we delete peaks (or bottoms) from the lowest relative height to the highest relative height and obtain $G(T_1|h), \ldots, G(T_{20}|h)$ for $0 \leq h \leq 2000/\,\mathrm{km}^2$. A part of the results is shown in Figure 7.9.

Suppose that we delete the peaks and bottoms whose relative height is less than $500/\,\mathrm{km}^2$. Then from Figure 7.9, the 20 cities are classified in four categorical groups:

$$G_1^{500} = \{\text{Aomori, Akita, Fukushima, Hakodate, Maebashi, Morioka, Tokorozawa}\}$$

$$G_2^{500} = \{\text{Kashiwa, Kasugai, Koshigaya, Miyazaki, Takamatsu, Takasaki, Toyama}\}$$

$$G_3^{500} = \{\text{Hiratsuka, Kawagoe, Kohchi, Odawara, Yokkaichi}\}$$

$$G_4^{500} = \{\text{Fujisawa}\}$$

We notice from this result that all cities have a tree structure (no loops), and that the number of cities is the largest for one or two peaks but it decreases as the number of peaks increases.

Fourth, using the functions $G(T_1|h), \ldots, G(T_{20}|h)$ for $0 \leq h \leq 2000/\,\mathrm{km}^2$, we calculate the overall structural similarity indexes between every pair of the twenty Japanese cities. The indexes are tabulated in Table 7.1.

To grasp the overall structural similarity among the 20 cities visually, we apply the (MDS) multi-dimensional scaling method to the distance matrix made from 1 minus the index value in each element in Table 7.1. The result is shown in Figure 7.10 (the square marks indicate cities).

In Figure 7.10, the 11 square marks that are close together around (1, 0) are remarkably eye-catching. The common feature of the population distribution structures of these cities is identified from Figure 7.9. The surface graphs of the 11 cities have up to two peaks at level $500/\,\mathrm{km}^2$; the surface graphs of the cities except for that of Odawara have one peak at level $750/\,\mathrm{km}^2$; and the surface graphs of all the cities have one peak at level $1000/\,\mathrm{km}^2$. In short, the trifle critical points, except for one peak, on the surface graphs of these cities are eliminated at relatively low levels.

Focusing on the left side of Figure 7.10, we notice that the square marks for Fujisawa, Kasugai, Kashiwa, and Toyama take low horizontal coordinate values compared with the other square marks. From Figure 7.9, we can identify the feature of the surface graphs of these four cities. Their surface graphs commonly have two peaks at the level $1750/\,\mathrm{km}^2$. As for the surface graphs of Fujisawa, Kasugai, and Kashiwa, they

Cities \ h	0	250	500	750	1000	1250	1500	1750	2000
Aomori		○	○	○	○	○	○	○	○
Akita		○	○	○	○	○	○	○	○
Fujisawa									
Fukushima			○	○	○	○	○	○	○
Hakodate		○	○	○	○	○	○	○	○
Hiratsuka					○	○	○	○	○
Kashiwa									
Kasugai									
Kawagoe							○	○	○
Koshigaya						○	○	○	○
Kohchi						○	○	○	○
Maebashi			○	○	○	○	○	○	○
Miyazaki				○	○	○	○	○	○
Morioka			○	○	○	○	○	○	○
Odawara					○	○	○	○	○
Takamatsu				○	○	○	○	○	○
Takasaki				○	○	○	○	○	○
Takorozawa		○	○	○	○	○	○	○	○
Toyama									○
Yokkaichi						○	○	○	○

Figure 7.9 *The surface graphs $G(T_i|h)$ of 20 cities in Japan with respect to $h = 0, 250, \ldots, 2000$*

have two peaks even at the highest level (2000/km^2). This is quite different from the feature of the surface graphs of the aforementioned 11 cities. This difference creates the large differences in the locations of the square marks between these four cities and the aforementioned 11 cities.

Among the four cities, or rather the 20 cities, Fujisawa is a distinctive city in the sense that the square mark for this city takes an extremely high vertical coordinate value. As can be seen in Figure 7.9, the number of peaks on the surface graph of Fujisawa is larger than those of the other cities at any levels between 500/km^2 and 1250/km^2. The differences between the location of the square mark for Fujisawa

Table 7.1 *Overall structural similarity indexes between every pair of the 20 Japanese cities*

	Aomori	Akita	Fujisawa	Fukushima	Hakodate	Hiratsuka	Kashiwa	Kasugai	Kawagoe	Koshigaya
Aomori	1	–	–	–	–	–	–	–	–	–
Akita	0.7575	1	–	–	–	–	–	–	–	–
Fujisawa	0	0	1	–	–	–	–	–	–	–
Fukushima	0.7635	0.8525	0	1	–	–	–	–	–	–
Hakodate	0.728	0.953	0	0.8525	1	–	–	–	–	–
Hiratsuka	0.507	0.507	0	0.537	0.507	1	–	–	–	–
Kashiwa	0.055	0	0.2575	0.0455	0	0.1615	1	–	–	–
Kasugai	0.2485	0.0245	0.2575	0.0455	0.0075	0.1295	0.773	1	–	–
Kawagoe	0.2605	0.2605	0	0.2905	0.2605	0.6375	0.4605	0.4285	1	–
Koshigaya	0.5825	0.4975	0	0.526	0.4975	0.778	0.3275	0.322	0.5755	1
Kohchi	0.5075	0.4925	0	0.4925	0.4925	0.856	0.127	0.095	0.586	0.699
Maebashi	0.827	0.7675	0	0.818	0.7675	0.555	0.0055	0.109	0.3085	0.576
Miyazaki	0.877	0.74	0	0.8015	0.7475	0.578	0.0385	0.155	0.3075	0.603
Morioka	0.7715	0.7915	0.009	0.7915	0.7915	0.535	0	0.0535	0.2885	0.537
Odawara	0.837	0.885	0	0.8905	0.851	0.845	0	0.1305	0.2605	0.4975
Takamatsu	0.8055	0.6805	0.0255	0.6805	0.6805	0.549	0.0925	0.1625	0.311	0.67
Takasaki	0.9535	0.774	0	0.785	0.7495	0.507	0.0335	0.219	0.2605	0.561
Tokorozawa	0.718	0.9445	0	0.9075	0.92	0.507	0	0.0535	0.2605	0.4975
Toyama	0.1715	0.109	0.1495	0.1535	0.109	0.2645	0.665	0.9275	0.553	0.425
Yokkaichi	0.494	0.4635	0.2125	0.4635	0.4635	0.6075	0.285	0.285	0.5055	0.7145

	Kohchi	Maebashi	Miyazaki	Morioka	Odawara	Takamatsu	Takasaki	Tokorozawa	Toyama	Yokkaichi
Aomori	–	–	–	–	–	–	–	–	–	–
Akita	–	–	–	–	–	–	–	–	–	–
Fujisawa	–	–	–	–	–	–	–	–	–	–
Fukushima	–	–	–	–	–	–	–	–	–	–
Hakodate	–	–	–	–	–	–	–	–	–	–
Hiratsuka	–	–	–	–	–	–	–	–	–	–
Kashiwa	–	–	–	–	–	–	–	–	–	–
Kasugai	–	–	–	–	–	–	–	–	–	–
Kawagoe	–	–	–	–	–	–	–	–	–	–
Koshigaya	–	–	–	–	–	–	–	–	–	–
Kohchi	1	–	–	–	–	–	–	–	–	–
Maebashi	0.5355	1	–	–	–	–	–	–	–	–
Miyazaki	0.5225	0.8785	1	–	–	–	–	–	–	–
Morioka	0.5205	0.821	0.788	1	–	–	–	–	–	–
Odawara	0.4925	0.799	0.779	0.8195	1	–	–	–	–	–
Takamatsu	0.5345	0.7645	0.789	0.7085	0.6805	1	–	–	–	–
Takasaki	0.4925	0.8485	0.8895	0.793	0.862	0.784	1	–	–	–
Tokorozawa	0.4925	0.7675	0.7345	0.7915	0.899	0.6805	0.8235	1	–	–
Toyama	0.2075	0.2265	0.263	0.1615	0.1925	0.2705	0.2835	0.113	1	–
Yokkaichi	0.5585	0.4635	0.4775	0.4635	0.4635	0.5315	0.4725	0.4635	0.393	1

and those for other cities are due to this specific feature of the surface graph of Fujisawa.

On the basis of these results, we can further investigate why these cities are structurally similar or dissimilar. An answer to this question will be given by examining the relationship between the distribution of population and the distributions of other attribute values of these cities. This examination, however, is beyond the scope of this chapter. An example is shown in (Okabe and Masuda, 1984).

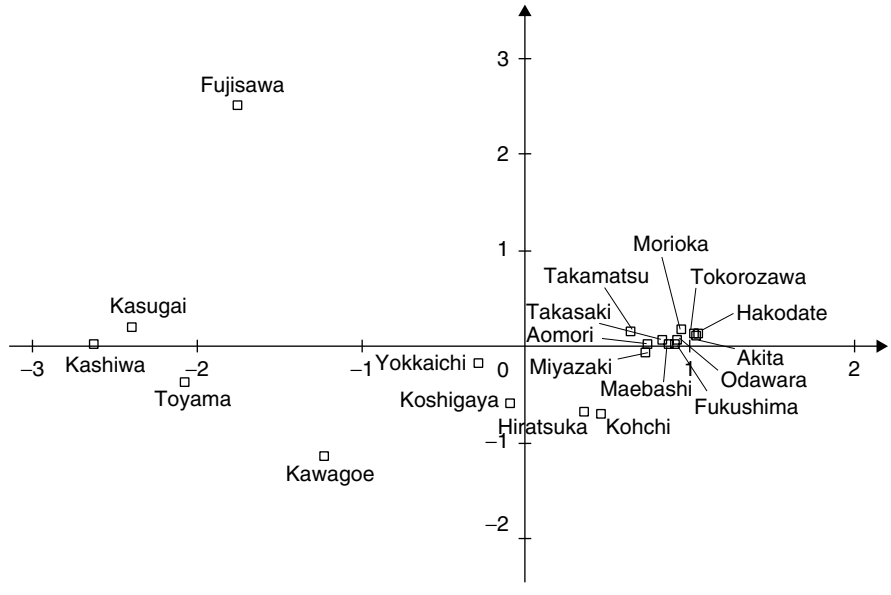

Figure 7.10 *The multi-dimensional scaling for the 20 cities in Japan*

7.6 CONCLUDING REMARKS

Summing up, we proposed in this chapter a new method for studying structural similarity among activity surfaces. First, we represented an activity surface by a surface network and defined structural similarity between two activity surfaces in terms of isomorphism between their surface graphs. Second, we defined structural similarity with respect to relative height (depth), and measured the overall structural similarity in terms of the index I defined by equation (7.7). As shown in Section 7.5, the proposed methods are useful for a comparative analysis of activity surfaces. In particular, the proposed method clearly reveals structural similarities and differences among activity surfaces.

Last, we note that although this chapter focused on population surfaces, the proposed method is so general that they may be applied to activity surfaces treated in the humanities and social sciences. We look forward to such applications.

ACKNOWLEDGEMENTS

We express our thanks to T. Kaneko for pre-processing data, and Y. Sadahiro and S. Rana for their valuable comments on an earlier draft of this chapter.

8

Topology Diagrams of Scalar Fields in Scientific Visualisation

Valerio Pascucci

8.1 INTRODUCTION

The purpose of visualisation is to aid the user in understanding the structure of the data (Tufte, 1983). Common scientific visualisation methods can be grouped into two broad classes. First are those methods whose aim is to detect structures and to present their display to the user. Critical to these methods is the definition of "structure", and how well such a definition matches the user's need. Second are those methods attempting to display the entire scalar field simultaneously, leaving to the user the interpretation of the rendered images. The combined use of these two types of methods helps to reinforce the information provided by their visualisation. We discuss in detail the computation of one method in the first class that computes and presents simple diagrams of the topology of a scalar field. Practical results are shown for a combination of topology and colour mapping in 2D and topology and isocontouring in 3D.

Fundamental work in scientific visualisation has dealt with determining *good* colour maps that effectively display the structures present in the data. Bergman et al. define rules on the basis of perception, user goals, and data characteristics to automatically select a colour map that will meet the user's requirements (Bergman et al., 1995). Histogram equalisation is a technique that spreads the data evenly over the range of colours, using the available colour space to its fullest (Rosenfeld and Kak, 1982).

This work was performed under the auspices of the US Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. UCRL-JC-139277"

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

The result is that each colour in the colour map is used an equal number of times. Gershon (Gershon, 1992) uses *Generalised Animation* to display an otherwise static scalar data in a dynamic way. Taking advantage of the ability of the visual system to detect dynamic changes, the animation draws attention to fuzzy details in the data, which may not be detected in a static representation.

Computation and display of level sets (Lorensen and Cline, 1987) is another fundamental technique used in the visualisation of scalar field. A large number of papers have been published to resolve several issues ranging form the computation of surfaces with correct topology (Lopes and Brodlie, 2003) to the computation of isosurface with optimal performance (Bajaj et al., 1996) and minimum storage overhead (van Kreveld et al., 1997). A good review of the subject can be found in (Bajaj et al., 1999).

Topological techniques are gaining importance in geometric modelling and visualisation (Fomenko and Kunii, 1997). Helman and Hesselink detect vector field topology by classifying the zeros of a vector field and performing particle tracing from saddle points (Helman and Hesselink, 1991). The resulting partitioning consists of regions with uniform flow. Globus et al. describe a software system for 3D vector topology and note that the techniques used can also be applied to the gradient flow of a scalar field (Globus et al., 1991). Bader et al. (1979) and Collard and Hall (1977) examine the gradient field of the charge density in a molecular system. The topology of this scalar field represents the bonds linking together the atoms of the molecule. Bader goes on to show how higher-level structures in the topology represent chains, rings, and cages in the molecule. Bader's example is a defining motivation for developing the automatic extraction and visualisation of topology from a scalar field, since it is one of many situations in which topology provides an intuitive and physically meaningful visualisation.

The contour tree (Pascucci and Cole-McLaughlin, 2002) is a related, non-embedded diagram that is becoming popular as a user-interface component for a better understanding of the structure and topology of the level sets in a scalar field. The embedded topology diagrams discussed in this chapter have been introduced in (Bajaj et al., 1998) for the smooth case. Recent advances (Edelsbrunner et al., 2003a) allow performing a more comprehensive analysis of a piecewise linear scalar field by building the Morse Complex of a scalar field, which consistently partitions the data in regions with flow lines having equal endpoints. More work is needed before this approach will be ready for practical use with large scientific datasets. In particular, one major aspect that needs further investigation is the development of a full multi-resolution topological model similar to the one proposed in (Bremer et al., 2003) for the 2D case. For this reason, we treat the 3D case with approximations that maintain the efficiency and simplicity of the 2D approach without losing its visual effectiveness in the 3D embedding.

8.2 DEFINITIONS

8.2.1 PL scalar fields

We consider the case of n-dimensional spaces $\mathbb{R}^n$, with $n = 2, 3$. A point p is a sequence of n real numbers $p = (x_1, \ldots, x_n)$. A point p is said to be an affine combination of the k points $\{p_1, \ldots, p_k\}$, if there is a set of k real numbers $\{a_1, \ldots, a_k\}$ such that $p = a_1 p_1 + \cdots + a_k a_k$ and $a_1 + \cdots + a_k = 1$. If, additionally, all the a_i are non-negative, the point p is said to be a convex combination of $\{p_1, \ldots, p_k\}$. A d-simplex

σ, with $0 \leq d \leq n$, is the set of points obtained by convex combination of $d+1$ independent points $(v_0, \ldots, v_d)$, called *vertices* of σ. A simplex σ_i is a face of σ_j – denoted $\sigma_i \prec \sigma_j$ – if all the vertices of σ_i are also vertices of σ_j. Vertices are 0-simplices, edges are 1-simplices, triangles are 2-simplices, and tetrahedra are 3-simplices. A simplicial complex K is a collection of h distinct simplices $\{\sigma_1, \ldots, \sigma_h\}$ such that the following two conditions are satisfied: (i) for any simplex $\sigma_j \in K$, all the faces $\sigma_i \prec \sigma_j$ are also in K and (ii) for any two simplices $\sigma_i, \sigma_j \in K$, their intersection $\sigma = \sigma_i \cap \sigma_j$ is either empty or a face of both: $\sigma = \sigma_i \cap \sigma_j \neq \emptyset \Rightarrow \sigma \prec \sigma_i, \sigma \prec \sigma_j$. The support space $|K|$ is the point set union of the simplices in K.

Consider a real-valued function f, defined for all the points of a domain D. A scalar field F is the pair (f, D). We assume that D is the support space of a complex K, and that f is defined explicitly only at the vertices of K. In the interior of a d-simplex, the function f is defined by linear interpolation of its values at the vertices. Specifically, the value of the field at p is $f(p) = a_0 f(v_0) + \cdots + a_d f(v_d)$, where $(v_0, \ldots, v_d)$ are the vertices of the simplex containing p, and the positive real numbers $(a_0, \ldots, a_d)$ are such that $p = a_0 v_0 + \cdots + a_d v_d$ and $a_1 + \cdots + a_d = 1$. Note that any point $p \in D$ is either a vertex or is contained in the interior of a simplex in K. Therefore, f is a continuous function defined in all D.

We assume that the function values at the vertices of K are all distinct, that is $i \neq k \Rightarrow f(v_i) \neq f(v_j)$. Enforcing this notion is crucial to providing sound mathematical definitions and robust algorithms. To enforce this assumption of *simplicity* (Edelsbrunner and Mücke, 1990), we define inequality tests that break ties by comparing the indices of the vertices. Whenever $f(v_i) = f(v_j)$, the test $f(v_i) < f(v_j)$ is replaced by $i < j$ and the test $f(v_i) > f(v_j)$ is replaced by $i > j$.

8.2.2 Critical points

We use Morse theory (Milnor, 1963) to characterise the scalar field F in terms of its gradient flow $\nabla f = [\partial f/\partial x^1, \ldots, \partial f/\partial x^n]^{\mathrm{T}}$. For a smooth function f, a point is called *critical* if $\nabla f = 0$. A critical point is *simple* if the Hessian $\nabla^2 f = (\partial^2 f/\partial x^i \partial x^j)$ is non-singular. Figure 8.1(a) shows a simple critical point, which must be isolated. Figures 8.1(b) and (c) show *degenerate* critical points with multiplicity two and three respectively. Figure 8.1(d) shows a set of degenerate critical points forming a submanifold of the domain D. The index of a non-degenerate critical point is defined as the number of negative eigenvalues of its Hessian.

The notions defined in Morse theory for smooth functions are extended to a piecewise linear field F defined on the support of a simplicial complex K. The *link* of vertex $v \in K$, denoted $lk(v)$, is the set of simplices not containing v that are faces of a simplex containing v. The upper link $\overline{lk}(v)$ is the set of simplices in $lk(v)$ with field value entirely greater than $f(v)$. The lower link $\underline{lk}(v)$ is the set of simplices in $lk(v)$ with field value entirely smaller than $f(v)$.

In general, we assume that the domain D is an n-manifold with boundary, decomposed into strata as described in (Goresky and MacPherson, 1988). The interior D of the domain contains all the n-dimensional critical points, while the boundary contains only $(n-1)$-dimensional critical points. In 1D, a critical point can only be a maximum

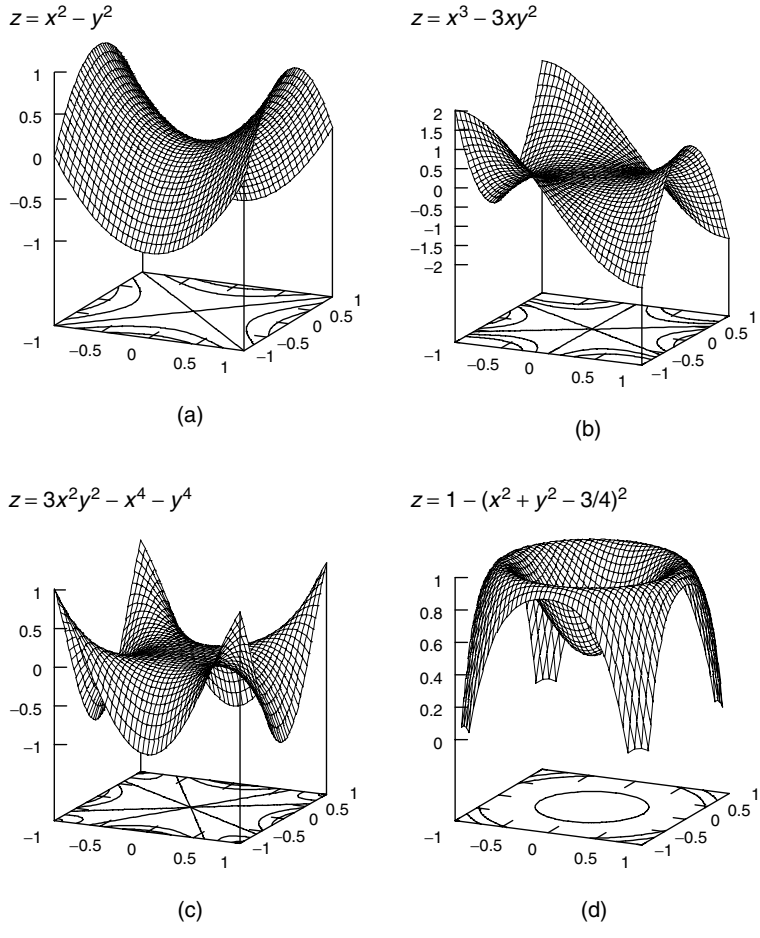

Figure 8.1 *Four functions defined in* $\mathbb{R}^2$. *The top row show their analytic representation* $z = f(x, y)$. *The bottom row shows their 3D plot together with contour lines projected on the* (x, y) *plane. The first three functions have a critical point of coordinates (0, 0). (a) A simple critical point; (b) a double critical point, also known as "monkey saddle"; (c) a triple critical point; and (d) set of critical points forming a 1-manifold (circle of radius* $\sqrt{3}/2$)

or a minimum. Figure 8.2 shows the classification of a critical point in 2D and in 3D based on the connectivity of its lower/upper link as follows:

1. a minimum has empty lower link;
2. a maximum has empty upper link;
3. a regular point has both lower and upper link made of one connected component;
4. any other point is a saddle.

In 2D, the number of components of the lower link is equal to the multiplicity of the saddle plus one. In 3D, the characterisation of multi-saddles is more complex. Intuitively, each independent annulus (loop) in the lower link corresponds to a

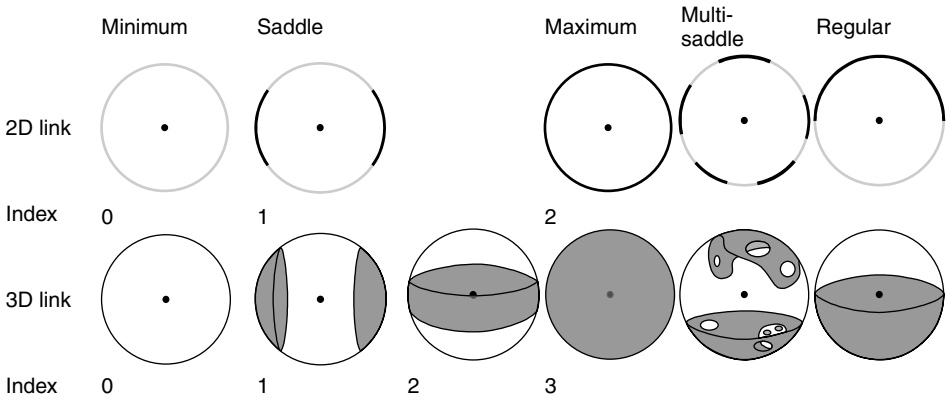

Figure 8.2 *The link of a point p in the interior of D is represented by a circle in* $\mathbb{R}^2$ *and by a sphere in* $\mathbb{R}^3$*. A simple critical point, a degenerate multi-saddle, and a regular point can be classified on the basis of the connectivity of their lower link (black portion of the link in 2D and opaque portion of the link in 3D)*

2-saddle and each connected component (except one) corresponds to a 1-saddle. An elegant characterisation of 3D multi-saddles based on reduced Betti numbers is provided in (Edelsbrunner et al., 2003a), together with a procedure for their decomposition into simple saddles. For visualisation purposes, we only differentiate among regular points, maxima, minima, saddles and therefore, classify them simply by counting the connected components of their lower and upper links as described above. This scheme applies directly to a piecewise linear (PL) field since it does require the function to be smooth.

8.3 TOPOLOGY DIAGRAMS

We compute the steepest descending/ascending lines starting from the saddle points and draw the resulting embedded graph to provide the user with intuitive information correlating the critical points in the data. It is also possible to visualise this information at different levels of resolution using cancellation of critical points in their order of importance.

8.3.1 2D steepest descending/ascending paths

Given a saddle point p in a 2D scalar field, we compute one steepest descending path in each connected component of its lower link. Consider the triangulated component of the lower link in Figure 8.3(a). For each candidate edge (p, v_i), with $f(v_i) < f(p)$, we compute the magnitude of its gradient flow $|\nabla f| = |p - v_i|/(f(p) - f(v_i))$. Similarly, for a candidate triangle (p, v_i, v_j), with $f(v_i), f(v_j) < f(p)$, we compute the magnitude of its gradient flow $|\nabla f|$, where the gradient is given by

$$\nabla f = \begin{bmatrix} v_i - p \\ v_j - p \end{bmatrix}^{-1} \begin{bmatrix} f(v_i) - f(p) \\ f(v_j) - f(p) \end{bmatrix}$$

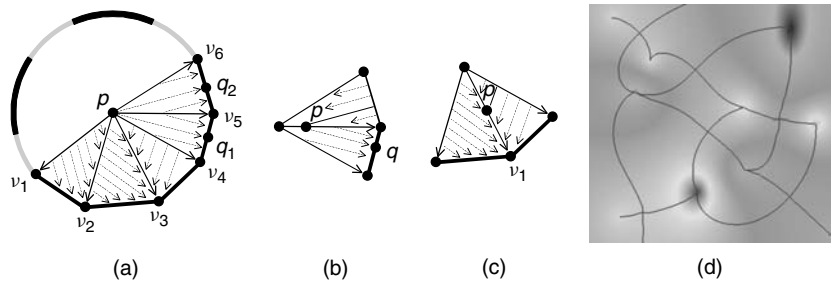

Figure 8.3 *One step of steepest descending path for a 2D scalar field. (a) Starting configuration at a saddle point. In the triangulated component of the lower link, there are three candidate gradient directions: the edge* (p, v_3) *and the gradient lines* (p, q_1) *and* (p, q_2)*. The one with highest slope is chosen to start the descending path from* p*. Note that this is also the configuration for a regular vertex; (b–c) steepest descending path extended from a point* p *in the middle of an edge; (d) network of steepest descending and ascending paths for a simple test function*

The gradient directions are depicted with dashed arrows. Locally, there are three segments of steepest descent: (p, v_3), (p, q_1), and (p, q_2). Comparing their three gradient magnitudes one determines the actual steepest descent, which forms the first portion of the descending path from p. The path is then extended repeatedly until a local minimum is reached. During this iterative procedure, the path can be extended from a saddle, from a regular vertex, or from a point in the middle of an edge. The same rule of the saddles applies to the case of a regular vertex. If the current end point of the path lies on an edge of the mesh, one needs to check the slope of the gradient along the edge and within the next adjacent triangle. Different configurations of this is case are shown in Figure 8.3(b–c).

In the PL case, as against the smooth case, descending gradient lines can merge. Once the current descending path merges with a previous descending path, they will not split because of the consistency of the local choice. After all the descending paths are computed, we construct the ascending paths with the same procedure applied to the field $-F$. In this case, we run into another difference compared with the smooth case since the PL ascending paths can intersect the descending paths. To avoid this problem and guarantee consistency in the construction, we make an exception to the iterative procedure above. We follow the line of a previously computed descending path (in the opposite direction) if the local choice of steepest ascending path leads to an intersection.

Figure 8.3(d) shows the topology diagram computed for a simple test function. The ascending and descending paths are drawn on top of the 2D field depicted with a grayscale colour map. Plate 4 demonstrates the use of the topology diagram to complement classical pseudo-colouring visualisation for a density field in an off-axis pion collision simulation (data courtesy of Lawrence Livermore National Laboratory). The visualisation in the top row uses a simple grayscale colour map. In this case, it is clear that much of the area of interest in the field is washed out. The visualisation in the second row uses a hue-based colour map varying from blue (low) to red (high), revealing more of the structure in the data. The visualisation in the third row is augmented with the topology diagram. This addition clearly brings out the detailed structure of the density field.

8.3.2 Persistence

While small-scale features are important in many scientific applications, in some circumstances, the visualisation user is interested only in the large-scale structure. For this situation, we apply a simple filter that is based on the concept of persistence. The basic idea is to remove pairs of critical points via topological cancellations. For example, in a 1D field, a local maximum can be cancelled with an adjacent local minimum. Figure 8.4(a–b) shows this type of cancellation, where the persistence associated with the pair of critical points (p, q) is ranked by the persistence defined as $|f(p) - f(q)|$. Figure 8.4(c–d) shows the cancellation of the maximum p with the saddle q. For a 2D field, all the cancellations involve a saddle and an extremum. This type of simplification can be used to generate a complete multi-resolution representation and re-meshing of the data. Discussion of this topic is beyond the scope of this chapter, and the interested reader is referred to (Bremer et al., 2003) for further details.

The bottom row of Plate 4 shows the use of this technique in the case of the pion topology. The large-scale structures in the data are preserved while the fine grain topological noise is filtered out. Simplification by persistence is a fundamental tool for the practical use of topological visualisation. In fact, in many cases the amount of topological noise can be so large that its direct display would cover the entire image and make the whole approach useless.

8.3.3 3D extension

A complete generalisation of the 2D scheme to the volumetric case is very challenging and computationally expensive. One main problem is the resolution of multi-saddles. To resolve multi-saddles in 2D, one needs only to count the number of components of the lower link. In 3D, one should instead determine the full Betti numbers of the lower and upper link. More importantly, in 2D, the choice of descending paths is easily made consistent by simply avoiding intersections. In 3D, the inability to compute the Hessian at a saddle becomes a major obstacle to the computation of consistent monotonic paths since they should start along the direction of its eigenvectors. There are paths connecting 1-saddles to 2-saddles that, at the current state of the art, can be determined

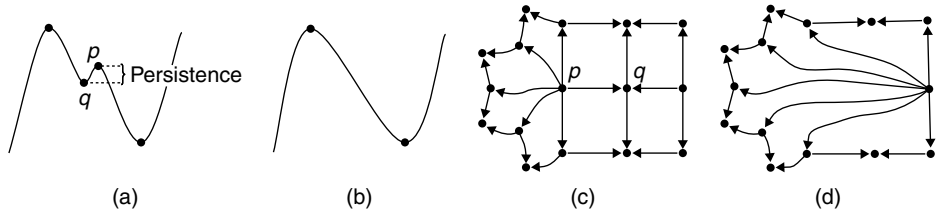

Figure 8.4 *Simplification of the topology of a scalar field by cancellation of pairs (p, q) of critical points. The persistence of each pair is the difference of their field value. (a) Two critical points of a 1D field with low persistence; (b) simplified version of the same 1D field; (c) topology diagram of a 2D scalar field in the neighbourhood of two critical points with low persistence. The arrows show descending directions and (d) topology diagram of the same field after cancellation of the two critical points*

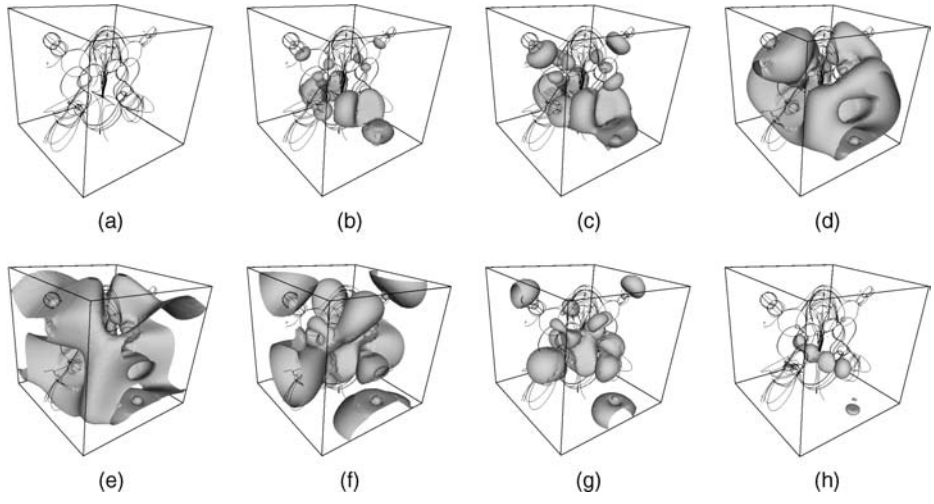

Figure 8.5 *Topology diagram of the 3D wave function computed for a high-potential iron protein. (a) Topology diagram and (b–h) combined visualisation of the topology diagram together with a sequence of level sets with increasing isovalue*

only from the computation of the full Morse Complex (Edelsbrunner et al., 2003a). In this context, we choose a simplified approach in which no distinction is drawn among the saddles of different index. Since in each loop of the lower/upper link one should choose two steepest descending paths, we trace paths from all the local maxima of the gradient direction and remove duplicates that lead to the same extremum. Noise removal is also achieved by generic extremum-saddle cancellations, even if formally a minimum should be paired only with a 1-saddle, a maximum should be paired only with a 2-saddle, and additionally one should also allow (1-saddle, 2-saddle) pairs. Figure 8.5 shows the visualisation of a 3D wave function computed for a high-potential iron protein (data courtesy of the Visualization lab, SUNY – Stony Brook). Figure 8.5(a) is the filtered topology diagram, which highlights the main topological features in the data. Figure 8.5(b–h) shows the same diagram together with a sequence of levels sets (isosurfaces) of increasing isovalue. These combined visualisations allow a better determination of the relationship among the different level sets and extrapolation of the possible shape of the level sets not being displayed. Fewer views into the data are needed to achieve the same confidence and accuracy in the understanding of the structure of the field.

8.4 CONCLUSIONS

Traditional techniques for scientific visualisation lack the ability to explicitly present the structure of a scalar field to the user. We have presented a definition of topology diagram for PL fields and a straightforward algorithm for its computation and rendering. For 2D fields, the approach follows rigorous Morse theoretical definitions. In the 3D case, we have introduced some approximations to maintain the efficiency and simplicity of the computation.

The resulting topology visualisation serves both to provide information, which is not available in commonly used scalar visualisation techniques, as well as to reinforce or to enhance the information provided by standard visualisation techniques. In other words, the efficiency of a visualisation tool is improved not by reducing the image rendering time but by providing guidance to the user in the data exploration process.

9

Topology-Guided Downsampling and Volume Visualisation

Martin Kraus and Thomas Ertl

9.1 INTRODUCTION

Interactive rendering of volumetric data is one of the great challenges in computer graphics, with applications in scientific visualisation, virtual reality, computer games, and so on. In analogy to the success of texture mapping techniques over two-dimensional geometric representations of fine details, researchers have expected a similar success of volume graphics compared to polygonal graphics. However, this breakthrough in volume graphics has never happened – among other reasons, because the third dimension of volumetric data requires volumetric representations to grow faster than the size of the frame buffer, while two-dimensional texture maps may grow at exactly the same speed as the frame buffer without loss of image quality.

Nonetheless, volume rendering has found its niches and is slowly gaining relevance in interactive rendering techniques. While there are no alternatives for some applications, for example, in three-dimensional medical imaging, the volumetric representation of extremely fine structures, for example, for furs or garments, offers additional realism that is hard to achieve with pure geometric representations.

There are numerous approaches to direct volume rendering; however, only a few achieve interactive frame rates for any but rather small meshes. With respect to hardware-accelerated volume rendering, implementations of texture-based volume rendering (see, for example, Engel et al., 2001) offer impressive performances but require the volume data to fit into the texture memory of graphics adapters. Software-based approaches avoid this restriction and benefit from several optimisations that are hard to

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

implement in hardware-based approaches, in particular, before the development of programmable graphics hardware. Probably, the most efficient software-based algorithm for volume rendering is the shear-warp algorithm (see for example, Schulze et al., 2003). Remarkably, texture-based volume rendering and the shear-warp algorithm are both based on uniform volumetric grids.

The reason for this predominance of uniform grids is at least twofold: Firstly, their implementation is well supported by means of texture mapping in graphics hardware and they can be processed efficiently in a fixed, linear order in software. Secondly, more general – in particular, unstructured – meshes propose many additional difficulties because of their geometry, for example, a non-convex boundary and a non-trivial visibility ordering of cells, to name just two characteristic features of unstructured meshes. Therefore, the predominance of uniform grids in interactive volume rendering is likely to continue in the future.

As mentioned, the fast growth of memory requirements for volume graphics is the most important reason for their limited popularity. Therefore, simplification methods for volume data are even more important than the corresponding decimation techniques for polygonal meshes. Downsampling is of particular interest in this context as it converts uniform grids into coarser but still uniform grids, which is an important advantage considering the exceptional role of uniform grids for interactive volume rendering.

Traditional downsampling methods include sub-sampling, that is, successively deleting vertices, and replacing groups of vertices (for uniform volumetric grids usually $2 \times 2 \times 2$) by one vertex with the average data value as suggested for two-dimensional mip maps by Williams in (Williams, 1983) and for three-dimensional mip maps by Levoy and Whitaker in (Levoy and Whitaker, 1990). One generalisation of this method is to filter a mesh before sampling it at a lower resolution; for a recent application see (He et al., 1996). Many algorithms for volume visualisation have been accelerated by employing downsampled meshes, for example, ray casting (Danskin and Hanrahan, 1992, Levoy and Whitaker, 1990), splatting (Laur and Hanrahan, 1991), and isosurface extraction (He et al., 1996, Shekhar et al., 1996, Ohlberger and Rumpf, 1997, Westermann et al., 1999). For all these techniques, downsampling is an essential pre-processing step.

However, traditional downsampling methods ignore and, therefore, destroy the topology of the original scalar field. As the topology of a scalar field is based on its critical points, topology preservation of a scalar field is often defined as the preservation of all critical points (see, for example, Bajaj and Schikore, 1998, Gerstner and Pajarola, 2000). The theoretical framework for this definition is provided by Morse theory (see Milnor, 1963).

While the topology of a scalar field is not uniquely defined, the topology of surfaces – and isosurfaces in particular – is well defined. In fact, the topology of isosurfaces is strongly related to the critical points of the corresponding scalar field. Thus, the topology of isosurfaces extracted from downsampled meshes will usually deviate strongly from the topology of the original isosurfaces, that is, the number of disconnected components, tunnels, and holes will strongly differ.

Unfortunately, the topology of an isosurface is, in many cases, its most important feature as it allows the user to navigate in a volume, to identify noise in a data set, or to estimate the quality and plausibility of extracted shapes or structures. Therefore, it

is often useful to use topology-preserving simplification techniques in order to extract isosurfaces with the correct topology even from coarse meshes. Examples of such techniques have been published in (Bajaj and Schikore, 1998, Gerstner and Pajarola, 2000). However, these approaches are limited to simplicial meshes and are, therefore, not very well suited for volume rendering algorithms for uniform grids.

Topology-guided downsampling, a method first published by us in (Kraus and Ertl, 2001), fills this gap by providing a simple algorithm for downsampling uniform grids without blindly destroying the topology of the scalar field. This is achieved by calculating critical points and determining the data values of the downsampled mesh from this classification. The method is named *topology-guided downsampling* as topology-preserving downsampling is impossible, in general. However, even an approximate preservation of topology is highly desirable if isosurfaces are extracted from the downsampled grid, for example, for interactive previewing, because many topological features of the isosurfaces are preserved. After describing topology-guided downsampling in Section 9.2, the generation of simplified isosurfaces with this downsampling method is presented in Section 9.3. In Section 9.4, we discuss some recent developments in direct volume rendering and applications of topology-guided downsampling in this context.

9.2 TOPOLOGY-GUIDED DOWNSAMPLING

In order to show how to use topology-related concepts for structured meshes, this section presents a downsampling method for uniform volume meshes that preserves much more of the topology of a scalar field than existing downsampling methods, by preferably selecting scalar values of critical points. In particular, many critical points that are lost by traditional downsampling methods can be preserved.

As topology-guided downsampling works as well in two dimensions as in three dimensions, the algorithm will be illustrated with the help of the two-dimensional scalar field $f(x, y)$ depicted in Figure 9.1(a), which is defined by a bilinear interpolation between scalar values at the vertices of a two-dimensional uniform grid. Isolines are extracted from this mesh by decomposition into triangular cells and slicing the resultant height field with a horizontal plane as depicted in Figure 9.1(b).

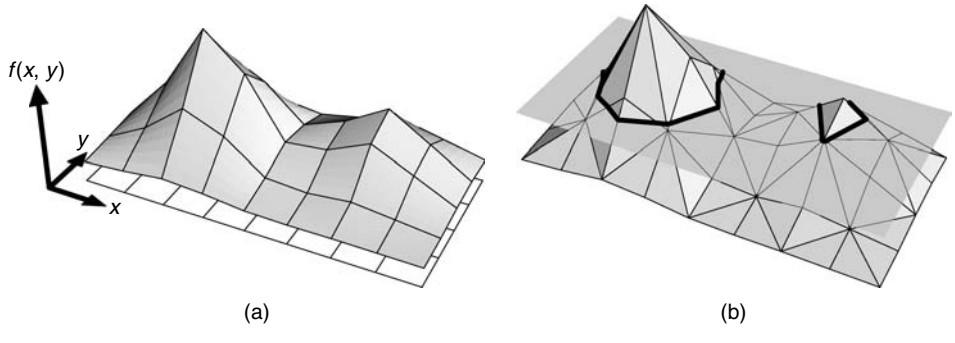

Figure 9.1 *(a) A two-dimensional scalar (height) field; and (b) piecewise linear approximation to an isoline in the scalar field of (a)*

In order to (approximately) preserve the topology of this scalar field, its critical points have to be preserved. The first step is therefore to identify critical points in two- and three-dimensional structured meshes.

9.2.1 Critical points in two dimensions

Critical points are local maxima, local minima, and saddle points. They indicate points where an isoline or isosurface changes its number of components or its genus. It is an important advantage of simplicial meshes, that is, triangular meshes in two dimensions and tetrahedral meshes in three dimensions, that all critical points are located at vertex positions. Therefore, two-dimensional structured meshes are usually decomposed into simplicial meshes as illustrated in Figures 9.1(b) and 9.2(a). It should be noted that this decomposition is only virtual, that is, it is not stored in any data structures but performed on the fly whenever it is required.

In order to handle vertices at the boundary of the mesh, the missing neighbours are (virtually) generated by mirroring neighbouring vertices across the boundary (see Gerstner and Pajarola, 2000). With this in mind, the decomposition into triangles employed in Figure 9.2(a) generates only two kinds of vertex neighbourhoods: one with four neighbours, as depicted in Figure 9.3(a), and another with eight neighbours, as depicted in Figure 9.3(b).

In analogy to (Gerstner and Pajarola, 2000), the corresponding surrounding polygon of a vertex is defined as the boundary of the adjacent triangles. The surrounding polygon defines an edge graph, which will be used in order to classify the surrounded vertex as a regular point, local maximum, local minimum, or saddle point.

This classification is achieved by marking each node of the edge graph, that is, each vertex neighbour of the surrounding polygon of a vertex. A neighbour is marked with 1 if its data value is greater than the value at the surrounded vertex, and a 0 otherwise. Then all edges between a 1 node and a 0 node are deleted, and the number of the remaining connected components of the edge graph is counted. The point is an extremum if this number is one. If it is two, then the point is regular; otherwise the point is a saddle point. The results of this classification for each vertex of the mesh of Figure 9.1(a) is visualised in Figure 9.2(a). Note that this classification ignores any degeneracies. This is legitimate as we are concerned only with an approximate preservation of critical points.

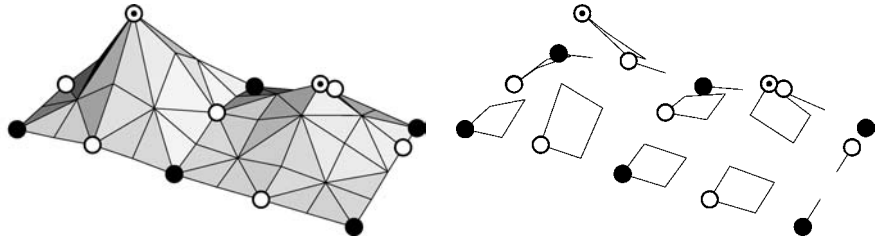

Figure 9.2 *(a) The critical points of the field of Figure 9.1(a). Maxima are marked with dotted circles, minima with disks, and saddle points with empty circles; and (b) the partitioning of the same mesh employed for downsampling*

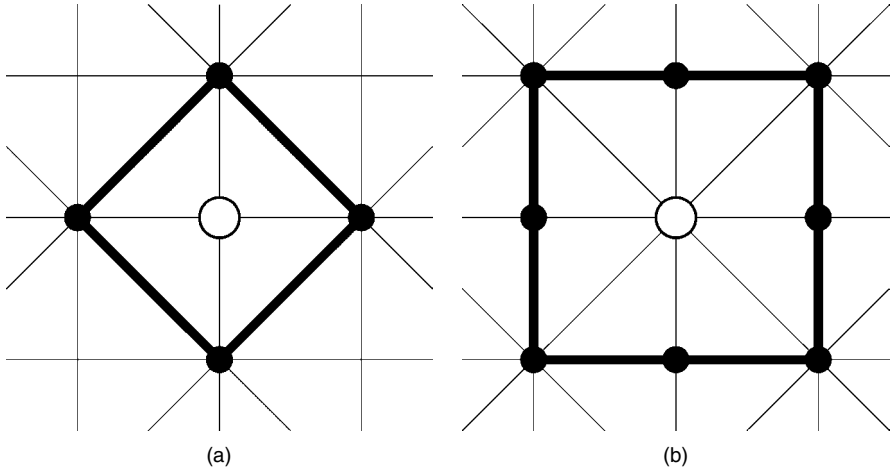

Figure 9.3 *The surrounding polygon (thick line) of a vertex with (a) four neighbours; and (b) eight neighbours*

9.2.2 Critical points in three dimensions

The first problem of a generalisation to three dimensions is to find a suitable tetrahedralisation of a structured hexahedral mesh. In (Carr et al., 2003) various decomposition schemes for three-dimensional structured meshes are discussed and a subdivision of each hexahedral cell into six square pyramids with their apices in the cell center is chosen, although this requires that new data points are interpolated. Topology-guided downsampling avoids new data points and therefore each hexahedron is subdivided into five tetrahedra. As mentioned in (Carr et al., 2003), this decomposition is not symmetrical as it generates two kinds of vertex neighbourhoods. Also note that the decomposition is only virtual, that is, it is performed on the fly.

In analogy to the two-dimensional case, the corresponding surrounding polyhedron of a vertex is defined by the boundary of the adjacent tetrahedra (see Gerstner and Pajarola, 2000). In the case of the decomposition of hexahedral cells into five tetrahedra, there are two different kinds of surrounding polyhedra: an octahedron and a triangulated cubeoctahedron (see Figures 9.4(a) and (b)). However, the approximative nature of the presented algorithm allows us to relax the need for a correct simplicial decomposition and employ the triangulated cubeoctahedron for all vertices.

The classification of vertices as regular points, local maxima, local minima, and saddle points is performed in a similar way to the two-dimensional case. In particular, nodes of the edge graphs defined by the surrounding polyhedra are marked in the same way: Nodes with a data value greater than the data value at the surrounded vertex with a 1, otherwise with a 0.

9.2.3 Preservation of critical points

One of the results of Morse theory is that all critical points of a scalar field have to be preserved in order to preserve the topology of all its isosurfaces. However, it is not

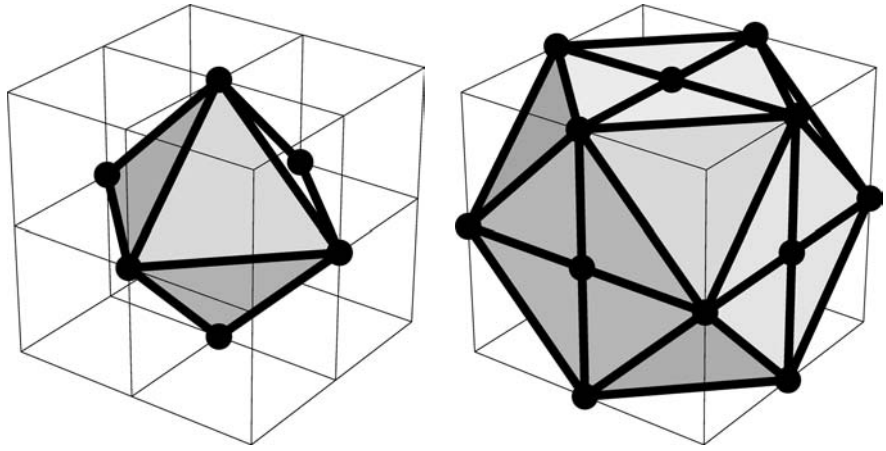

Figure 9.4 *Surrounding polyhedra of a vertex: (a) six neighbours define an octahedron; and (b) 18 neighbours define a triangulated cubeoctahedron*

necessary to preserve the exact geometric position of the critical points. Nonetheless, the scalar values at all critical points have to be preserved exactly. Otherwise the topology of isosurfaces for isovalues in the interval between the old and the new scalar value at a critical point is changed. For example, if a local maximum is preserved but its scalar value $v_{\max}$ is decreased to $v'_{\max} < v_{\max}$, all isosurfaces for isovalues in the interval $[v'_{\max}, v_{\max}]$ will be modified topologically. This is what usually happens to local extrema with the traditional combination of filtering and downsampling.

An example is given in Figures 9.1, 9.2(b), and 9.5. The scalar field of Figure 9.1(a) is downsampled by averaging the scalar values (i.e. heights) over groups of four (or less) vertices as indicated in Figure 9.2(b) (for now the marks of the critical vertices should be ignored). Each group of vertices corresponds to one new vertex of the downsampled mesh depicted in Figure 9.5(a). Because of the averaging, the height of both maxima is decreased in the new field. Therefore, the isolines for the same

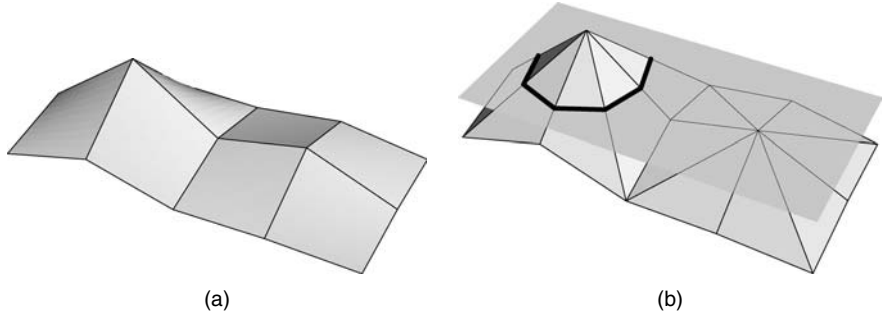

Figure 9.5 *(a) The scalar field obtained by averaging downsampling of the mesh of Figure 9.1(a); and (b) piecewise linear approximation to an isoline in the field of (a) for the same isovalue as in Figure 9.1(b)*

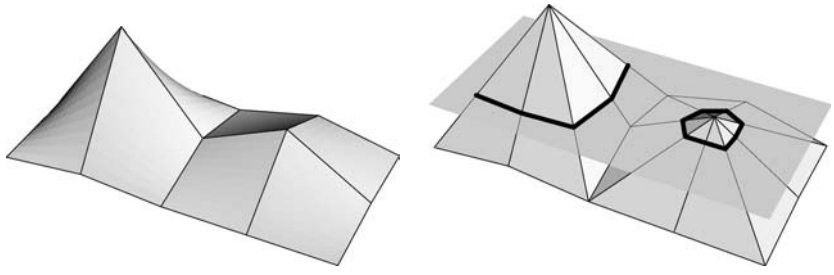

Figure 9.6 *(a) Same as Figure 9.5(a) but for topology-guided downsampling; and (b) same as Figure 9.5(b) for the field in (a)*

isovalue are topologically different for the original field and its downsampled version as illustrated by Figures 9.1(b) and 9.5(b).

The goal of the presented method is to avoid these changes whenever possible; therefore, linear filtering has to be avoided. Thus, an appropriate downsampling principle is to select and thereby preserve the scalar values of critical points. Although this selection does not guarantee the preservation of critical points, the preservation of the selected scalar values is a necessary condition for the preservation of critical points.

The selection is illustrated in Figure 9.2(b), where all critical points are marked. In this example, each group of vertices contains exactly one critical point. The scalar value of each critical point is then used for downsampling instead of the average height of the group of vertices. The resultant downsampled mesh is depicted in Figure 9.6(a). Figure 9.6(b) demonstrates that the topology of the isoline of Figure 9.1(b) is preserved with this downsampling technique. The following section describes topology-guided downsampling for three-dimensional meshes in more detail.

9.2.4 Steps of the algorithm

Topology-guided downsampling reduces the number of vertices of a volumetric structured mesh with even dimensions by a factor of eight by replacing groups of $2 \times 2 \times 2 = 8$ vertices by one vertex. For each disjoint group of 8 vertices, the following steps are performed in order to determine the scalar value of the new vertex. (If not given implicitly, the position of the new vertex is determined by the average position of the 8 vertices.)

1. For each vertex of the group, compute whether it is a regular point, a saddle point, or an extremum. Also, compute the average scalar value of these vertices.
2. If there is no critical point, the average scalar value is the result.
3. If there is only one critical point, its scalar value is the result.
4. If there are multiple saddle points but no extremum, the scalar value of the saddle point with the largest absolute distance to the average scalar value is the result.
5. If there are (multiple) saddle points but only one extremum, the scalar value of the extremum is the result.
6. Otherwise, the scalar value of the extremum with the largest absolute distance to the average scalar value is the result.

Steps 1 to 3 are motivated by the considerations described above. Steps 4 to 6 reflect an interest in the most "important" critical points, since many saddle points would not exist without a neighbouring extremum and distant critical points are likely to have more influence on the topology of isosurfaces than critical points close to the average scalar value.

This downsampling procedure can be applied repeatedly – each time reducing the number of vertices by a factor of 8. However, in comparison to averaging downsampling methods, much more of the topological information is preserved by this algorithm, as is demonstrated in the next section.

9.3 EXTRACTING ISOSURFACES FROM DOWNSAMPLED GRIDS

Topology preservation has seldom played an important role for the simplification of volumetric grids. This is partly due to the fact that a complete preservation of the topology of a scalar field is impossible with downsampling techniques. More importantly, the shape and topology of isosurfaces was considered less important for volume rendering as long as isosurfaces could not be rendered with the help of volume rendering techniques but had to be approximated by polygonal meshes. Since these polygonal meshes can be simplified without affecting their topology, the simplification of the volume data was of less interest. Moreover, isosurface extraction was often accelerated by hierarchical space partitioning, which was successfully combined with topology preservation (see Gerstner and Pajarola, 2000).

However, the simplification of volumetric meshes offers considerable advantages as it reduces the amount of data and, therefore, accelerates any visualisation method, including the extraction of isosurfaces. Furthermore, the simplification of a volumetric mesh is independent of any visualisation parameter, including isovalues; thus, the simplification does not have to be repeated when the user modifies these parameters. In contrast to this early simplification, any decimation of a polygonal approximation to an isosurface has to be re-computed for each new isovalue. Thus, topology-guided downsampling combines the advantages of an early mesh simplification with an approximate preservation of the topology of isosurfaces.

Our first example is a CTA (computer tomography angiography) volume data set showing blood vessels around an aneurysm. It is well suited for the demonstration of topology-guided downsampling as it contains noise and structures of very different sizes. The resolution of this data set is $128 \times 128 \times 60$ voxels and 8 data bits per voxel. In order to visualise it, an isosurface for a fixed isovalue is extracted with a simple marching tetrahedral algorithm after decomposing the uniform mesh into tetrahedra as explained in the previous section. Figure 9.7(a) depicts the resultant isosurface of the original data set. All isosurfaces are rendered using flat shading with surface normals calculated directly from each triangle in order to emphasise the underlying grid structure even for very fine meshes. Of course, pre-integrated volume rendering for uniform grids (see Section 9.4) could also be used to render these images.

The downsampling results of the presented algorithm will be compared to a simple averaging downsampling that replaces eight vertices by one vertex with the average data value as employed in (Williams, 1983, Levoy and Whitaker, 1990, Danskin and Hanrahan, 1992). More general filtering and downsampling methods, for example, (He

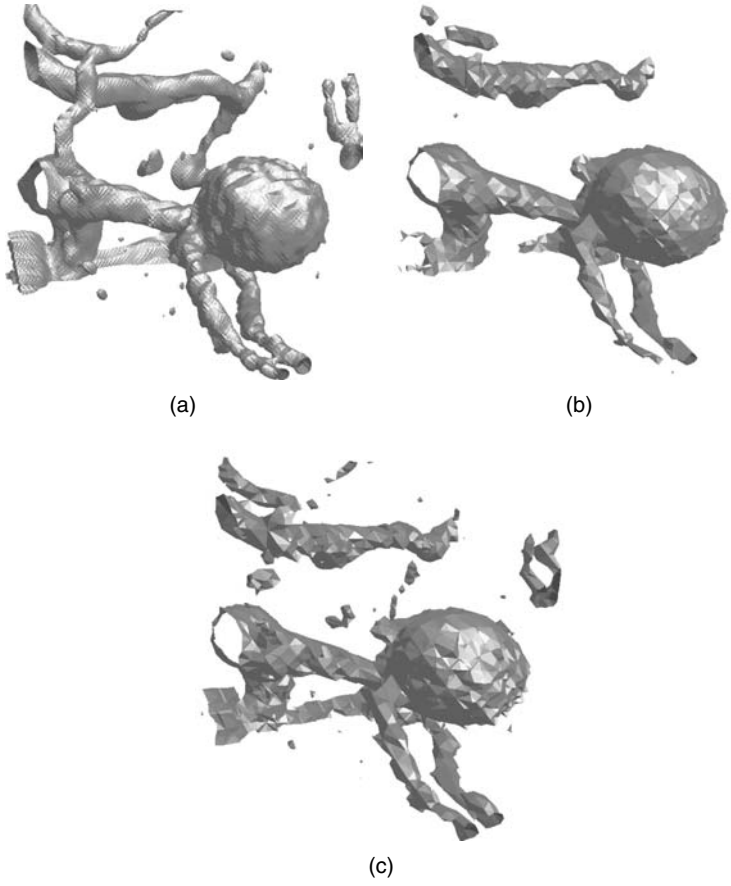

Figure 9.7 *(a) An isosurface extracted from a* $128 \times 128 \times 60$ *CTA volume data set; (b) same isosurface extracted from a mesh downsampled to dimensions* $32 \times 32 \times 15$ *with averaging downsampling; and (c) same as (b) with topology-guided downsampling*

et al., 1996), suffer essentially from the same problems for a comparable downsampling rate.

Figure 9.7(b) depicts the isosurface for the same isovalue as in Figure 9.7(a) but extracted from a downsampled volume of dimensions $32 \times 32 \times 15$ using traditional averaging downsampling. In contrast, Figure 9.7(c) depicts the result for the same settings but uses topology-guided downsampling as described in the previous section with the cubeoctahedron being the surrounding polyhedron of all vertices. The compression rate is $(1/8)^2 \approx 1.6\%$ in both cases. Note that none of the two downsampling methods depends on a particular isovalue, that is, the user may choose an isovalue after the downsampling, which is only a pre-processing step.

Obviously, the noise manifesting itself in small disconnected parts of the original isosurface in Figure 9.7(a) is partially preserved with topology-guided downsampling in Figure 9.7(c) but is almost completely lost with averaging downsampling in Figure 9.7(b). More importantly, several crucial connections of blood vessels visible in Figure 9.7(a)

become disconnected in Figures 9.7(b) and (c). However, topology-guided downsampling preserves at least parts of the vessels while averaging downsampling results in larger gaps, or even the complete vanishing of parts of vessels, for example, at the top of Figure 9.7(b).

Our second example is a CT scan of a bonsai, which features a sharp but very complex border between air and the plant with many fine details. Figure 9.8(a) depicts the whole isosurface. The grid's resolution of $256 \times 256 \times 128$ vertices is high enough to reconstruct single leaves. This way of representing a tree is related to shape modelling techniques based on voxelised scenes (see He et al., 1996). Again, we will show that topology-guided downsampling preserves more details of the shape for higher downsampling rates, which is crucial for this kind of applications.

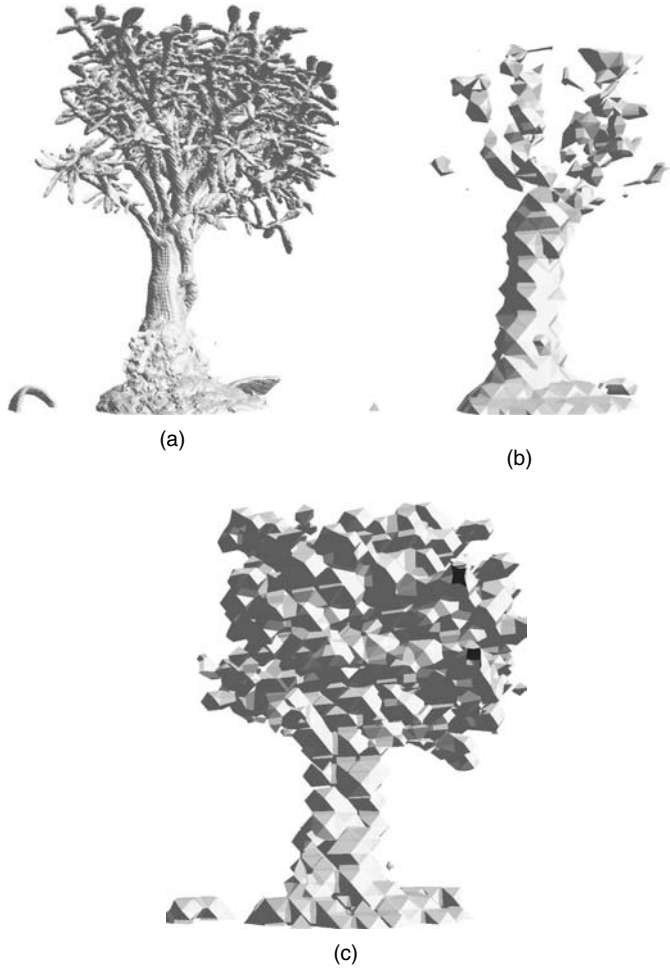

Figure 9.8 *(a) An isosurface extracted from a CT scan of a bonsai; (b) same isosurface but extracted from a grid of dimensions* $32 \times 32 \times 16$ *with averaging downsampling; and (c) same as (b) with topology-guided downsampling*

Figures 9.8(b) and (c) show the same isosurface after three downsampling steps. While the shape is no longer recognisable after averaging downsampling in Figure 9.8(b), topology-guided downsampling preserves a coarse representation of the original shape, as shown in Figure 9.8(c). (The isosurface in Figure 9.8(c) was clipped at the borders of the volume; this resulted in two dark holes.)

This example suggests that topology-guided downsampling is not only useful for scientific volume visualisation but also for shape modelling based on volume graphics, in particular, if models have to be represented with different levels of detail.

9.4 DOWNSAMPLING FOR DIRECT VOLUME RENDERING

There has been considerable progress in recent years with respect to the rendering of isosurfaces with the help of hardware-accelerated volume rendering techniques. Westermann and Ertl (1998) were one of the first to propose a method based on texture-based volume rendering for isosurfaces. In (Westermann et al., 2000), Westermann et al. extended this technique to multiple isosurfaces. Rezk-Salama et al. (2000) employed programmable colour computations of modern graphics adapters in order to implement advanced shading computations for isosurfaces generated by texture-based volume rendering. Further progress was initiated by Engel et al. (2001), who applied the concept of pre-integration to texture-based volume rendering, and thus achieved a high image quality at interactive frame rates without the need for an extreme rasterisation performance. Moreover, pre-integrated volume rendering allows us to render any number of shaded isosurfaces of any shape. On the basis of this idea, Meissner et al. (2002) discussed shading of multiple isosurfaces. Furthermore, pre-integration was also combined with the shear-warp algorithm by Schulze et al. in (Schulze et al., 2003).

Unfortunately, the high frame rates offered by pre-integrated, texture-based volume rendering (as proposed in Engel et al., 2001) are only possible if the mesh data fit into the local texture memory of the graphics adapter, otherwise bandwidth limitations will reduce the frame rates dramatically. Therefore, it is essential to reduce the texture memory requirements accordingly, that is, to simplify the corresponding uniform grids such that the textures fit into the available texture memory. As mentioned, pre-integrated volume rendering allows for the rendering of isosurfaces; thus, the preservation of the scalar field's topology can become just as important for volume rendering as it is for the extraction of isosurfaces. Therefore, topology-guided downsampling is one of the few simplification techniques that is appropriate for this purpose.

Programmable graphics hardware has also been employed for the implementation of advanced data structures. One important disadvantage of uniform meshes is their lack of adaptivity, that is, the fixed resolution and the box-shaped boundary. In order to compensate for this non-adaptivity, uniform meshes usually need to have a higher resolution and a larger domain than unstructured meshes for the same problem. This, however, results in large mesh sizes, which are inappropriate for texture-based volume rendering algorithms because of the usually quite limited texture memory of graphics hardware. Thus, the main advantage of uniform meshes, that is, the support by texturing hardware, is often severely diminished.

In order to overcome this dilemma, we proposed the concept of adaptive volume textures in (Kraus and Ertl, 2002). Independently, McCool (2000) proposed even more

advanced concepts in (URL #2); in particular, the "sparse blocked texture storage" is a very similar technique. Adaptive texture maps offer at least a limited form of adaptivity in the twofold sense of a locally adaptive resolution and an arbitrary boundary of the mesh. They are based on a two-level representation of mesh data that is appropriate for hardware-accelerated on-the-fly decoding with programmable texturing hardware. This kind of texture compression is particularly well suited for the texture-based volume rendering algorithms mentioned above.

The generation of adaptive volume textures requires the generation of a hierarchy of downsampled grids. If the choice of the downsampling level is based on an approximation error between the original and the downsampled data, a downsampling method based on simple averaging or linear filtering is often appropriate. However, the choice of the downsampling level may also be based on topological features of the data, for example, because the topology of extracted isosurfaces is more important for a particular application than any overall approximation error. Moreover, even if the choice is based on an approximation error, a large approximation error might be acceptable and a preservation of topological features can be desirable at the same time. In these cases, topology-guided downsampling is the appropriate downsampling algorithm. Similarly, it is useful for the generation of many other hierarchical data representations.

9.5 SUMMARY

In this chapter, we have reviewed the basic algorithm of topology-guided downsampling; in particular, the role of critical points and, more specifically, the importance of preserving the data values at critical points was discussed. Although the basic steps of the algorithm were explained in two dimensions, our focus in this chapter was on applications of topology-guided downsampling to three-dimensional scalar fields. Therefore, we illustrated the algorithm with volumetric data from medical imaging. Of course, topology-guided downsampling may also be applied to many other problems in volume graphics, for example, visualisation of flow data or geophysical data, or rendering of natural phenomena such as smoke, clouds, fire, or semi-transparent fluids.

We have discussed techniques of indirect volume visualisation, that is, the extraction and rendering of isosurfaces, and direct volume rendering, namely, pre-integrated, and texture-based volume rendering. As mentioned, the possibilities offered by programmable graphics hardware with respect to the rendering of isosurfaces and the handling of advanced data structures have led to a strong need for topology-preserving algorithms for the simplification of uniform grids because hardware-accelerated texture mapping usually relies on uniform grids. Since a complete preservation is not possible for these meshes in general, the approximate topology preservation of topology-guided downsampling appears to be the most appropriate downsampling method.

As volume rendering becomes more popular and, at the same time, more demanding because of the ever-growing sizes of data sets, innovative solutions are strongly required for many of the basic problems in volume graphics. In this chapter, we have shown that topology-related concepts can lead to successful solutions for problems in volume rendering not only of unstructured but also of uniform volumetric meshes. Moreover, we are convinced that there are many more applications of topology-related approaches in volume graphics that are still to be discovered.

10

Application of Surface Networks for Augmenting the Visualisation of Dynamic Geographic Surfaces

Sanjay Rana and Jason Dykes

10.1 INTRODUCTION

The animation of geographic surfaces as a temporal series (e.g. an evolving part of sea coast) and attribute series (e.g. a sequence of population density surfaces) is done widely in geovisualization. These animations are popular because they reveal the spatial variations in a single frame, thus eliminating the effort to memorise and match differences. The techniques for animating surfaces have evolved from simple paper cartoons (McCloud, 1993) to sophisticated hardware–software driven solutions of modern times (Ware, 2000). Geovisualization researchers in collaboration with computer graphics researchers have tried to change the static nature of geospatial datasets' visualisation with ever-advancing and aesthetically appealing geovisualization interfaces. As a consequence, surface animation function is often a standard component of many current geographic information systems (GIS). However, despite the vast improvements in technology, a question posed in early 1960s by Bertin (Bertin, 1967), "whether animation helps in a better understanding", is still thrown back and forth between geovisualization researchers. There have been a number of attempts to characterise the issues in animated geovisualization (Emmer, 2001, Ogao and Blok, 2001, Ogao and Kraak, 2001, Koussoulakou, 1990, Dibiase et al., 1992, Peterson, 1993, MacEachren, 1995a,b, Ware, 2000, Slocum et al., 1990). Please refer to Ware (2000)

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

for a comprehensive comparison between advantages and disadvantages of animated visualisation and identifications of suitable research directions.

Bertin's main argument against animated maps is that the presence of motion distracts a user's attention from the visual properties (e.g. colours, shape etc.) of symbols, thereby resulting in a limited interpretation. Unlike static maps, an animated map requires continuous attention to the stream of information. Bertin's criticism is further strengthened by Miller's (1956) observation that humans could only follow about 7 ± 2 visual cues simultaneously. In other words, it cannot be guaranteed that animation will be useful for interpretation due to the free flow of information. Although Dibiase et al. (1992) and MacEachren (1995a,b) proposed methods to control the transient symbology in animation, formal and generic guidelines for the use of these visual variables do not exist. Therefore, whilst their effectiveness is largely unknown, Bertins' (Bertin, 1967) objections are not fully satisfied. Please see (Gershon, 1992, and Acevedo and Masuoka, 1997), for studies on the implications of dynamic visual variables, such as frequency, frame rate, and others, on time-series animations.

We believe that in most cases this limitation of animated geovisualization has arisen from mainly two sources-namely, the conceptual (e.g. design-related issues) and implementation (e.g. software, hardware) limitations. In the not so distant past, limited hardware capabilities and non-graphics oriented languages restricted the scope of animation. Certainly, the hardware and software available to generate animations has improved significantly (Earnshaw and Watson, 1993, Gahegan, 1999) but a desktop solution for our often massive surface datasets still seems some years away. On the contrary, conceptual limitations are less well defined but at least they do not require an in-depth understanding of modern sophisticated computer hardware and software. The above limiting factors start to take effect from the start of the geovisualization process, that is, preparation of spatial datasets (e.g. lack of spatio-temporal continuity in spatial datasets) and then eventually lead to interpretation stage as information overload. In our view, visualisations available as part of the AIDS Data animation project (URL #3) is one such example of poor design and implementation, in which because of the high and sudden variations in successive frames, the inter-frame variations in spatial patterns appear as movements to the viewer. MacEachren (1995a,b) offers a perceptual and cognitive treatment for such misleading interpretations. The combined effects of these factors are distraction, poor retention, and lack of clear expression of the information (Morrison et al., 2000, Gahegan, 1999, Openshaw et al., 1994).

In this chapter, we offer an approach that will address both the design and implementation-related limitations in geovisualization. Our approach is based on an extension of the proposals by Pascucci (see Chapter 8) on using the surface network for scientific visualisation to represent dynamic geographic surfaces. We will demonstrate how the surface network representation offers both intuitive design insights and also improvements in implementation. A much detailed implementation of our approach can be found in an earlier version of this work in (Rana and Dykes, 2003). In this chapter, we present the parts of our approach related to the role of surface networks in geovisualization and steps leading to it.

Our approach takes advantage of techniques from computer graphics and geography. It must be stressed at this stage that the exact implementation of any of our approaches should inevitably vary according to the context of the visualisation (data,

user, use, etc.). We want to emphasise that the approach proposed here is not in any way the "optimal" or the "most effective" method to use. At the same time, our aim is to provide sufficient explanation in the following sections to demonstrate the proposed approach and use examples to illustrate ways in which it may be applied in a flexible manner.

10.2 PROPOSAL

Like in previous chapters, in order to realise a surface network, we assume that the geographic surface is a doubly continuous function of the form $z = f(x, y)$, where z is the property (e.g. elevation, population density etc.) being mapped and associated with a point (x, y). Although this topological integrity of surface is required to ensure mathematic tractability, it is not crucial for visualisation. Therefore, the visualisation of any surface that contains surface network features, namely, the peaks, pits, passes, ridges, and channels, could equally benefit from our approach even though it might not be based on a consistent surface network. This is because we believe that surface network in any form highlights the *information* of the surface (data) where the definition of *information* is based on Shanon's Information Theory (Salomon, 1998) that *information* is only the useful part of the data. We will supplement this argument with more reasons in Section 10.2.2.

10.2.1 Step 1: Ensuring high or increased spatio-temporal continuity

Commonly available, digital surface datasets (e.g. rasters) that model continuous geographic phenomena often have coarse spatial and temporal solutions. The most common reason for poor spatio-temporal resolutions in socio-economic geographic surfaces is deliberate aggregation of high-resolution point dataset into large cells in order to protect the privacy of the population. For example, the UK population density surfaces available from Census Dissemination Unit, UK (URL #4) have a spatial resolution of 200 m (Figure 10.1) although the original dataset is collected at household levels. In some cases, poor spatio-temporal resolution could also be a result of commercial/strategic interests and/or the lack of resources to collect data at higher resolutions.

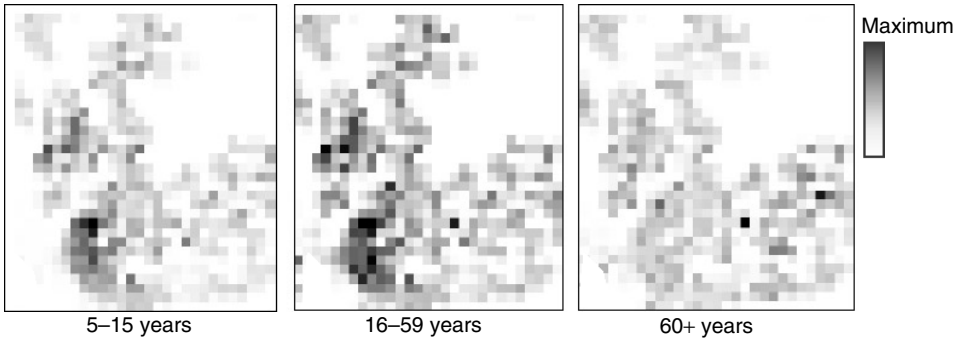

Figure 10.1 *Population density surfaces of the different age cohorts in an area in NE Leicester, UK*

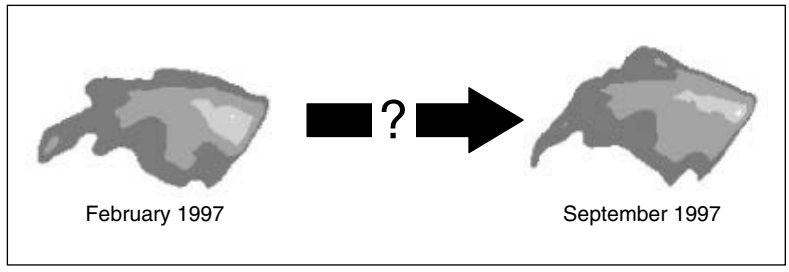

Figure 10.2 *Digital elevation models of a sand spit at Scolt Head Island, North Norfolk, UK. Two situations are shown representing the results of survey of the feature in 1997*

For example, Figure 10.2 shows the digital elevation model of sand spit under active denudation at Norfolk coast, England, generated from height data collected only twice a year (Source: Jonathan Raper, City University, UK). Clearly, the end user, that is, the visualiser/animator of these spatial datasets, is unaware of the information related to lost details. The lost details are, however, indispensable since they only can create the impression of a smooth continuity, fundamental to animation. In the absence of the vital information about the surface, the cartographic principle of introducing arbitrary details can be applied to generate aesthetically pleasing surfaces. In other words, a "cartographic lie" will have to be incorporated in the visualisation process. We propose the following two methods for increasing the spatio-temporal continuity.

10.2.1.1 Increase spatial and attribute resolution

There are many types of spatial interpolation possible for generating a smooth surface from the coarse-resolution surfaces. Openshaw et al. (1994) suggested the use of density estimation methods, such as those proposed by Gatrell (1994) and Bracken (1994), for creating a smooth map display of socio-economic data. More recently, Paddenburg and Wachowicz (2001) studied the use of spatial generalisation to reduce noise in raster surfaces and concluded that this pre-processing reveals the *true information* otherwise suppressed by noise. Simpler methods such as fitting a bivariate quadratic polynomial function (Wood, 1998) through the surface to derive the smooth interpolated forms of the surface are equally effective. Spatial resolution could be done in the following two ways:

(i) Increasing attribute (thematic) resolution by interpolation to the current spatial resolution. The interpolation of the attribute values of the surface (e.g. elevation) using a quadratic polynomial function also results in an increase in the attribute resolution as the abrupt differences between adjoining attribute values are reduced. This can be considered "attribute smoothing".
(ii) Increasing the spatial resolution by interpolation to a higher spatial resolution. This process involves the generation of additional data and the visual effect is one of spatial smoothing. In cartographic terms, this process could be considered as an "exaggeration".

10.2.1.2 Increase inter-frame continuity

As shown in Figure 10.2, because of practical limitations, ordered sequences of terrain surfaces could not be sampled frequently enough to create a continuous temporal series; yet, the feature changes constantly in the dynamic coastal environment in which it is subject to denudation and deposition (Raper, 2000). The temporal gaps lead to abrupt changes in the animation of dynamic surfaces (Shepherd, 1995). In the case of the socio-economic surface animation (Figure 10.1), the variations between the different age cohorts will also be unnoticeable because of the sudden and subtle changes. Attempts to reduce the abrupt jumps between successive situations depicted in an animation so as to increase inter-frame continuity include adjustment of the "duration" dynamic visual variable, either by slowing the sequence or through direct user control. Alternatively, additional situations can be derived from the data to smooth transitions. This step is also an "exaggeration" effect with the aim to include "microsteps between larger steps", as these are found to be beneficial to the viewer (Morrison et al., 2000).

A number of techniques exist for generation of animations in this way. One of the simplest methods is *blending*, through which a smooth transition of intermediate situations or "microsteps" can be achieved. Blending is used widely in the computer graphics field for transforming one particular shape or object into another (Gomes et al., 1998). It is also available in commercial graphics software such as 3D Studio Max (URL #5), which provide tools for applying the technique to both raster and vector spatial datasets. A basic implementation of blending involves a linear interpolation between the two consecutive situations (frames); however, a more sophisticated non-linear interpolant could also be used to visualise punctuated phenomena. The MapTime software (Slocum et al., 2001) makes use of the first of these options for generating intermediate frames between two situations[1].

10.2.2 Highlight the information in surface

As mentioned in the introduction, human visual processing has limited capabilities for interpreting the parallel information streams that characterise dynamic processes (Ware, 2000). Human cognition processes, especially the working memory can follow at most 7 ± 2 simultaneous cues (Miller, 1956, Ware, 2000). Therefore, highlighting the *information* is a key cartographic objective when scenes are complex. Morrison et al. (2000) indicated that a clear apprehension and expression of the conceptual message is essential in animated graphics. Tobler (1970) proposed reducing complex processes into component parts or simplified representations. Dransch (2000) identifies a number of factors that may enhance the cognition process in multimedia systems, including the need to "increase the important information". This can be achieved through careful and meaningful simplification.

The two key types of information in dynamic geographic surfaces are the structure of the surface and the local importance of points (locations). However, in the common representations of surfaces such as the colourmap (equivalent cartographic representation is a hypsometric tint) and contour map (or coloured isopleth), the transfer of the

[1] Situations are termed *key frames* in the context of the MapTime software.

structural information is dependent upon the contour interval, and spatial and thematic resolutions (Bajaj and Schikore, 1996; See also Chapter 8). Therefore, a representation of surface is required, which would provide an objective and yet natural representation of the surface morphology and structure.

Fowler and Little (1979) proposed that the fundamental topographic features of a surface, namely, the peaks, passes, pits, ridges, and channels, are sufficient to describe the significant information about a surface. These topographic features constitute the surface network, therefore, an application of surface network in computer graphics has been the visualisation of the structure of surfaces. For example, Helman and Hesselink (1991), and Bajaj and Schikore (1996) demonstrated that the surface network representation could enhance the graphic representation of vector and scalar surfaces significantly as compared to the use of colourmaps and contour maps. The surface is also broken down to five main information streams (3 point types and 2 line types), which would make the changes easily observable. Helman and Hesselink (1991) reported that the surface network representation helped in both visualisation and reduction in storage space. Therefore, it can be argued that the derivation of surface networks from dynamic surfaces has the potential to highlight the information when animating sequences of surfaces for visualisation, thus reducing the load on the viewer and potentially aiding interpretation.

In terms of Dransch's proposals (2000), extracting surface networks will correspond to a step aimed at increasing the important information, reducing the information overload, and helping in the creation of a mental model of the dynamic processes. For example, Figure 10.3 shows a comparison between the surface network representation, contours, and colourmaps of the Norfolk coast sand spit based on their ability to describe the structural information of the surface.

In summary, a surface network representation is useful for the visualisation of dynamic surfaces animation because of the following reasons:

(i) The consistent definition of surface network means that it can be used to quantify and isolate changes. The surface network provides a frame of reference that could be used to track changes in the surface, for example, the rate of the displacement of the ridge lines through an animation could indicate the behaviour of the surface under changing conditions.

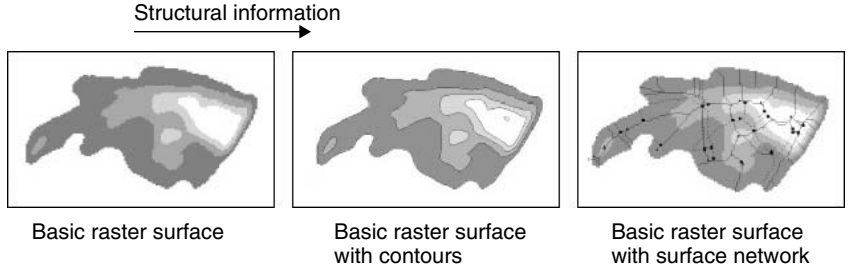

Figure 10.3 *Increase in the structural information delivery with the addition contours and surface network overlays*

(ii) The use of point and line symbols to represent surfaces enables the viewer to take advantage of their natural propensity to interpret attribute change between successive scenes as motion and reduces the possibility of minor variations in visual variables being interpreted as such. The surface network is thus conceptually similar to the ideas of topological rendering of volume data sets proposed by Upson and Kerlick (1989), and Kerlick (1990).

10.2.2.1 Generation of the surface network

As shown in Chapters 3 and 4, there are various methods for the generation of surface networks, for example, depending upon the digital elevation model, that is, whether raster, triangulated irregular network or contours. It is beyond the scope of this chapter to discuss the methods in detail as they have already been dealt in detail in Chapters 3 and 4. However, we will highlight an important issue regarding the generation of feature networks, especially related to raster geographic surface. Automated raster processing suffers from the limitation that the results of the analysis are subject to the size of the local neighbourhood, that is, the window or kernel, centred at the study point, used to perform the processing. In terms of feature extraction, this results in scale dependencies as topographic features exist across a range of scales and will be detected by kernels of different sizes (Wood, 1996a,b). For example, the triangulation-based feature extraction method has the limitation that it only triangulates over the local neighbourhood of a point; therefore, larger scale features may remain undetected. Similarly, the bivariate quadratic surface fitting also has the limitation that kernel size is fixed for each iteration of feature extraction but unlike the previous approach the kernel size can be increased/decreased iteratively until a visually acceptable level of simplification has been achieved. While it is clear that the extraction of the features at all the scales cannot be guaranteed, surface networks offer a form of representation of the surface that may be suitable for visualisation that takes advantage of animation. It is likely to lead to insight into the nature of both the simplification and the simplified surface. Later in Section 10.3.3, an example of the idea of scale series using animation will be discussed to investigate the effects of scale dependency and interesting insights provided by them.

10.3 IMPLEMENTATION

Our study data were the population density surfaces and digital elevation models shown in Figures 10.1 and 10.2. The implementation revealed some promise and highlighted a number of issues. Various controls to support animated, sequential, and conditional interaction (Krygier et al., 1997) were implemented in an application surface network visualiser (SNV) for animating surface networks such as those derived here, in order to support visualisation. SNV allows an animation of an ordered sequence of surface networks and a range of levels of sophistication of visualisation tasks (such as query, interactions) as identified by Crampton (2002). SNV also allows a graphic lag whereby a user-defined rate of change in the lightness of symbols representing the surface network features is used to fade in and fade out between successive situations. In its approach, graphic lag essentially implements a type of epichronic symbolism

(Shepherd, 1995), similar to the ideas of Levy et al. (1970) and Openshaw et al. (1994), for controlling the brightness and luminosity of symbol colours, respectively.

We used a curvature-based feature extraction to derive the surface network (see Chapter 4 for more information). This feature extraction is available in the software LandSerf written by Jo Wood (URL #6).

10.3.1 Animation of temporal series

Figure 10.4 shows the inter-frame continuity achieved by blending (using linear interpolation) the terrain of the sand spit recorded in February 1997 into that recorded in September 1997. While we can clearly observe the variations in relief of the surface, it is not possible to assess the changes in the structure, as the structural changes are not obvious from the field view. Figure 10.5 shows a part of the same sequence of the blending with an overlay of the surface network, in which the changes in the structure can be identified. Note the detection of the changes in topographic features that are significant at this scale of measurement at the top right of the spit in Figure 10.5. The animation can be accessed online and assessed (URL #7).

10.3.2 Animation of attribute series

The interpolation of the population density surfaces to 200-m (same as original spatial resolution) spatial resolution (Figure 10.6) improves the spatial continuity from that

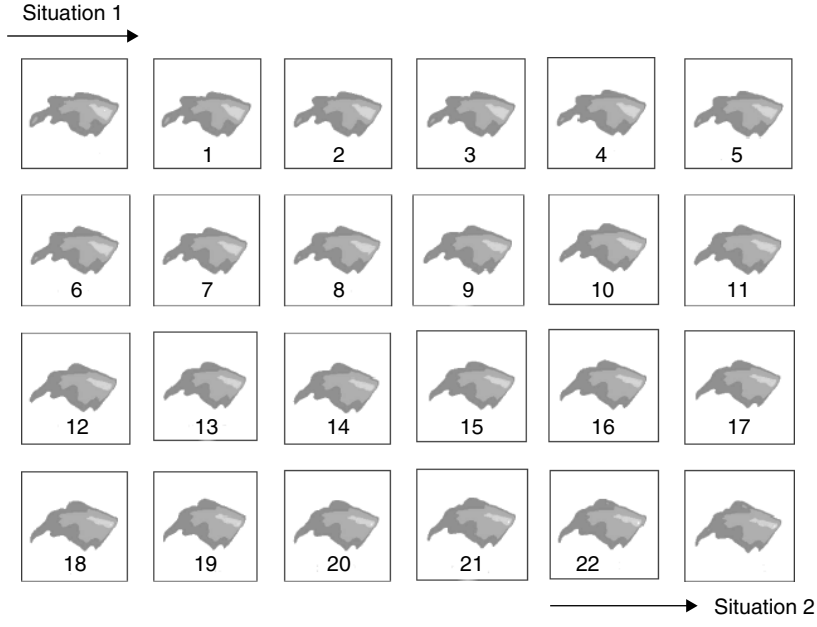

Figure 10.4 *22 Intermediate surfaces (microsteps) generated by blending the February, 1997 surface (Situation 1) into the September, 1997 surface (Situation 2)*

Figure 10.5 *Use of the surface network representation to visualise the changes in the morphology of the sand spit. The box indicates an area of interest. Note that the surface network variations highlight changes that are not evident from the representation that uses colour to show variation in elevation*

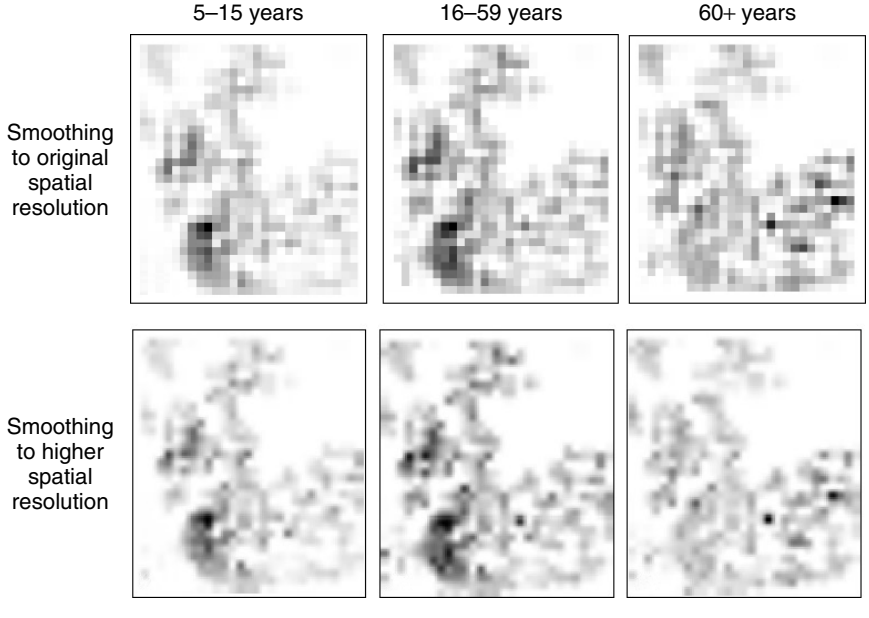

Figure 10.6 *Smoothing population density surfaces by interpolating to the original (i.e. 200 m) and higher (i.e. 50 m) resolution. (See Figure 10.1 for the original data)*

shown in Figure 10.1 but the cell edges were still visible. Further smoothing by interpolating to 50-m cell resolution resulted in a less noisy distribution that is more suitable for animation and topographic feature extraction (Figure 10.6). The surfaces interpolated to a spatial resolution of 50 m were thus used in this instance to derive the surface networks. The surface networks of the population density surfaces (Figure 10.7) reveal the following characteristic spatial patterns of the different age groups:

- The surface networks of the 5 to 15 years and 60+ years age group are generally sparser than the 15 to 59 years age group.
- Some of the local population density peaks suppressed in the colourmaps (Figure 10.6) are revealed by the surface network representation. For example, see the cluster of points in the NW quadrant with some of the highest deviation from the average

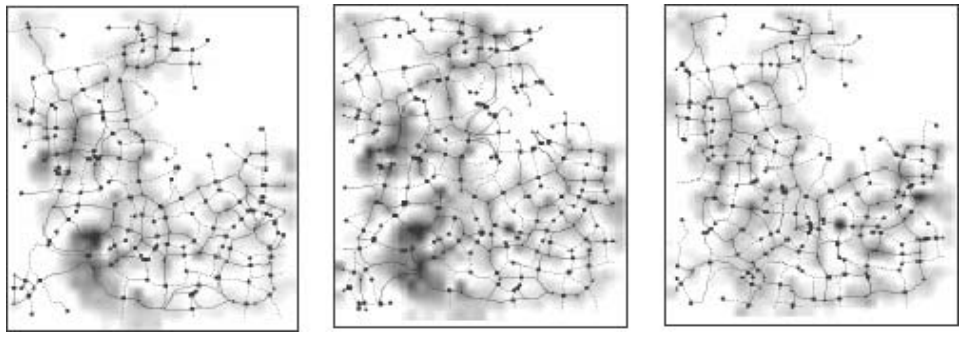

Figure 10.7 *Use of surface network representation to compare the population density distribution of three age cohorts in NE Leicester. The surface network was extracted from the 50-m grid resolution surfaces in Figure 10.6 using a filter window size 9. Note how the variations in the density of surface network highlights the inhomogeneous age distribution in some parts (e.g. south-east) of the study area*

population density. Interactive techniques for extracting/suppressing information in SNV allow the user to focus in on particular areas of interest such as this.

- The stability of population centres in areas of high population through successive age cohorts until the older generation is assessed, which displays significant variation.

10.3.3 Scale series

Because of the multi-scaled nature of the properties characterising most surfaces, there are multiple valid surface-network representation of the surfaces (Figure 10.8). The scale dependency can provide interesting insights into the structural organisation of the mapped property. For instance, Figure 10.8 reveals the gradual aggregation of the topographic features around the major urban centres with an increase in the feature extraction window size. However, it remains difficult to arrive at a particular extraction window size suitable for representing a surface. The choice is likely to be governed by the scale of the area of interest. A window size that could identify most topographic features in the surface should be the first choice. For instance, in the case of Figure 10.8, if one is interested in identifying minor variations in the surface, then window size 5 is useful and similarly window size 19 will be the most capable one for detecting larger variations. Animation can however be used to address these issues. An ordered sequence of feature networks generated across a range of scales can be a useful tool for visualisation. Using the animation techniques presented here may help us gain insight into the scale dependence of morphometric feature networks. Indeed, animation is likely to be a useful tool to sequence through any number of alternative graphical representations of the data set derived from the flexible application of the framework.

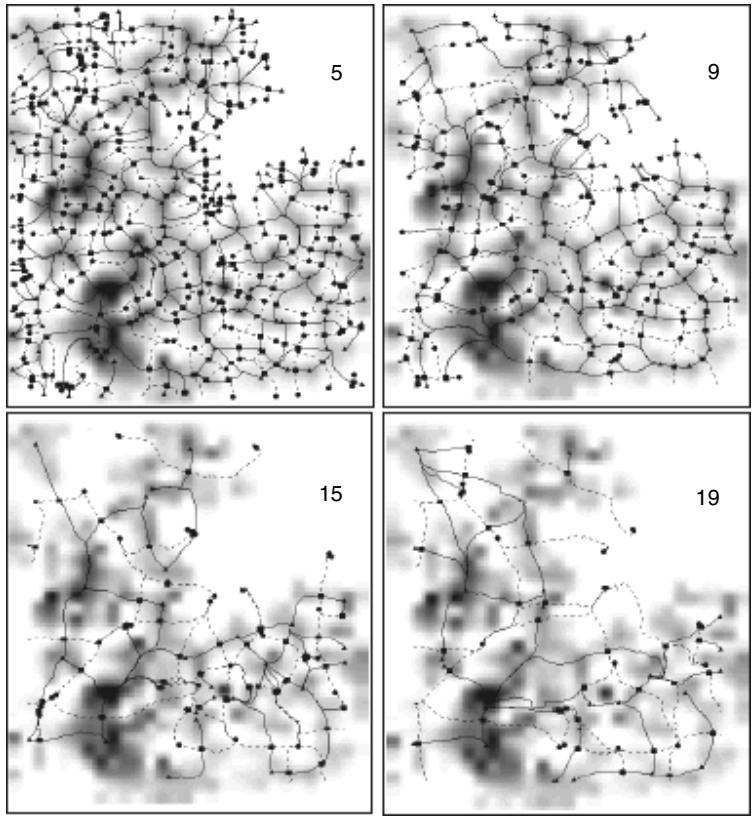

Figure 10.8 *Scale dependency of the surface network in the case of the 16 to 59 years age group population density surface. Note how the varying extraction window size (number on the top-right) influences the detection of features of varying geographic extent. In this case, therefore, the surface network representation provides an objective method for the identification of age group distribution structure*

10.4 CONCLUSIONS, DIRECTIONS, AND PROPOSALS

There is no denying the fact that geovisualization is an inexact science. Some researchers will in fact argue in favour of keeping geovisualization as informal and open-ended in order to preserve the exploratory spirit. We believe that while graphical methods for visualisation should draw on appropriate theoretical literature and exhibit graphic logic, the degree of success with which the process of visualisation is aided by graphic tools lies in the "eye of the beholder". This uncertainty could be the cause of nightmares to visualisation software developers trying to develop the most effective visualisation system. These efforts are questionable when experimental and theoretical evidences in the literature have suggested that human visual processing system does not have the propensity to interpret complex animated sequences particularly successfully. This is because of our limited cognitive capabilities for processing parallel streams of more than

7 ± 2 information (Miller, 1956). In the words of Morrison et al. (2000), "The drawback of animation may not be the cognitive correspondences between the conceptual material and the visual situation but rather perceptual and cognitive limitations in processing a changing visual situation". Here we have endeavoured to demonstrate that the combination of methods employed in various related disciplines to support visualisation offers some opportunity for solutions to this situation. A generic approach is introduced through which various data transformations are applied in a manner that corresponds with established cartographic practice. Techniques have been demonstrated using examples depicting ordered variations in time, attribute, and scale. The aim of these techniques is to draw parallels between existing cartographic practice and opportunities for information visualisation that address identified limitations in processing animated sequences of surfaces for prompting thought and insight. The proposal can be summarised as follows:

(i) Increasing the inter-frame spatial and attribute continuity by removing small-scale variations and focusing on broader trends. A clear parallel exists between this process and that of smoothing in static cartography.
(ii) Highlight the information content by way of a surface network representation. The aim is to make the significant information about the surface explicit, an important objective under the identified limitations of animated cartography.

We also hope that we have highlighted the opportunity for a more generic application of cartographic techniques to the complex graphics that we generate in our efforts to gain insight into dynamic and multifaceted phenomena. We have achieved this by drawing analogies between the techniques developed in cartography to highlight information in maps and our own efforts to generate tools and techniques that make animations suitable for visualisation.

We have developed a working software to demonstrate our ideas. It can be accessed by the reader in order to assess the ideas presented and their implementation (URL #8). The software is available for an online informal evaluation similar to the way recommended by Blok and others (URL #9). We anticipate extending the software further in response to formative evaluation. We plan to add functionalities that include increasing the levels of interactivity in SNV by adding a graphic lag to the coloured symbols and intelligent zooming that relates to particular features. One of our long-term objectives is to integrate the different techniques into single software so that various decisions could be implemented and visualised in real time. The current version of SNV is "loosely coupled" according to Rhyne's (1997) model of levels of integration software for GI processing and visualisation, a typical situation when developing rapid prototypes and experimenting with ideas.

Finally, we believe that the suitability of the visualisation (tools and representations) should be evaluated in relation to their qualities for particular applications. There are no established universal guidelines for interactive environments and theories of interactive geovisualization; therefore, any formal evaluation of prototype implementation requires a sympathetic appreciation of the deliverables of such prototype, at the same time maintaining a sceptical outlook to assess the claims of a visualisation framework. In any event, the success of any human–computer interaction (e.g. geovisualization)

depends not only on the capabilities of the software systems but also on the willingness of the human to adapt to the computing environment (Dunne, 1999).

ACKNOWLEDGEMENTS

We would like to thank Jonathan Raper for data, Jo Wood for software, and David Martin for his comments. Population density surface datasets were obtained from the 1991 Census, Crown Copyright. ESRC/JISC purchase.

11

An Application of Surface Networks in Surface Texture

Paul J. Scott

11.1 INTRODUCTION

In practice, all engineered surfaces depart to some extent from being atomically flat. They contain surface features such as peaks, valleys, ridges, course lines, and so on, which may vary in both height and spacing. The predominate surface patterns are primarily determined by the surface creation processes such as grinding, milling, etching, turning, and so on. The scale of the surface texture also depends on the surface creation process. Typical values for peak-to-valley height differences for precision-engineered surfaces are of the order of a micrometre. Typical surface wavelengths for surface texture are of the order of a micrometre to the order of a millimetre. In essence, surface texture is just like a natural landscape but at a much smaller scale.

Until recently, surface texture of engineered surfaces was analysed by measuring profiles across the surface and characterising the features contained within these profiles. Recent developments in surface texture instrumentation make it possible to measure small areal patches from the surface, in order to analyse the surface texture. Typically, these patches are of the order of a millimetre square and contain from 256 by 256 to 4,096 by 4,096 height values in a square lattice. Figure 11.1 shows a pseudo-photograph (computer generated photograph) of the surface of a grinding wheel reconstructed from 512 by 512 measured height values from a 1 mm $\times$ 1 mm patch. This example is extremely rough, with a peak-to-valley height difference of 263 μm.

The rest of this chapter is organised into the following structure: Section 11.2 gives a brief introduction to the reasons why surface texture is characterised. Section 11.3

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

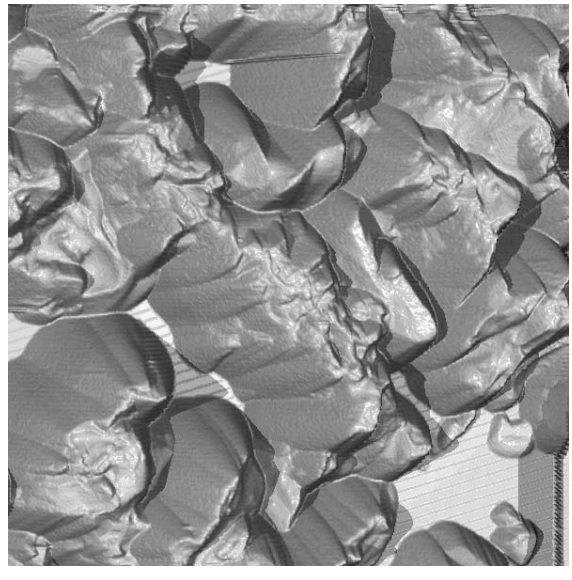

Figure 11.1 *Grit on a worn grinding wheel* $1\,mm \times 1\,mm$

discusses the pattern analysis approach to surface texture characterisation and introduces surface networks, used in segmentation of the surface, in preparation for pattern analysis. The two main problems associated with segmenting surface texture data, namely, edge effects and over-segmentation, are discussed and solutions offered. Segment combination is discussed in some detail and is seen as a two-stage process, namely, identification of the peaks and pits to be kept (significant peaks/pits) and pruning out the other peaks and pits (insignificant peaks/pits). Section 11.4 gives a generic overview of the necessary and sufficient properties that classification of a set of events into significant and insignificant must satisfy in order to give unique and stable results. Section 11.5 discusses different approaches to pruning a change tree. Two examples of the described approach used to solve practical problems are described in Section 11.6. Finally, in Section 11.7, concluding remarks and acknowledgements are offered.

11.2 SURFACE TEXTURE CHARACTERISATION

There are two main uses for surface texture analysis:

- Control of the manufacturing process, which can be further subdivided into the following:
  - *Process monitoring*: used to ensure that a manufacturing process is within acceptable limits.
  - *Process diagnostics*: used when setting up a manufacturing process or when things go wrong and the cause of process problems are required to be diagnosed.
- Control of the functional performance of the component, which can also be further subdivided into the following:

- *Functional prediction*: simulations using the measured data can be used to predict the functional performance of a component.
- *Functional diagnostics*: mainly used when a component fails to perform its desired function and the cause of failure is required to be diagnosed.

Traditional surface texture characterisation, as defined in ISO 4287 (1997), uses surface texture parameters based on statistical attributes, such as peak-to-valley height, root-mean-square and so on, to characterise the cloud of measured height values. These surface texture parameters are termed *field parameters*. Field parameters were primarily developed for monitoring the production process, and achieve this aim very successfully.

Unfortunately, when process or functional diagnostics are required, field parameters are very blunt instruments. A medical analogy is useful to illustrate this point. Many field parameters such as peak-to-valley height are analogous to taking a patient's temperature – a high temperature indicates that something is wrong but it could be anything from a cold to cancer. Field parameters are not very diagnostic.

The approach adopted here for surface texture diagnostics is based on pattern analysis. Pattern analysis is used to assess and characterise the patterns contained in the surface texture. First, segmenting the surface texture into "features" using surface networks, and then statistically characterising attributes of these features and/or the relationships between them achieve this. Parameters that characterise surface features and their relationships are termed *feature parameters*. Continuing the medical analogy, characterising features of the patient (sore throat, running nose, chest pains, shadow on a chest X-ray etc.) is diagnostic.

In summary, *field parameters* (statistical characterisation of a cloud of points) are very good for process monitoring but they are not as diagnostic as *feature parameters* (characterisation of surface features and their relationships).

11.3 PATTERN ANALYSIS OF SURFACE TEXTURE

In order to use pattern analysis for surface texture we need to define the texture primitives, segmentation, and structural relationships. The approach we adopt is based on using critical points, lines, and areas as the texture primitives (Schalkoff, 1992).

11.3.1 Texture primitives and segmentation

More than a hundred years ago, Maxwell (1870) proposed dividing a landscape into regions consisting of hills and regions consisting of dales. A Maxwellian hill is an area from which the maximum uphill paths lead to one particular peak, and a Maxwellian dale is an area from which the maximum downhill paths lead to one particular pit. By definition, the boundaries between hills are course lines (watercourses), and the boundaries between dales are ridge lines (watershed lines). Maxwell was able to demonstrate that ridge and course lines are the maximum uphill and downhill paths emanating from saddle points and terminating at peaks and pits. Recently, the Maxwellian dale (watershed lines) has emerged as the primary tool of mathematical morphology of

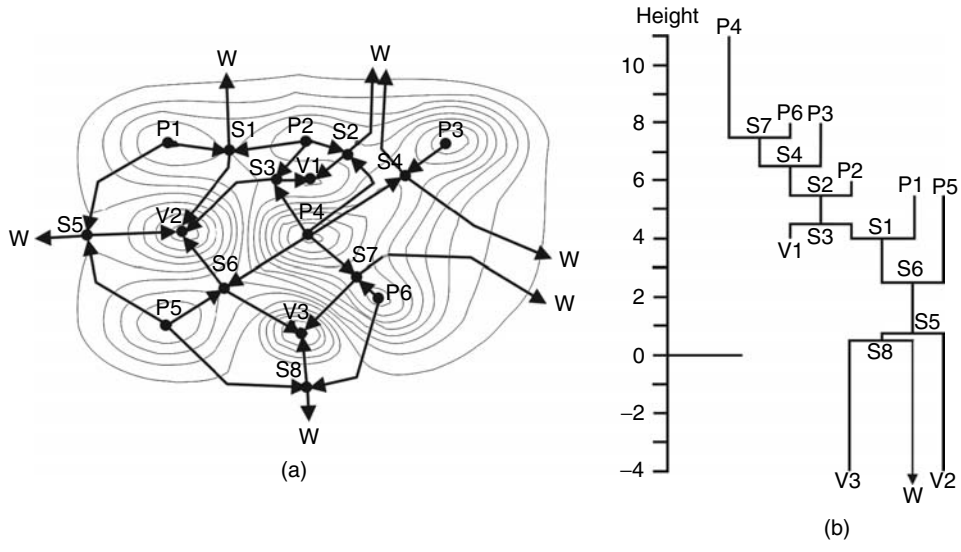

Figure 11.2 *(a) Primitives – critical points, lines, and areas; (b) associated change tree. Key: P = peak, V = pit, S = saddle point; ridgelines connect peaks to saddle points; courselines connect pits to saddle points*

image segmentation, as preparation for pattern analysis. It can easily be seen that the segments are the hills and dales. Figure 11.2(a) gives a schematic diagram of critical points and lines defining hills and dales on artificial data.

Unfortunately, segmenting surface texture data or image into Maxwellian dales/hills is often disappointing, as the surface/image is over-segmented into a large number of insignificant tiny, shallow dales/hills rather than a few significant large dales/hills. This is not a result of any algorithm used but is the nature of the surface texture data itself. What is required is to merge the insignificant dales/hills into larger significant dales/hills.

It is proposed to extend Maxwell's definitions and to define a dale as consisting of a single dominant pit surrounded by a ring of ridge lines connecting peaks and saddle points, and to define a hill as consisting of a single dominant peak surrounded by a ring of course lines connecting pits and saddle points. Within a dale or hill there may be other pits/peaks but they will all be insignificant compared to the dominant pit/peak.

It is also important to consider edge effects. We require the combined segments near the edge to have the same or similar attributes to those near the centre. Ockham's Razor (*non sunt multiplicanda entia praeter necessitatem*–entities are not to be multiplied beyond necessity) is used to extend contour lines outside the area of interest in such a way that a minimum number of new critical points are created. Ockham's Razor leads to two possible solutions called the *virtual pit* and the *virtual peak*, each being the dual of the other. In this chapter, the concept of the virtual pit is adopted (see Takahashi, et al., 1995 for more details). A virtual pit is assumed to be a point of infinite depth to which all the boundary points are connected. (A virtual peak is assumed to be a point of infinite height to which all the boundary points are connected.) The adoption of the

virtual pit greatly simplifies the resulting discussion and also allows the network to satisfy the Euler–Poincaré formula:

$$\#(\text{peaks}) + \#(\text{pits}) = \#(\text{saddles}) + 2 \tag{11.1}$$

11.3.2 Structural relationships

A useful way to organise the relationships between critical points in hills and dales and still retain relevant information, is that of a change tree. Kweon and Kanade (1994) introduced the concept of a topographic change tree to describe the connectability of a surface. The change tree represents the relationships between contour lines from a surface and is one example of a more general topological object called a *Reeb Graph* (Takahashi et al., 1995). The vertical direction on the change tree represents height. At a given height, all individual contour lines are represented by a point that is part of a line representing that contour line continuously varying with height. Saddle points are represented by the merging of two or more of these lines into one, peaks and pits are represented by the termination of a line (see Figure 11.2(b)).

Consider filling a dale gradually with water. The point where the water first flows out of the dale is a saddle point. The pit in the dale is connected to this saddle point in the change tree. Continuing to fill the new lake, the next point where the water flows out of the lake is also a saddle point. Again, the line on the change tree, representing the contour of the lake shoreline, will be connected to this saddle point in the change tree. This process can be continued and establishes the connection between the pits, saddle points, and the change tree. By inverting the landscape, peaks become pits, and so on, and a similar process will establish the connection between peaks, saddle points, and the change tree.

11.3.3 Areal combination

In practice, change trees can be dominated by very short contour lines, due to noise and so on, which hinders interpretation (over-segmentation of the surface/image by Maxwellian hills and dales). A mechanism is required to simplify the change tree, which reduces the noise, but retains relevant information. Areal segment combination is such a mechanism, leaving the change tree simplified but still containing relevant information. Areal segment combination is seen as a two-stage process, namely, identification of the peaks and pits to be kept (significant peaks/pits) and pruning out the other peaks and pits (insignificant peaks/pits).

The following is an outline of the areal segment combination algorithm. The simplified algorithm presented here assumes that the virtual pit condition has been applied. An algorithm without the virtual pit condition was presented in (Scott, 1998).

Step 1 Assuming the virtual pit condition, find all Maxwellian hills and dales and their associated peaks and pits and generate the full change tree.

Step 2 Classify all peaks and pits as significant or insignificant according to the function of the surface.

Step 3 Prune out the insignificant peaks and pits from the change tree. That is to say, combine the peaks and pits with the adjacent saddle point they are connected to in the change tree.

The resulting change tree will indicate the significant peaks, pits, edge peaks and pits, and the relationships between them. Hence, the change tree has been pruned, reducing the noise, but retaining relevant information.

Not all attributes from a hill or a dale can be used to determine significant or insignificant peaks and pits. The process of determining which peaks/pits are significant or insignificant must satisfy certain mathematical properties to give unique stable results. This is discussed at a very generic level in the next section. Here the set of "events" can be the set of peaks or the set of pits with the motif function labelling each peak/pit as either significant or insignificant.

11.4 WHICH SEGMENTS/MOTIFS TO COMBINE

A motif function consists of splitting a set of "events" into two distinct sets called the *significant events* and the *insignificant events*. For the motif function to give unique and stable results the motif function must satisfy the following three properties (Scott, 1992):

P1 Each event is allocated to one and only one of these two sets (i.e. the set of significant events and the set of insignificant events).
P2 If a significant event is removed from the set of events, then the remaining significant events are contained in the new set of significant events.
P3 If an insignificant event is removed from the set of events, then the same set of significant events are obtained.

It can be shown (Scott, 2004) that all motif functions that satisfy these three properties can be mapped one to one onto a certain subset of morphological closing filters. Morphological closing filters are widely used in image analysis. They are set functions with the following three defining properties (Serra and Vincent, 1992):

1. All sets are subsets of their own closings.
2. A closing of a closing of a set is the closing of the original set.
3. A closing of a subset is a subset of the closing of the original set.

The particular subset of the closing filters that the motif functions map onto are the closings with the following properties (Scott, 2004):

If two sets of events give the same closing, then their intersection also gives the same closing.

For any closing that satisfies this property we can map it one to one onto a particular motif function as follows.

For any set of events, consider the smallest subset of this set that gives the same closing as the original set of events. It can be shown that this particular subset is unique and well defined and corresponds to the set of significant events and its complement, with

respect to the set of events, corresponds to the set of insignificant events. The inverse mapping is also well defined. Proofs of these results can be found in (Scott, 2004).

This is a powerful result since it allows one to construct all possible motif combination functions from the morphological closing filters whose properties, including how to generate all possible finite closing filters, are very well known (see Serra and Vincent, 1992).

The classification of peaks and pits as significant or non-significant must be a motif function that satisfies the above three properties to give stable results.

11.5 CHANGE TREE PRUNING

In the literature, there are now several publicised references to methods that are analogous to pruning a change tree (Wolf, 1991a, Bleau and Leon, 2000, Barré and Lopez, 2000).

Wolf (1991a) presents a method to prune a Pfaltz graph. A Pfaltz graph is another topological object that can be used in the efficient calculation of a change tree (Takahashi, 1995). Hence, pruning a Pfaltz graph is equivalent to pruning a change tree.

Very recently, methods to merge watersheds (Maxwellian dales) have appeared in the literature (Bleau and Leon, 2000, Barré and Lopez, 2000). Watershed merging is equivalent to change tree pruning only if the triangulation of the lattice is assumed to be a continuous surface (i.e. triangular facets).

11.5.1 Wolf pruning

One first calculates for each peak and pit the height difference between the peak or pit, and the adjacent saddle point they are connected to on the change tree. Wolf's pruning method consists of finding the peak or pit with the smallest height difference and combining it with the adjacent saddle point on the change tree. The other peak or pit also connected to this saddle point is now connected to another saddle and so its height difference is adjusted to reflect this. The process is then repeated with that peak or pit with the smallest height difference compared to its adjacent saddle point on the change tree being eliminated until some threshold is reached. This threshold could be when all remaining height differences are above a fixed value or alternatively when a fixed number of peaks or pits are left. It is easily proved that both criteria lead to a motif function that satisfies the three required properties given in Section 11.4. Using the change tree given in Figure 11.3, P6 to S7, P2 to S2, and V1 to S3 all take the value of the smallest height difference of 0.5. Pruning these leads to the change tree given in Figure 11.3.

Using Wolf pruning until five peaks and five pits are left on a surface gives a stable definition of the ten-point height parameter defined as the average height difference between the five "highest peaks" and the five "deepest valleys". These peaks/pits may not be the highest/lowest but they will be the tallest. Note that Mount Everest may be the highest mountain on the Earth but it is not the tallest (base to peak); Mauna Kea in Hawaii is the tallest.

This example illustrates that attributes of a hill or dale can be used to create a functional motif function that can be used for pruning the change tree as long as they satisfy the three properties given in Section 11.4.

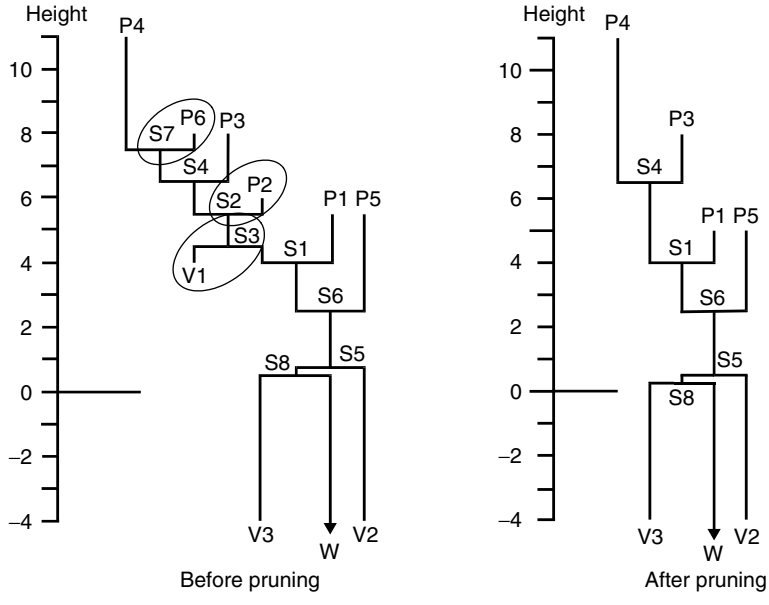

Figure 11.3 *Wolf pruning on a change tree*

11.5.2 Watershed merging (merging Maxwellian dales)

Automatic watershed merging has recently been investigated in response to over-segmentation of the Maxwellian dales in images, as preparation for pattern recognition. The basic idea is to replicate the process of watershed merging that takes place when rain falls over a real landscape: smaller watersheds fill progressively until an overflow occurs. The water then flows to a nearby larger or deeper watershed, in which the watersheds that overflow are merged.

Algorithmically, the Maxwellian dales are found using an algorithm based on immersion simulations (Vincent and Soille, 1991). The insignificant dales are detected and filled in up to their lowest overflow height and the Maxwellian dales are again found from the filled-in surface. This procedure is repeated until no insignificant dales are left.

In practice, the pixels belonging to a filled-in insignificant dale are often distributed between two or more significant dales and so is not equivalent to a change tree pruning operation. The distribution of pixels from insignificant dales is entirely due to the discrete nature of the data and the current way the insignificant dales are filled. At a non-filled-in pixel that neighbours the filled-in region, the local slope alters. This can lead to the merging of insignificant dales with two or more significant dales.

On continuous data filled-in insignificant dales are always merged with one and only one other dale (assuming all saddle points have unique heights) and so are equivalent to a change tree pruning operation. A modification of the current filling-in algorithm that does not lead to local slope changes at non-filled-in pixels is required to simulate continuous data.

11.6 EXAMPLES

11.6.1 Example 1 – grinding wheel (identifying active grains)

The cutting edges on a grinding wheel are geometrically undefined in location and shape. In order to ascertain the qualitative measurement of cutting edges, it is necessary to develop techniques to identify the individual cutting edges from topographic data.

Blunt and Ebdon (1996) describe an approach based on using local peaks to count the number of cutting edges. Unfortunately, using the number of local peaks produces an overestimate (409 peaks in Plate 6(a)). Blunt and Ebdon (1996) recognised this counting problem and suggested sub-sampling the measured data to achieve the "correct count". The optimal sub-sampling corresponds to approximately one peak on each grinding wheel grain.

Hence, changing the grain size changes the distance of the optimal sub-sampling. Owing to the non-uniform packing and grain shapes, this may vary considerably within a given grinding wheel. Wolf pruning at different thresholds, Plate 6(b–d), produces different counts of the number of significant hills. By comparing these counts to manual count, it was determined that Wolf pruning at 5% (i.e. 5% of the peak-to-valley of the data) produces the correct count for all grain sizes. For Plate 6(c) this produces 60 peaks. Wolf pruning has the added advantage that the significant peak in each segment is given, allowing further analysis. For example, a height analysis can distinguish which of these peaks could be active, that is, come into contact with the workpiece. Thus Wolf pruning can help in the characterisation of grinding wheels.

11.6.2 Example 2 – anodised-extruded aluminium

Plate 7(a) shows 0.5×0.5 mm portion of the surface of anodised-extruded aluminium. The texture consists of three types of features: extrusion marks, crystal boundaries of the anodising and isolated deep pits. The extrusion marks are easily seen running across the surface, as are the connected crystal boundaries of the anodising. The anodising is porous and results in deep isolated pits within each crystal. The manufacturers of this surface require a separate characterisation of these three types of features in order to control the manufacturing process, especially between the deep isolated pores and the connected valleys at the crystal boundary.

In order to control the production process, a sample of the anodised extruded aluminium is measured and inspected. Currently, this inspection is carried out by eye. Wolf pruning at 15% (Plate 7(b)) discriminates between the connected crystal boundaries and the deep isolated pits. Thus, Wolf pruning allows the inspection process to be automated.

11.7 CONCLUDING REMARK

In this chapter, an application of surface networks in surface texture has been presented. The pattern analysis approach to surface texture characterisation is outlined and surface networks, used in segmentation of the surface, in preparation for pattern analysis is introduced. The two main problems associated with segmenting surface

texture data, namely, edge effects and over-segmentation, are discussed and solutions offered. Segment combination is discussed in some detail and is seen as a two-stage process, namely, identification of the peaks and pits to be kept (significant peaks/pits) and pruning out the other peaks and pits (insignificant peaks/pits). A generic overview of the necessary and sufficient properties that classification of a set of events into significant and insignificant must satisfy in order to give unique and stable results is also given. The different approaches to pruning a change tree are also discussed. Finally, two examples of the described approach used to solve practical problems are presented.

ACKNOWLEDGEMENTS

The Author wishes to thank his colleagues at Taylor Hobson and The Centre for Precision Technologies at the University of Huddersfield for their co-operation in the preparation of this chapter. Finally, the author would like to thank the directors of Taylor Hobson for granting permission to publish.

12

Application of Surface Networks for Fast Approximation of Visibility Dominance in Mountainous Terrains

Sanjay Rana and Jeremy Morley

12.1 INTRODUCTION

Visibility analysis of terrain is now an integral part of many disciplines (Rana, 2003b). Some typical applications include the military plans (e.g. watch towers, troop movements, flight paths – Franklin et al., 1994), communication/facilities allocation (e.g. TV/Radio transmitters – Lee, 1991, De Floriani et al., 1994, Kim et al., 2002), landscape analysis (e.g. visibility graphs – O'Sullivan and Turner, 2001) and environmental modelling (e.g. terrain irradiation – Wang et al., 2000a).

Most existing research has focused on broadly two main aspects of terrain visibility analysis, namely, visibility index[1] computation time and accuracy of the viewshed (area covered by the visible terrain). While formal methods for modelling viewshed uncertainty were established early in the last decade (e.g. Fisher, 1991, 1992, 1993), the search for algorithms to optimise visibility computation continues to remain an attractive topic for considerable research (e.g. Izraelevitz, 2003, Rana, 2003a) and is also the motivation for this work.

Without any risk of generality, if we ignore the algorithmic and implementation-related (e.g. hardware) factors that influence the computation, the computation time of visibility index is directly proportional to $O(ot)$, where o is the number of observers

[1] Visibility index is generally expressed in terms of the number of visible points or the visible ground area.

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

(viewpoints) and t is the number of targets on the terrain. In a so-called *Golden Case*, all the points, n, on terrain are used as observers and targets, that is, the visibility indices of all points on terrain is computed by drawing lines of sights (LOSs) to all other points on the terrain. Thus, the computation time in a *Golden Case* is $O(n^2)$ because $o = t = n$, which is clearly exhaustive and time consuming. On the other hand, optimised visibility index computation methods are based on strategies to reduce the observer–target pair comparisons, for example, by choosing a polyhedral terrain model (e.g. Triangulated Irregular Network or TIN – De Floriani and Magillo, 1994) instead of a grid and by using algorithmic heuristics (Franklin et al., 1994, Franklin, 2000, Wang et al., 2000b). Accordingly, there are two main types of optimisation strategies, namely the *Reduced Observers Strategy* and *Reduced Targets Strategy*. As the names suggest, *Reduced Observers Strategy* and *Reduced Targets Strategy* reduce the observers (e.g. random sampling of observers) and targets (e.g. limiting the maximum visibility distance as in horizon culling) parts of the computational load respectively. The visibility indices derived in a *Golden Case* and either of the optimisation strategies are respectively referred to as the *Absolute Visibility Indices* (AVI) and *Estimated Visibility Indices* (EVI) of terrain points.

In many applications, however, finding out the location of visibly dominant (i.e. visibility dominance) observers has more practical use than exact visibility indices of observers (Franklin, 2000). In addition, as seen above, visibility indices can be biased by the number of targets. Therefore, the aim of most visibility analyses is to identify visibly dominant locations in the terrain. There are potentially many ways for calculating visibility dominance. In this work, visibility dominance is calculated by normalising the visibility index as follows:

$$d_i = \frac{v_i - v_{\min}}{v_{\max} - v_{\min}} \tag{12.1}$$

where v_i and d_i are respectively the values of visibility index and visibility dominance at an observer i. $v_{\min}$ and $v_{\max}$ are respectively the minimum and maximum visibility indices on the terrain.

In this chapter, we propose methodologies, with examples, which employ the surface network data structure (see Part I for a background) both in the *Reduced Observers Strategy* and *Reduced Targets Strategy* for a fast approximation of visibility dominance. Our proposal is based on the findings of Lee (1992), who reported that fundamental topographic features, namely, peaks, pits, passes, ridges, and channels dominate the visibility of other ground locations and therefore could be good viewpoint sites. In a previous article (Rana, 2003a), we demonstrated that fundamental topographic features, due to the exhaustive and characteristic spatial coverage (especially in mountainous uplands), are the ideal set of targets to reduce the targets part of the visibility index computation load. In essence, we employed the *Reduced Targets Strategy* to reduce the visibility index computation time by drawing the LOS from observers to only the fundamental topographic features. For brevity, we will use the term *topographic features* in place of the *fundamental topographic features* in the following part of the chapter. In this chapter, we present another implementation of the *Reduced Targets Strategy* using topographic features and evaluate whether it is possible to use topographic features as part of the *Reduced Observers Strategy* for reliable visibility dominance pattern.

As mentioned earlier, because of the selective sampling of observers and targets, optimised algorithms will either underestimate or overestimate the visibility dominance of non-topographic feature points on terrain. This uncertainty, arising because of the level of sampling (abstraction), is closely similar to the uncertainties referred as *Object Generalisation* (Weibel and Dutton, 1999). To our knowledge, no proposals existed for assessing such uncertainty in visibility analysis literature until the methods discussed in this chapter were first presented in our earlier work (Rana, 2003a). Our aim was to evaluate whether the overall visibility dominance pattern was realistic, albeit approximated. The following section proposes two simple methods based on an iterative comparison between the AVI and EVI for assessing this uncertainty.

12.2 PROPOSAL

A target is considered visible if a LOS from an observer can be drawn to it without an obstruction by an intermediate point (an exception is provided by (Wang, 2000b), who used reference planes to establish the visible areas). The most common approach in previous *Reduced Targets Strategy* based methods (e.g. Franklin et al., 1994) has been to draw the LOS from an observer to an arbitrary small number of randomly located targets on the terrain. In the earlier work (Rana, 2003a) based on small study areas, we demonstrated that the computation time can be reduced substantially without any significant loss of visibility information if the LOSs are drawn only to the topographic features. Of course, the underlying assumption of this proposal is that the terrain is not devoid of topographic features. This is true for mountainous terrain except in upland plateaus, although in this case the visibility indices will be similar everywhere.

The methodology for the computation of visibility indices using topographic feature as targets consisted of three steps: (a) extract the topographic features; (b) compute the visibility dominance of each point using the topographic features as targets; and (c) assess the uncertainty in the visibility dominance.

As a *Reduced Observers Strategy* to reduce computation time, we evaluated whether the visibility dominance of non-topographic feature points could be derived by interpolating the visibility dominance of topographic feature points. In other words, we assume a spatial autocorrelation of the visibility dominance, that is, points near visually dominant points will tend to have a higher visibility dominance. The *Reduced Observers Strategy* also consists of four steps: (i) extract the topographic features; (ii) compute the visibility dominance of only the topographic features by drawing the LOS to all the points in terrain; (iii) interpolate/extrapolate the visibility dominance of other points; and (iv) assess the uncertainty in the visibility index.

12.2.1 Methodology

12.2.1.1 Extraction of topographic features

A number of approaches have been proposed for the automated extraction of surface networks from DEMs and TINs (e.g. see Takahashi et al, 1995, Wood, 1998). Refer

to Chapters 3 and 4 for more information on feature extraction. We decided to use the feature extraction method of Wood (1998; see also Chapter 4), partly on the basis of the advantages he outlined against the other methods and partly because of its easy availability in the user-friendly freeware LandSerf (Wood, 1998).

It is clear that the success of our strategies depends upon the accuracy of the non-trivial topographic feature classification. It is well known that most automated topographic feature extraction methods are vulnerable to the noise in the DEM (Jenson and Domingue, 1988) and, most importantly, have scale dependency limitation (Wood, 1999). While smoothing the DEM before extracting the features can eliminate the first limitation, the latter seems to remain a difficult conceptual problem yet to be completely solved. Because of scale dependency, the automated feature extraction identifies features only at a certain scale (e.g. features of a fixed geographic extent), while features at other scales remain undetected. Therefore, the assessment of an appropriate scale for the particular DEM requires iterating through a number of *feature extraction scales* (e.g. in LandSerf, one could achieve this by iterating with a different window or kernel size for the feature extraction with visual verification).

Lastly, although the fundamental topographic features are a significantly reduced number of targets, there may still be too many for certain terrains, for example large desiccated DEMs, and thus still lead to long visibility index computation time. Two simple ways (amongst perhaps many others) of reducing the number of topographic features are – (i) resample the topographic features along ridges and channels by a skip interval, and (ii) limit the topographic features to certain scales. Rana (2003a) provides examples of using such sampling for various purposes in visibility analyses.

12.2.1.2 Visibility analysis

The study area for the current work is a 100-m resolution raster DEM (5548 cells) of the Cairngorms in Scotland (Figure 12.1(a)). Note that our proposal is generic and could be applied to an irregular terrain model such as TIN. Visibility analysis was carried out in ESRI's ArcView GIS and all the parameters were the defaults of the

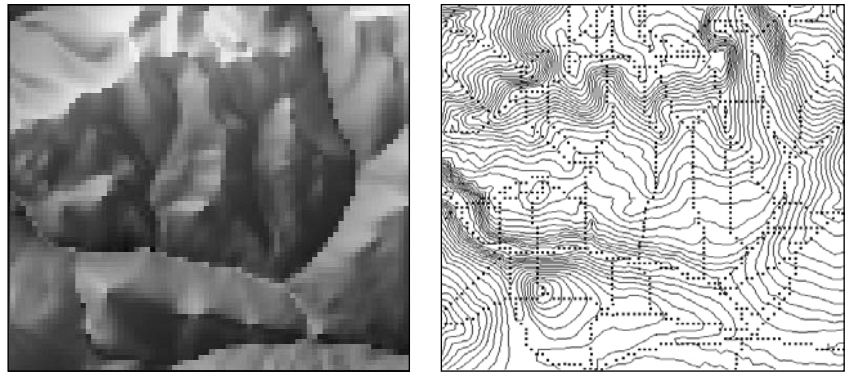

Figure 12.1 *(a) Hill-Shaded DEM of SE Cairngorm Mountains, Scotland. Minimum elevation = 395 m and maximum elevation = 1054 m and (b) 910 topographic feature targets, overlaid on DEM contours*

Visibility Request in ArcView. In the experiment, the observer eye level is at 1 m above the local ground level and the targets are at local ground level. The observer is capable of seeing from the ground zero until infinity (i.e. no horizon culling), across the full range of azimuths and from the zenith to nadir. The experiments were done on a 1-GHz Intel-Pentium III processor based–personal computer, with 512-MB RAM. We recorded the CPU time taken by ArcView for each visibility computation.

12.2.1.3 Interpolation of visibility dominance

As part of the *Reduced Observers Strategy*, Natural Neighbours Interpolation (NNI) (Sibson, 1981) was used to derive the visibility dominance at non-topographic feature points. NNI is a simple, robust, and objective (no requirements for search radius, neighbourhood type) method for interpolation in two dimensions. NNI produces a surface everywhere in the convex hull of the point set that is continuous in slope except at the points. NNI available in ArcView was used to derive surface at the same spatial resolution as DEM, that is, 100 m.

12.2.1.4 Uncertainty assessment

Geospatial uncertainty modelling generally involves the derivation of deviations between the measured and estimated values with the eventual aim of generating models that could predict the behaviour of causes of uncertainty (e.g. systematic or random) and the process under observation. For example, Fisher (1991, 1992, 1993) suggested methods based on Monte Carlo analysis for assessing the effect of noise in a DEM and the robustness of algorithms for computing visibility indices. In our case, the uncertainty is essentially the deviation between the absolute and estimated visibility dominance values arising because of the selective sampling of targets and observers. The only previous example known to us, which dealt with the estimation of uncertainty in a *Reduced Targets Strategy* is that of Franklin et al. (1994). They compared the visibility indices of an arbitrary number of spatially distributed locations on the terrain, computed from their exhaustive R2-visibility algorithm (similar to our Golden Case), with their optimised methods. Though the results are encouraging, their sampling methods (i.e. the selection of the test points) could not be regarded as formal and objective for two important reasons. Firstly, since there is no prior knowledge about the statistical distribution of the visibility pattern, it is not possible to estimate the number of random points required to fully capture the sensitivity of the visibility dominance distribution of a terrain. However, the choice of the number of random points is critical, as it will dictate the computation time. Secondly, since viewshed at a location is generally anisotropic, that is, the visual spread varies according to directions, the random locations could therefore bias the uncertainty estimation. One of the conclusions of this work is that the visibility pattern is highly dependent upon the spatial distribution and the number of the random points. It should be noted that the aim of uncertainty assessment in this work was only to quantify the deviations and did not involve any form of predictive modelling based on deviations.

We used the following two methods for the uncertainty assessment based on a slight modification of the Franklin et al. (1994) method:

Method 1: Spatial correlation between absolute and estimated visibility dominance

This method compares the similarity between the overall visibility pattern shown by absolute and estimated visibility dominance values. The comparison can be based on either the deviations at the topographic feature locations or random locations as follows:

Type 1: Absolute versus estimated visibility dominance at topographic feature locations:

(i) Calculate the absolute visibility dominance of the topographic feature locations by drawing the LOS to all the terrain points.
(ii) Calculate the correlation coefficient between two sets of absolute and estimated visibility dominance values. The correlation coefficient should suggest the similarity between the two visibility patterns. This method is similar to that of Franklin et al. (1994) except that the definition of our sample locations is objective and more *natural*. However, statistically it remains only an approximate test, especially when using exceptional terrains (e.g. mountains next to a plain), where the topographic features are not distributed uniformly across the terrain.

Type 2: Absolute versus estimated visibility dominance at random locations:
Unlike the Type 1 method, this method is relatively more exhaustive but time consuming. It is an abridged form of the Monte Carlo method of uncertainty modelling and involves an iterative comparison between the absolute and estimated visibility dominance at sets of random locations but with the important exception that no subsequent model parameter estimation is done in this method. The steps are as follows:

(i) Generate random sample locations: As mentioned before, since there is no prior knowledge about the visibility dominance distribution it is non-trivial to determine the optimum number of random sample locations sufficient to capture the visibility pattern. We propose, without formal proof, that randomly placed locations equal in number to the number of unique EVI would be sufficient if we assume that no part of the study area is completely hidden from the set of topographic features. Thus, a frequency histogram of the EVI (computed using topographic features) represents unique viewsheds. In other words, we assume that each viewshed will be assigned to at least one sample location.
(ii) Compute the absolute visibility dominance values at the random locations by drawing the LOS to all the points on the terrain.
(iii) Calculate the correlation coefficient between the absolute and estimated visibility dominance values.
(iv) Repeat steps (i) – (iii) a number of times. Again, due to the lack of any prior information about the statistical distribution of the visibility dominance, statistically it is difficult to decide a specific number of iterations. In a practical exercise, it would ultimately depend upon the amount of time available to the researcher for the experiment.
(v) Choose the lowest and the highest correlation coefficient as indicators for the worst- and the best-case approximation. Other statistical measures such as mean and standard deviation of correlation coefficient will indicate the overall approximation.

Method 2: Error in the visibility dominance

In the previous methods, the correlation coefficients only give an indication of the reliability of estimated visibility dominance. However, these do not reveal the level of approximation in the estimated visibility dominance. A simple method for measuring the uncertainty in the estimated visibility dominance is as follows:

$$\text{Average Error } (\%) = \pm \frac{\sum_{i=1}^{n} \frac{|d_i' - d_i|}{d_i}}{n} \times 100 \tag{12.2}$$

Where d_i' = estimated visibility dominance, d_i = absolute visibility dominance and n = number of observers (targets).

12.3 RESULTS

12.3.1 Automated extraction of the topographic features

After iterating with various window (kernel) sizes followed by visual inspection, we found that a 5×5 (500×500 m) window was suitable to extract most linear (ridge, channel) and point (peak, pit, pass) topographic features, present in the Cairngorm DEM (Figure 12.1(a)), where 910 topographic features have been extracted as the optimum targets and observers (Figure 12.1(b)). The automated extraction of the topographic features took less than five seconds.

12.3.2 Visibility analysis and uncertainty assessment

Since our study area was small, we calculated the *Golden Case* visibility patterns of our study areas (Figure 12.2(a)). These visibility dominance patterns were thus the ideal standards, that is, the absolute visibility dominance. It took 537 s to compute the golden case.

12.3.2.1 Reduced targets strategy

It took only 91 s to compute an estimated visibility dominance of the study area based on topographic features. Figure 12.2(b) shows the pattern of the estimated visibility dominance and it is clear from the figures that the overall pattern of the visibility indices is similar to the Golden Case. In fact, as indicated by the correlation coefficient and R^2 values, there is statistically very little difference between the measured and estimated visibility dominance from pixel to pixel (Figure 12.3(a)). The ridges and peaks have high visibility indices compared to the pits, passes, and channels. The average error in the estimated visibility dominance values is $\pm 17\%$ and the residuals are uniform (Figure 12.3(b)), which together prove that we have successfully optimised the computation without losing significant amount of visibility information.

On the basis of Method 1 for uncertainty estimation, Figure 12.4(a) shows the relation between the absolute visibility dominance and estimated visibility dominance at

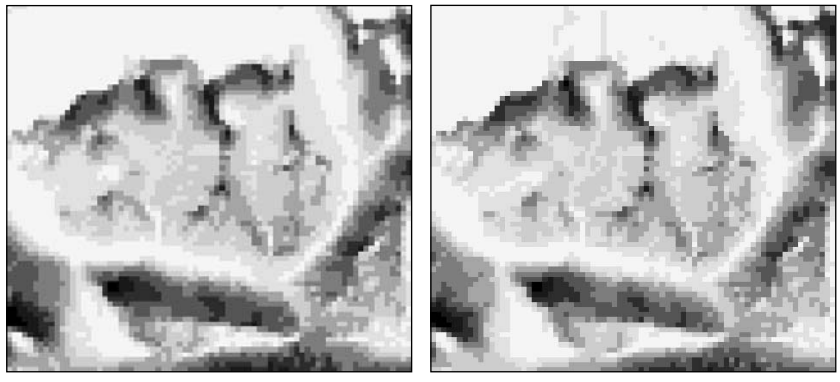

Figure 12.2 *Comparison between the (a) Golden Case–based visibility dominance and (b) topographic features–based estimated visibility dominance. Darker coloured areas have more visual dominance than lighter coloured areas*

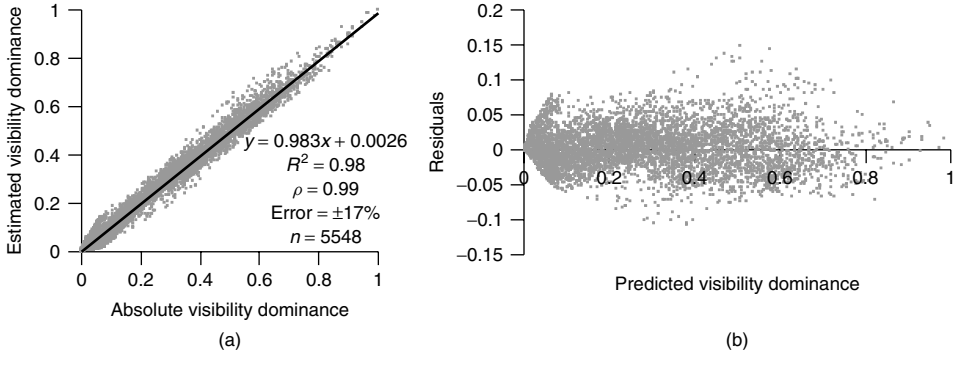

Figure 12.3 *Uncertainty assessment based on the entire DEM. (a) Absolute versus estimated visibility dominance of all locations, and (b) residuals based on the linear regression between absolute and estimated visibility dominance values of all locations*

the locations of topographic features. The strong correlation coefficient of 0.98 suggests that the optimisation successfully represents the overall visibility pattern. As part of method 2 of uncertainty assessment to perform a more exhaustive calculation, we collected 19 sets of randomly located points on the terrain. Each set comprised 418 points (unique number of estimated visibility dominance values). We then calculated the correlation coefficient and error between the measured visibility dominance and estimated visibility dominance for each of these sets of random points. Figure 12.4(b) shows the wide variation in the quality of the estimated visibility pattern at various points on the terrain thus supporting the exercise to validate the quality of the estimated visibility by repeated random sampling. There seems to be no correlation between the error and correlation coefficient, which suggests that these measures qualify different aspects of the visibility dominance pattern.

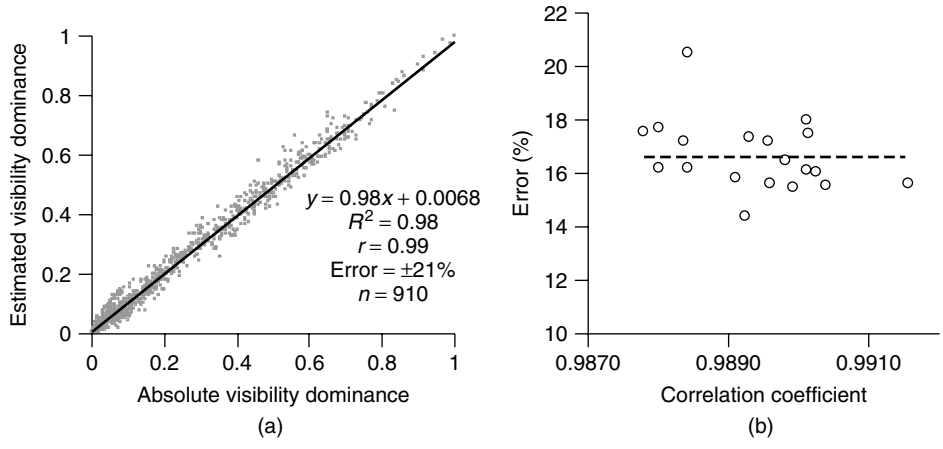

Figure 12.4 *Uncertainty assessment based on selective sampling. (a) Absolute versus estimated visibility dominance of the topographic features and (b) correlation coefficient versus errors at a set of random locations*

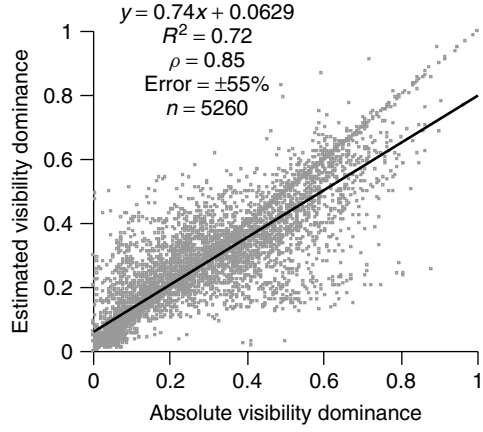

Figure 12.5 *Comparison between the absolute visibility dominance and estimated visibility dominance values based on the reduced observers strategy*

12.3.2.2 Reduced observers strategy

ArcView took only few seconds to interpolate the measure visibility dominance values across the study area. Since NNI is an exact interpolation, method 1 for uncertainty assessment, that is, a comparison between the absolute and estimated visibility dominance values only at topographic features was not appropriate because the deviations would have been zero or negligible. Instead, we performed a single comparison for the entire terrain involving all the measured and the estimated visibility dominance values. Figure 12.5 shows a considerable increase in the error (±55%) but the correlation coefficient (0.85) and R^2 (0.72) values suggest a reasonably strong similarity between the measured and estimated dominance patterns.

12.3.3 Optimisation of computation time

It is clear that we have been able to reduce the computation time substantially by at least five times in our experiments. The optimisation is linear as time saved was merely due to a linear reduction in the number of comparisons unlike other approaches such as by Izraelevitz (2003) in which previous computations are recycled to reduce computation time. The CPU usage could be further optimised by combining the current approaches with further *Reduced Targets Strategies* such as horizon culling.

12.4 CONCLUSION AND FUTURE WORK

In general, there is a compromise between performance and accuracy in any practical visibility computation (Franklin et al., 1994). In this work, we have shown that the use of the fundamental topographic features as optimum targets and observers can be used to decrease the visibility computation time substantially without any significant visibility information loss. This approach is especially useful for a fast approximation of visibility dominance in mountainous terrain. The reduced sampling of the targets on the terrain, however, introduces an uncertainty in the visibility indices of the observers on the terrain.

In the current work, the use of the correlation coefficient and the simple statistical measures such as correlation coefficients and R^2 values as measures of a visibility pattern quality and uncertainty provide only a global pattern matching, but visibility is a directional property. We anticipate developing ways in which we could estimate the visual integrity in our optimised approach. Although our observation that at certain numbers, both topographic-feature targets and random targets would produce similar quality of visibility estimation is based on thorough experimentation of the current study areas, experiments with other DEM will be useful to fully validate this empirical observation.

We believe it is more important to realise that visibility, as a property of terrain location, could not be modelled since it is derived only after a LOS test with other locations. Therefore, it is invariant of the local properties (e.g. elevation, slope, aspect) and global properties (e.g. geographic setting i.e. fault etc.) of a location. Thus based on the current study, we believe that the regression between measured and estimated visibility dominance only provides the information about the similarity or the amount of approximation.

Finally, an interesting intellectual challenge still remains in understanding of the effect of the topographic feature extraction scale on the computed visibility pattern.

ACKNOWLEDGEMENTS

Author wishes to thank Toshihiro Osaragi (Tokyo Institute of Technology), Mike Batty, Daryl Lloyd (University College London), and Young-Hoon Kim, Steve Wise (University of Sheffield) for providing critical feedback and materials support, which substantially improved the original manuscript.

Conclusion

13

Issues and Future Directions

Sanjay Rana

By now, the role of topological surface data structures in a wide variety of morphometric analysis should hopefully be clear to the reader. The work also highlighted a number of issues in the existing approaches to model and generate topological surface data structures. In this chapter, I will first highlight those shortcomings of the topological surface data structures, principally surface network, which in a way also indicate the potential directions for future work.

13.1 SHORTCOMINGS OF SURFACE NETWORK MODEL

Although the surface network model provides a natural and sophisticated representation of surfaces, it has been considered merely as an interesting proposal by many researchers. Pfaltz (1976) himself was aware of the several flaws and commented about the topological properties of the surface network graph that "it is unknown whether these properties are sufficient to guarantee the realizability". It is therefore no surprise that surface network has received little recognition amongst most textbooks as a data structure. I think the surface network model suffers from the following three main drawbacks:

1. Non-representation of all surfaces and surface features.

- *Not all surfaces can be realised as surface networks.* As discussed in Chapter 4, the fundamental restrictive condition with the surface network model is the assumption that surfaces are C^2 continuous everywhere, so that features such as overhangs (e.g. glaciated terrains, dunes, plateaus), holes (e.g. karstic terrains, cracks), break in slope

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana
 ISBN: 0-470-85151-1

(e.g. alluvial fans, scarps, corners) are absent, and the critical points and lines are clearly defined on surface. This condition is clearly an exception than a rule for many surfaces especially for terrains, which are used to generate surface networks. Commonly available digital surface datasets are often full of sensor and interpolation noise (von Minusio, 2002). Therefore, it is clearly not possible to realise the surface networks for all types of surfaces, especially for those that do not have the entire set of critical points and lines required for the surface networks. In this respect, the non-constrained nature of contour trees makes them more suitable to represent the topology of any surface.

- *Problems in representing well-behaved surfaces.* As described in the last point, it is an oversimplification to assume that naturally occurring surfaces are Morse functions, indicating a sense of equilibrium in nature[1]. However, it would also be unrealistic to suppose that it is impossible to derive surface networks, although maybe even in small regions (as discussed in Chapter 5). These exceptional surface patches could be regarded as well-behaved surfaces. Different kinds of limitations exist in representing well-behaved surfaces.

A common concern amongst geomorphologists regarding surface network is that it does not represent many important hydrological features, for example, junctions and bifurcations because the ridges and channels could only meet at the critical points. Although, Wolf (Chapter 2) suggested that the channel junctions and ridge bifurcations could be represented as an infinitesimally closely located artificial pair of pit–pass and pass–peak respectively, Wolf does not prove how to derive the topological connections for the new pass, added at the junction and bifurcation, in order to satisfy Rule P4 (see Section 2.4). The explanation used by Wolf, (1990) to indicate the new topological links at junctions and bifurcations remains to be proven in practice. Similarly, the gullies (small channels) on hill faces connecting to the main channel, a feature common to any mountainous terrain, is not included. These gullies, called the inner leafs of the channel network in the interlocking ridge and channel network model by Werner (1988), are a prominent terrain feature and relevant in hydrological modelling for catchment analyses. Again, the problem here is that these gullies start from a point on the hill face (called *source nodes*; Werner 1988), which is not a critical point.

2. Scaling

Surface features are organised in a hierarchy, expressed as a variation in their spatial extents. For example, in the case of terrain surface, a gully on a slope face has small spatial extent compared to the channel it drains into. The position of the feature in hierarchical arrangement can be regarded as the scale of the feature. Wood (1999) demonstrated that a location on terrain could be a part of the features of different scale and feature types. Griffin and Colchester (1995) presented an example of such hierarchy in the multi-scaled nature of medical image surfaces. Therefore, all surfaces are inherently composed of multi-scaled features. It therefore implies that a surface would have multiple surface networks representing the feature scales of a terrain. To my knowledge, the existing surface network model (or for that matter Morse Theory)

[1] Morse (1925) and Pfaltz (1976) suggest that points in inequilibrium, for example, degenerate points could be decomposed into non-degenerate points but it has not been demonstrated in practice widely.

does not address how such individual surface networks could be unified into a single surface network model of a terrain.

3. Uncertainty

While there is no denying the fact that the abstraction of a surface as a surface network could significantly reduce the disk space requirements of a terrain, it still remains an approximation of surface based entirely on a minimal set of points and lines. As mentioned earlier, surface network would inevitably fail to capture all the variations present in the surface, which could lead to a considerable amount of uncertainty in the surface. In general, the uncertainty will depend upon the deviation of the surface from an ideal Morse function and would vary spatially across the surface. At present, there are no proposals for ascertaining the uncertainty associated with a surface network.

13.2 LIMITATIONS OF THE METHOD FOR GENERATING SURFACE NETWORKS

As can be seen in Chapters 3 to 6, there are various ways for generating surface networks, however, none of them appear to be consistent. The only topologically consistent surface network known to me was created by Gert Wolf because he derived it manually by digitising critical points and lines from a map. I think other methods for the automated generation of surface are not always successful due to the following reasons:

1. Scale dependency

As mentioned in the previous section, surfaces have a multi-scaled arrangement of critical points and lines. But many feature extraction methods are scale-dependent because they only explore a fixed area around a point to classify the point into a feature type. Therefore, critical points and lines, features that could fit within the search space, are extracted resulting into a scale-dependent extraction of surface networks. The surface network generation methods described in Chapters 3, 4, and 6 suffer from scale dependency in different ways.

- *Scale dependency of TIN-based surface network generation method (Chapters 3 and 6)* Triangulation-based surface network generation methods use only the adjacent neighbours of a TIN vertex for the classification of the critical points and thus have a fixed scale of observation. In a later work, Takahashi, et al., (1995) suggested referring to the scale-space theory (Witkin, 1983, Lindeberg, 1994) prior to the extraction of surface network. However, it is uncertain how the current method of triangulation can be "extended" to detect larger features.
- *Scale dependency of raster surfaces based surface network generation (Chapter 4)* Although the method by Schnedier and Wood (Chapter 4) allows a multi-scale extraction of the surface network, but since only the cell at the centre of the filter windows could be classified, the number of cells that could be classified reduces as the filter window grows in size.

2. Delineation of topological links

- *Broken surface networks* In all the automated methods for the generation of surface networks, the surface network is built incrementally by tracing the ridges and channels from the passes. It is assumed that tracing the steepest (shallowest) gradient or the ridge (channel) axes starting from the pass will ultimately lead to either a peak(pit) or to the edge of the surface (external pit or peak). However, as discussed above, real surfaces are seldom sufficiently smooth enough for a successful delineation of the ridges (channels). As a result, ridges and channels do not necessarily terminate at peaks and pits respectively.
- *Junctions, bifurcations are not extracted* None of the surface network extraction methods extract junctions and bifurcations in the way suggested by Wolf (Chapter 2).

13.3 LIMITATIONS OF THE METHODS FOR GENERALISATION OF SURFACE NETWORK

Despite the simplicity and robustness of the method for the surface network contraction proposed by Pfaltz (1976) and Wolf (1989), it has three main limitations that restrict its use as a practical terrain generalisation method:

1. Limitations of weight measures

According to Weibel and Dutton (1999) the first step in the generalisation of spatial datasets is the cartometric evaluation of the dataset, which involves an assessment of the dataset in order to select the portions suitable for generalisation. For example, in the case of surface network, cartometric evaluation involves the assignment of weights (based on elevation differences) and using it to rank (by using the selection criteria) the peaks and pits for contraction. Mark (1977) and Wolf (1988) have argued that all weights and selection criteria must be based on elevation. However, it is simple to prove that elevation and elevation differences alone will provide little information about the importance of a point. For example, two peaks could have ridges with equal elevation differences but of different extent. Pfaltz (1976, p. 92) first raised the potential arbitrariness in assigning weights and selecting points for contraction. Most crucially, the existing weight measures could be extended to take into account the area-based morphometric measures and weights such as slope, and so on. In addition, it is unclear how critical points with equal weights should be ordered for contraction, as a method based on lexico-graphical ordering (Chapter 3) produces quite contrasting results depending upon the order (Rana, 2000a,b).

2. Sequential generalisation of surface network graph

Pfaltz (1976), Mark (1977) and Wolf (1988) proposed an iterative and sequential (based on rank) generalisation of surface network until a surface with a desired simplicity has been reached. However, it is sometimes desirable to influence the generalisation sequence for the sake of structural integrity (Pfaltz 1976, p. 92; Wolf 1988) or when the sequence could be anomalous (e.g. two points with equal weights). Currently, there are no proposals to achieve an arbitrary generalisation sequence.

3. Purely topological nature of generalisation

Existing generalisation method of surface network is only able to achieve surface simplification at a topological level, that is, while there is a surface corresponding to the original surface network, simplifications in surface networks do not have a morphologic expression. For example, if after a generalisation three new ridge edges are created there are no corresponding ridges produced in the surface. This is perhaps the most critical limitation of existing generalisation methods and prevents it from being used as a practical terrain generalisation method. Wolf (1988) did not consider the construction of the generalised morphology based on changes in the topological links and merely triangulated the critical points left after generalisation. Pascucci and others (see Chapter 8) have proposed an approach for regenerating a terrain surface after contraction; however, they do not provide any justification in terms of the validity of the resultant terrain, such as done in terrain evolution modelling (Burrough, 1998).

13.4 FUTURE DIRECTIONS

In addition to tackling the problems raised above, I believe that a huge potential of the surface network, Reeb graphs, and contour trees exist in applications related to visualisation of multi-dimensional datasets and spatial analyses (e.g. spatial queries). Although I think the book contains majority of the prominent works in this topic, there were many others research works that could have increased the strength of the book. Some interesting works on surface networks missing from the book include Carroll Johnson's work on Crystallographic Topology (URL #7), James Helman and Lamertus Hesselink's work on vector flow topology visualisation, and Warrick Dawes work on modelling hydrology under constraints of surface topology (Dawes and Short, 1994), just to mention a few. In this book, most examples were based on two-dimensional surfaces, however, like John Pfaltz (see Foreword) I also believe that the real benefit of surface networks is in the representation and storage of multi-dimensional surfaces. Another interesting field, which I hope to dwell on in the future, is the application of surface networks to model complex surfaces associated with dynamical systems, although I am sure it would be non-trivial to extend the surface network model to non-Morse functions. It will be an interesting challenge to come up with a set of inequalities for dynamical systems, like the one given by Morse Theory for smooth functions, and develop a simple model to understand the behaviour of the complex surfaces using those inequalities.

References

Acevedo, W., and Masuoka, P., 1997. Time-series animation techniques for visualizing urban growth, *Computers & Geosciences*, **23**, 423–435.

Ahuja, R.K., Magnanti, T.L., and Orlin, J.B., 1993. *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ.

Anoshkina, E.V., Belyaev, A.G., Okunev, O.G., and Kunii, T.L., 1994a. Ridges and ravines: a singularity approach, *International Journal of Shape Modeling*, **1**, 1–11.

Anoshkina, E.V., Belyaev, A.G., and Kunii, T.L., 1994b. Ridges, ravines and caustic singularities, *International Journal of Shape Modeling*, **1**, 13–22.

Apostol, T.M., 1969. *Calculus Volume II*, John Wiley & Sons, New York.

Artzy, E., Frieder, G., and Herman, G.T., 1981. The theory, design, implementation, and evaluation of three-dimensional surface detection algorithms, *Computer Graphics and Image Processing*, **15**, 1–24.

Attene, M., Biasotti, S., and Spagnuolo, M., 2003. Shape understanding by contour-driven retiling, *The Visual Computer*, **19**, 127–138.

Bader, R., Nguyen-Dang, T.T., and Tal, Y., 1979. Quantum topology of molecular charge distributions II. Molecular structure and its charge, *Journal of Chemical Physics*, **70**, 4316–4329.

Bajaj, C.L., Pascucci, V., and Schikore, D.R., 1996. Fast isocontouring for improved interactivity, In *Proceedings of the IEEE Symposium on Volume Visualization*, San Francisco, CA, 39–46.

Bajaj, C.L., Pascucci, V., and Schikore, D.R., 1997. The contour spectrum, In *Proceedings of IEEE Visualization 1997*, Phoenix, AZ, 167–173.

Bajaj, C.L., Pascucci, V., and Schikore, D.R., 1998. Visualization of Scalar Topology for Structural Enhancement, In *Proceedings of the IEEE Conference on Visualization*, Research Triangle Park, NC, 51–58.

Bajaj, C.L., Pascucci, V., and Schikore, D.R., 1999. Accelerated isocontouring of scalar fields, In: C. Bajaj (Ed.), *Data Visualization Techniques*: *Volume 6 Trends in Software*, John Wiley & Sons, Chichester.

Bajaj, C.L., and Schikore, D.R., 1996. *Visualization of Scalar Topology for Structural Enhancement*, Technical Report CSD-TR-96-006, Department of Computer Sciences, Purdue University.

Bajaj, C.L., and Schikore, D.R., 1998. Topology preserving data simplification with error bounds, *Computers & Graphics*, **22**, 3–12.

Banchoff, T.F., 1970. Critical points and curvature for embedded polyhedral surfaces, *American Mathematical Monthly*, **77**, 475–485.

Barré, F., and Lopez, J., 2000. Watershed lines and catchment basins: a new 3D-motif method, *International Journal of Machine Tools and Manufacture*, **40**, 1171–1184.

Bassett, K.A., 1972. Numerical methods for map analysis, In: C. Board, R.J. Chorley, P. Haggett, and D.R. Stoddart (Eds.), *Progress in Geography*, 4, Arnold, London.

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana

ISBN: 0-470-85151-1

Bergman, L., Rogowitz, B., and Treinish, L., 1995. A rule-based tool for assisting colormap selection, In *Proceedings of the IEEE Conference on Visualization '95*, Atlanta, GA, 118–125.

Berry, B.J.L., Simmons, J.W., and Tennant, R.J., 1963. Urban populations: structure and change, *Geographical Review*, **53**, 389–405.

Bertin, J., 1967. *Semiologie Graphique*, Mouton, Paris.

Biasotti, S., Falcidieno, B., and Spagnuolo, M, 2000. Extended Reeb graphs for surface understanding and description, In *Proceedings of the 9th Discrete Geometry for Computer Imagery Conference*, Lecture Notes in Computer Science, Springer-Verlag, Uppsala.

Biasotti, S., Falcidieno, B., and Spagnuolo, M., 2002. Shape abstraction using computational topology techniques, In: U. Cugini, and M. Wozny (Eds.), *From Geometric Modeling to Shape Modeling*, Kluwer Academic Publishers, London.

Bleau, A., and Leon, L.J., 2000. Watershed-based segmentation and region merging, *Computer Vision and Image Understanding*, **77**, 317–370.

Blunt, L., and Ebdon, S., 1996. The application of three-dimensional surface measurement techniques to characterizing grinding wheel topography, *International Journal of Machine Tools and Manufacture*, **36**, 1207–1226.

Bondy, J.A., and Murty, U.S.R., 1978. *Graph Theory with Applications*, The Macmillan Press, London, Basingstoke.

Bracken, I., 1994. Towards improved visualization of socio-economic data, In: H.M. Hearnshaw, and D.J. Unwin (Eds.), *Visualization in Geographical Information Systems*, John Wiley & Sons, Chichester.

Bremer, P.-T., Edelsbrunner, H., Hamann, B., and Pascucci, V., 2003, A multi-resolution data structure for two-dimensional Morse-Smale functions, In: G. Turk, J.J. van Wijk, and R.J. Moorhead (Eds.), *IEEE Visualization 2003*, Seattle, WA, 139–146.

Brown, J.L., 1991. Vertex based data dependent triangulations, *Computer Aided Geometric Design*, **8**, 239–251.

Brush, J.E., 1968. Spatial patterns of population in Indian cities, *Geographical Review*, **58**, 362–391.

Burrough, P.A., and McDonnell, R.A., 1998. *Principles of Geographical Information Systems*, Oxford University Press, Oxford.

Carr, H., Snoeyink, J., and Axen, U., 2003. Computing contour trees in all dimensions, *Computational Geometry: Theory and Applications*, **24**, 75–94.

Casetti, E., 1967. Urban population density patterns: an alternate explanation, *Canadian Geographer*, **11**, 96–100.

Casetti, E., 1969. Alternate urban population density models: an analytical comparison of their validity range, In: A.J. Scott (Ed.), *Studies in Regional Science*, Pion, London.

Cayley, A., 1859. On contour and slope lines. *The London, Edinburgh and Dublin. Philosophical Magazine and Journal of Science*, **XVIII**, 264–268.

Cignoni, P., Montani, C., Puppo, E., and Scopigno, R., 1996. Optimal isosurface extraction from irregular volume data, In *Proceedings of the IEEE Symposium on Volume Visualization*, 31–38.

Clark, C., 1951. Urban population density, *Journal of Royal Statistical Society, Section A*, **114**, 490–496.

Clark, C., 1958. Urban population densities, *Bulletin de l'Institute International de Statistique*, **36**, 60–68.

Clark, C.G., 1967. *Population Growth and Land Use*, St. Martin's Press, New York.

Collard, K., and Hall, G., 1977. Orthogonal trajectories of the electron density, *International Journal of Quantum Chemistry*, **12**, 623–637.

Cormen, T.H., Leiserson, C.E., and Rivest, R.L., 1990. *Introduction to Algorithms*, MIT Press, Cambridge, MA.

Crampton, J.W., 2002. Interactivity types in geographic visualization, *Cartography and Geographic Information Science*, **29**, 85–98.

Danskin, J., and Hanrahan, P., 1992. Fast algorithms for volume ray tracing. In *Proceedings of the 1992 Workshop on Volume Visualization*, Boston, MA, 91–98.

Dawes, W.R., and Short, D.L., 1994. The significance of topology for modelling the surface hydrology of fluvial landscapes, *Water Resources Research*, **30**, 1045–1055.

de Berg, M., and van Kreveld, M., 1997. Trekking in the Alps without freezing or getting tired, *Algorithmica*, **18**, 306–323.

De Floriani, L., and Magillo, P., 1994. Visibility algorithms on triangulated terrain models, *International Journal of Geographical Information Systems*, **8**, 13–41.

De Floriani, L., Marzano, P.K., and Puppo, E., 1994. Line-of-sight communication on terrain models, *International Journal of Geographical Information Systems*, **8**, 329–342.

De Floriani, L., Mesmoudi, M.M., and Danovaro, E., 2002. Extraction of critical nets based on a discrete gradient vector field, In: I. Navazo, and P. Slusallek (Eds.), *Proceedings of Eurographics 2002*, Blackwell Publishers, Saarbrucken.

De Floriani, L., and Puppo, E., 1992. A hierarchical triangle-based model for terrain description, In: A.U. Frank, I. Campari, and U. Formentini (Eds.), *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Lecture Notes in Computer Science, Springer-Verlag, Heidelberg.

De Saint-Venant, 1852. Surfaces á plus grande pente constituées sur des lignes courbes, Bulletin de la sc.philomath. de Paris.

Dey, T.K., Edelsbrunner, H., and Guha, S., 1999, Computational topology, In: B. Chazelle, J.E. Goodman, and R. Pollack (Eds.), *Advances in Discrete and Computational Geometry*, Contemporary Mathematics 223, AMS, Providence, RI.

DiBiase, D., MacEachren, A.M., Krygier, J.B., and Reeves, C., 1992. Animation and the role of map design in scientific visualization, *Cartography and Geographic Information Systems*, **19**, 201–204.

Dransch, D., 2000. The use of different media in visualizing spatial data, *Computers & Geosciences*, **26**, 5–9.

Duncan, B., Sabagh, G., and van Ansdol Jr., M.D., 1961–1962. Patterns of city growth, *American Journal of Sociology*, **67**, 418–429.

Dunne, A., 1999. *Hertzian Tales: Electronic Products, Aesthetic Experience and Critical Design*, RCA Computer Related Design Research, Royal College of Art, London.

Dyn, N., Levin, D., and Rippa, S., 1990. Data dependent triangulation for piecewise linear interpolation, *IMA Journal of Numerical Analysis*, **10**, 137–154.

Earnshaw, R.A., and Watson, D., 1993. Animation and scientific visualization, *Tools and Applications*, Academic Press, London.

Edelsbrunner, H., Harer, J., Natarajan, V., and Pascucci, V., 2003a. Morse complexes for piecewise linear 3-manifolds, In *Proceedings of the 19th ACM Symposium on Computational Geometry*, San Diego, CA, 361–370.

Edelsbrunner, H., Harer, J., and Zomorodian, A., 2003b. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds, *Discrete & Computational Geometry*, **30**, 87–107.

Edelsbrunner, H., Letscher, D., and Zomorodian, A., 2002. Topological persistence and simplification, *Discrete & Computational Geometry*, **28**, 511–533.

Edelsbrunner, H., and Mücke, E.P., 1990. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms, *ACM Transactions on Graphics*, **9**, 66–104.

Emmer, N.N.M., 2001. Determining the effectiveness of animations to represent geo-spatial temporal data: a first approach, In *Proceedings of the 4th Association of Geographic Information Laboratories in Europe Conference on Geographic Information Science*, Brno, 585–589.

Engel, K., Kraus, M., and Ertl, T., 2001. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading, In *Proceedings of Graphics Hardware 2001*, Los Angeles, CA, 9–16.

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping, *Zeitschrift für Geomorphologie*, **36**, 274–295.

Feuchtwanger, M., and Poiker, T.K., 1987. The surface patchwork – an intelligent approach to terrain modelling, In *Proceedings of the 5th Annual North West Conference on Surveying and Mapping*, Whistler.

Fisher, P.F., 1991. First experiments in viewshed uncertainty – The accuracy of the viewshed area, *Photogrammetric Engineering & Remote Sensing*, **57**, 1321–1327.

Fisher, P.F., 1992. First experiments in viewshed uncertainty: simulating the fuzzy viewshed, *Photogrammetric Engineering & Remote Sensing*, **58**, 345–352.

Fisher, P.F., 1993. Algorithm and implementation uncertainty in viewshed analysis, *International Journal of Geographical Information Systems*, **7**, 331–347.
Florinsky, I.V., 2002. Errors of signal processing in digital terrain modelling, *International Journal of Geographical Information Science*, **16**, 475–501.
Fomenko, A.T., and Kunii, T.L., 1997. *Topological Modeling for Visualization*, Springer-Verlag, Heidelberg, New York.
Fowler, R.J., and Little, J.J., 1979, Automatic extraction of irregular network digital terrain models, *Computer Graphics*, **13**, 199–207.
Franklin, W.M., 2000. Approximating visibility, In *Proceedings of the 1st International Conference on Geographic Information Science*, Savannah, GA, 126–138.
Franklin, W.M., Ray, C.K., and Mehta, S., 1994. *Geometric Algorithms for Siting of Air Defense Missile Batteries*, Technical Report on Contract No. DAAL03-86-D-0001, Battelle, Columbus Division, Columbus, OH, 116.
Frank, A., Palmer, B., and Robinson, V., 1986. Formal methods of the accurate definition of some fundamental terms in physical geography, In *Proceedings of the 2nd International Symposium on Spatial Data Handling*, Seattle, WA, 583–599.
Freeman, H., and Morse, S.P., 1967. On searching a contour map for a given terrain profile, *Journal of the Franklin Institute*, **284**, 1–25.
Fujii, A., 1978. Study of activity contour part I: report on the structural concept "ridge" of closed curve, *Transactions of the Architectural Institute of Japan*, **267**, 121–128 (in Japanese).
Gahegan, M., 1999. Four barriers to the development of effective exploratory visualization for the geosciences, *International Journal of Geographical Information Science*, **13**, 289–309.
Gatrell, A., 1994. Density estimation and the visualization of point systems, In: H.M. Hearnshaw, and D.J. Unwin (Eds.), *Visualization in Geographical Information Systems*, John Wiley & Sons, Chichester.
Gershon, N.D., 1992. Visualization of fuzzy data using generalized animation, In *Proceedings of the IEEE Visualization '92*, Boston, MA, 268–273.
Gerstner, T., and Pajarola, R., 2000. Topology preserving and controlled topology simplifying multiresolution isosurface extraction, In *Proceedings of the IEEE Visualization 2000*, Salt Lake City, UT, 259–266.
Gesch, D., Oimoen, M., Greenlee, S., Nelson, C., Steuck, M., and Tyler, D., 2002. The National elevation data set, *Photogrammetric Engineering and Remote Sensing*, **68**, 5–32.
Globus, A., Levit, C., and Lasinski, T., 1991. A tool for visualizing the topology of three-dimensional vector fields, In *Proceedings of the IEEE Conference on Visualization '91*, San Diego, CA, 33–40.
Gold, C., and Cormack, S., 1986. Spatially ordered networks and topographic reconstructions, In *Proceedings of the 2nd International Symposium on Spatial Data Handling*, Seattle, WA, 74–85.
Gomes, J., Costa, B., Darsa, L., and Velho, L., 1998. *Warping and Morphing of Graphical Objects*, Morgan Kaufmann, San Francisco, CA.
Goodchild, M.F., 1990. Spatial information science, In *Proceedings of the 4th International Symposium on Spatial Data Handling*, Zurich, 3–12.
Goresky, M., and MacPherson, R., 1988. *Stratified Morse Theory*, Springer-Verlag, New York.
Griffin, L.D., and Colchester, A.C.F., 1995. Superficial and deep structure in linear diffusion scale space: critical points, isophotes and separatrices, *Image and Vision Computing*, **13**, 543–557.
Griffiths, H.B., 1976. *Surfaces*, Cambridge University Press, New York.
Griffiths, H.B., 1981. *Surfaces*, 2nd Edition, Cambridge University Press, Cambridge, MA.
Guillemin, V., and Pollack, A., 1974. *Differential Topology*, Prentice Hall, Englewood Cliffs, NJ.
Haggett, P., and Bassett, K.A., 1970. The use of trend-surface parameters in inter-urban comparisons, *Environment and Planning*, **2**, 225–237.
Harary, F., 1971. *Graph Theory*, 2nd Edition, Addison-Wesley, Reading, MA.
Helman, J.L., and Hesselink, L., 1991. Visualizing vector field topology in fluid flows, *IEEE Computer Graphics & Applications*, **11**, 36–46.
He, T., Hong, L., Varshney, A., and Wang, S.W. 1996. Controlled topology simplification, *IEEE Transactions on Visualization and Computer Graphics*, **2**, 171–184.

Hilaga, M., Shinagawa, Y., Komura, T., and Kunii, T.L., 2001. Topology matching for fully automatic similarity estimation of 3D shapes, ACM Computer Graphics, In *Proceedings of SIGGRAPH 2001*, Los Angeles, CA, 203–212.

Hirsch, M.W., 1976. *Differential Topology*, Springer-Verlag, New York.

Howie, C.T., and Blake, E.H., 1994. The mesh propagation algorithm for isosurface construction, *Computer Graphics Forum*, **13**, 65–74.

Ikeda, T., Kunii, T.L., Shinagawa, Y., and Ueda, M., 1992. A geographical database system based on the homotopy model, In: T.L. Kunii, and Y. Shinagawa (Eds.), *Modern Geometric Computing for Visualization*, Springer-Verlag, Tokyo.

ISO 4287, 1997. Geometrical Product Specifications (GPS) – Surface texture: Profile method – Terms, definitions and surface texture parameters.

Itoh, T., and Koyamada, K., 1995. Automatic isosurface propagation using an extrema graph and sorted boundary cell lists, *IEEE Transactions on Visualization and Computer Graphics*, **1**, 319–327.

Izraelevitz, D., 2003. A fast algorithm for approximate viewshed computation, *Photogrammetric Engineering and Remote Sensing*, **69**, 767–774.

Jenson, S.K., and Domingue, J.O., 1988. Extracting topographic structure from digital elevation data for geographic information systems analysis, *Photogrammetric Engineering and Remote Sensing*, **54**, 1593–1600.

Johnson, C.K., 1977. Peaks, passes, pales, and pits: a tour through the critical points of interest in a density map, In *Abstracts of the American Crystallographic Association Meeting*, Pacific Grove, CA, 30.

Johnson, C.K., Burnett, M.N., and Dunbar, W.D., 1999. Crystallographic topology and its applications, In: P. Bourne, and K. Watenpaugh (Eds.), *Crystallographics Computing 7: Proceedings from the Macromolecular Crystallography Computing School*, Oxford University Press, Oxford.

Johnson, G.G., and Vance, V., 1967. Application of a Fourier data-smoothing technique to the meteoric crater, Ries Kassell, *Journal of Geographical Research*, **72**, 1741–1750.

Jones, C.B., 1991. Database architecture for multi-scale GIS, In *Proceedings of AUTOCARTO 10*, Baltimore, MD, 1–14.

Jun, C., Kim, D., Kim, D., Lee, H., Hwang, J., and Chang, T., 2001. Surface slicing algorithm based on topology transition, *Computer Aided Design*, **33**, 825–838.

Kerlick, G.D., 1990. Moving iconic objects in scientific visualization, In *Proceedings of the IEEE Conference on Visualization '90*, Phoenix, AZ, 124–129.

Kettner, L., and Snoeyink, J., 2001. A prototype system for visualizing time-dependent volume data, In *Proceedings of the 17th Annual ACM Symposium on Computational Geometry*, Medford, MA, 327–328.

Kim, Y.-H., Rana, S., and Wise, S., 2002. Exploring multiple viewshed analysis using terrain features and optimisation techniques, In *Proceedings of GISRUK 2002*, University of Sheffield, Sheffield.

Kim, Y.-H., Wise, S., and Rana, S., 2003. A solution approach for multiple viewshed location problems, In *Proceedings of the RGS-IBG Annual Conference*, London.

Kindle, J.H., 1950. *Theory and Problems of Plane and Solid Analytical Geometry*, McGraw–Hill, New York.

King, L.J., 1969. *Statistical Analysis in Geography*, Prentice Hall, Englewood Cliffs, NJ.

Koenderink, J.J., and van Doorn, A.J., 1979. The structure of two dimensional scalar fields with applications to vision. *Biological Cybernetics*, **33**, 151–158.

Koenderink, J.J., and van Doorn, A.J., 1998. The structure of relief, *Advances in Imaging and Electron Physics*, **103**, 66–150.

Koussoulakou, A., 1990. *Computer-Assisted Cartography for Monitoring Spatiotemporal Aspects of Urban Air Pollution*, Ph.D. thesis, Delft Press, University of Delft, Delft.

Kramer, C., 1958. Population density patterns, *CATS Research News*, **2**, 3–10.

Kraus, M. and Ertl, T., 2001. Topology-Guided Downsampling, In K. Mueller, and A. Kaufman (Eds.), *Volume Graphics 2001*, Springer-Verlag, New York.

Kraus, M., and Ertl, T., 2002. Adaptive texture maps, In *Proceedings of Graphics Hardware 2002*, Saarbrucken, 7–15.

Krumbein, W.C., 1956. Regional and local components in facies maps, *Bulletin of the American Association for Petroleum Geology*, **40**, 2163–2194.

Krygier, J.B., Reeves, C., DiBiase, D.W., and Cupp, J., 1997. Design, implementation and evaluation of multimedia resources for geography and earth science education, *Journal of Geography in Higher Education*, **21**, 17–39.

Kweon, I.S., 1991. *Modeling Rugged Terrain by Mobile Robots with Multiple Sensors*, Ph.D. thesis, Department of Robotics, Carnegie Mellon University, Pittsburg, PA, 131.

Kweon, I.S., and Kanade, T., 1991. Extracting topographic features for outdoor mobile robots, In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, 1992–1997.

Kweon, I.S., and Kanade, T., 1994. Extracting topographic terrain features from elevation maps, *Computer Vision, Graphics, and Image Understanding*, **59**, 171–182.

Latham, R.F., and Yeats, M.H., 1970. Population density growth in metropolitan Toronto, *Geographical Analysis*, **2**, 177–185.

Laur, D., and Hanrahan, P., 1991. Hierarchical splatting: a progressive refinement algorithm for volume rendering, *ACM SIGGRAPH Computer Graphics*, **25**, 285–288.

Lazarus, F., and Verroust, A., 1999. Level set diagrams of polyhedral objects, In *Proceedings of the 5th ACM Symposium on Solid Modeling and Applications*, Ann Arbor, MI, 130–140.

Lee, J., 1991. Analyses of visibility sites on topographic surfaces, *International Journal of Geographical Information Systems*, **5**, 413–429.

Lee, J., 1992. Visibility dominance and topographical features on digital elevation models, In *Proceedings of the 5th International Symposium on Spatial Data Handling*, Charleston, SC, 622–631.

Levoy, M., and Whitaker, R., 1990. Gaze-directed volume rendering, *ACM SIGGRAPH Computer Graphics*, **24**, 217–223.

Levy, M.A., Pollack, H.N., Pomeroy, and P.W., 1970. Motion picture of the seismicity of the earth, 1961–1971, *Bulletin of the Seismological Society of America*, **60**, 1015–1016.

Lindeberg, T., 1994. *Scale-space Theory in Computer Vision*, Kluwer Academic Press, Dordrecht.

Livnat, Y., Shen, H.-W., and Johnson, C.R., 1996. A near optimal isosurface extraction algorithm using the span space, *IEEE Transactions on Visualization and Computer Graphics*, **2**, 73–84.

Lopes, A., and Brodlie, K., 2003. Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing, *IEEE Transactions on Visualization and Computer Graphics*, **9**, 16–29.

Lorensen, W.E., and Cline, H.E., 1987. Marching cubes: a high resolution 3D surface construction algorithm, *ACM SIGGRAPH Conference*, **21**, 163–169.

MacEachren, A.M., 1995a. *How Maps Work: Issues in Representations and Design*, Guildford Press, New York.

MacEachren, A.M., 1995b. *Some Truth with Maps: A Primer on Design and Symbolization*, Association of American Geographers, Washington, DC.

Mark, D.M., 1977. *Topological Randomness of Geomorphic Surfaces*, Technical Report No. 15, Geographic Data Structures Project, ONR Contract N00014-75-C-0886, 138.

Maxwell, J.C., 1870. On hills and dales, *The London, Edinburgh, and Dublin, Philosophical Magazine and Journal of Science, Series 4*, **40**, 421–427.

McCloud, S., 1993. *Understanding Comics*, Kitchensink Press, Northampton, USA.

Meissner, M., Guthe, S., and Strasser, W., 2002. Interactive lighting models and pre-integration for volume rendering on PC graphics accelerators, In *Proceedings of Graphics Interface 2002*, Calgary, 209–218.

Miller, G.A., 1956. The magical number seven plus, or minus two: some limits on our capacity for processing information, *The Psychological Review*, **63**, 81–97.

Mills, E.S., 1972. *Studies in the Structure of the Urban Economy*, The Johns Hopkins University Press, Baltimore, MD.

Milnor, J.W., 1963. Morse theory, *Annals of Mathematics Studies*, **51**, 153.

Morrison, J.B., Betrancourt, M., and Tverksy, B., 2000. Animation: does it facilitate learning? In *American Association of Artificial Intelligence Spring Symposium Smart Graphics 2000*, Stanford, CA, 53–60.

Morse, M., 1925. Relations between the critical points of a real function on n independent variables, *Transactions of the American Mathematical Society*, **27**, 345–396.

Morse, M., and Cairns, S.S., 1969. *Critical Point Theory in Global Analysis and Differential Topology*, Academic Press, New York, London.

Morse, S.P., 1968. A mathematical model for the analysis of contour-line data, *Journal of the Association for Computing Machinery*, **15**, 205–220.

Morse, S.P., 1969. Concepts of use in contour map processing, *Communications of the ACM*, **12**, 147–152.

Muth, R.F., 1965. Spatial structure of the housing market, papers, *Regional Science Association*, **15**, 173–183.

Muth, R.F., 1969. *Cities and Housing*, University of Chicago Press, Chicago, IL.

Nackman, L.R., 1982. *Three-dimensional Shape Description Using the Symmetric Axis Transform*, Ph.D. thesis, Department of Computer Science, University of North Carolina, Chapel Hill, NC, 182.

Nackman, L.R., 1984. Two-dimensional critical point configuration graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 442–450.

Newling, B.E., 1969. The spatial variation of urban population densities, *Geographical Review*, **59**, 242–252.

Norcliffe, G.B., 1969. On the use and limitation of trend surface models, *Canadian Geographer*, **13**, 338–348.

O'Sullivan, D., and Turner, A., 2001. Visibility graphs and landscape visibility analysis, *International Journal of Geographical Information Systems*, **15**, 221–237.

Ogao, P.J., and Blok, C.A., 2001. Cognitive aspects on the representation of dynamic environmental phenomena using animations, In: C. Rautenstrauch, and S. Patig (Eds.), *Environmental Information Systems in Industry and Public Administration*, Idea Group Publishers, Harrisburg, PA.

Ogao, P.J., and Kraak, M.-J., 2001. Geospatial data exploration using interactive and intelligent cartographic animations, In *Proceedings of the International Cartographic Conference*, Beijing, 2649–2657.

Ohlberger, M., and Rumpf, M., 1997. Hierarchical and adaptive visualization on nested grids, *Computing*, **59**, 365–385.

Ohtomo, A., 1979. *Analysis of Japanese Urban Population Distributions*, Taimeido, Tokyo (in Japanese).

Okabe, A., and Masuda, S., 1984. Qualitative analysis of two-dimensional urban population distributions in Japan, *Geographical Analysis*, **16**, 301–312.

Okabe, A., and Sadahiro, Y., 1994. A statistical method for analyzing the spatial relationship between the distribution of activity points and the distribution of activity continuously distributed over a region, *Geographical Analysis*, **26**, 152–167.

Openshaw, S., Waugh, D., and Cross, A., 1994. Some ideas about the use of map animation as a spatial analysis tool, In: H.M. Hearnshaw, and D.J. Unwin (Eds.), *Visualization in Geographical Information Systems*, John Wiley & Sons, Chichester.

Paddenburg, A.V., and Wachowicz, M., 2001. The effect of generalisation on filtering noise for spatio-temporal analyses, In *Proceedings of the 6th International Conference on GeoComputation*, Brisbane (On CD-ROM).

Palmer, B., 1984. Symbolic feature analysis and expert systems. In *Proceedings of the 1st International Symposium on Spatial Data Handling*, Zurich, 465–478.

Pascucci, V., and Cole-McLaughlin, K., 2002. Efficient computation of the topology of level sets, In *Proceedings of IEEE Visualization 2002*, Piscataway, NJ, 187–194.

Peterson, M.P., 1993. Interactive cartographic animation, *Cartography and Geographical Information Systems*, **20**, 40–44.

Peucker, T.K., 1973. *Geographic Data Structures. Progress Report After Year One*, Technical Report No. 1, Geographic Data Structures Project, ONR Contract N00014-73-C-0109, 44.

Peucker, T.K., and Douglas, D.D., 1975. Detection of surface-specific points by local parallel processing of discrete terrain elevation data, *Computer Graphics and Image Processing*, **4**, 375–387.

Peucker, T.K., Fowler, R.J., Little, J.J., and Mark, D.M., 1976. *Triangulated Irregular Networks for Representing Three-Dimensional Surfaces*, Technical Report No. 10, Geographic Data Structures Project, ONR Contract N00014-75-C-0886, 70.

Peucker, T.K., Fowler, R.J., Little, J.J., and Mark, D.M., 1978. The triangulated irregular network, In *Proceedings of the Symposium on Digital Terrain Models*, St. Louis, MO, 516–540.

Pfaltz, J.L., 1976. Surface networks, *Geographical Analysis*, **8**, 77–93.

Pfaltz, J.L., 1978. *Surface Networks, an Analytic Tool for the Study of Functional Surfaces*, Final Report on NSF Grant DCR-74-13353, 99.

Porteous, L.R., 1994. *Geometric Differentiation for the Intelligence of Curves and Surfaces*, Cambridge University Press, Cambridge, MA.

Rana, S., 2000a. Experiments on the generalisation and visualisation of surface networks, In *Proceedings of GISRUK 2000*, University of York, York.

Rana, S., 2000b. *Experiments on the Generalisation and Visualisation of Surface Networks*, Centre for Advanced Spatial Analysis, University College London, Working Paper Series No. 24, 19.

Rana, S., 2003a. Fast approximation of visibility dominance using topographic features as targets and the associated uncertainty, *Photogrammetric Engineering and Remote Sensing*, **69**, 881–888.

Rana, S., 2003b. Visibility analysis, *Environment and Planning-B*, **30**, 641–642.

Rana, S., and Dykes, J., 2003. A framework for augmenting the visualisation of dynamic raster surfaces, *Information Visualization*, **2**, 126–139.

Rana, S., and Morley, J., 2002. *Surface Networks*, Centre for Advanced Spatial Analysis, University College London, Working Paper Series No. 43, 72.

Rana, S., and Wood, J., 2000. *Weighted and Metric Surface Networks – New Insights and an Interactive Application for Their Generalisation in Tcl/Tk*, Centre for Advanced Spatial Analysis, University College London, Working Paper Series No. 25, 26.

Raper, J., 2000. *Multidimensional Geographic Information Systems*, Taylor & Francis, London.

Reeb, G., 1946. Sur les points singuliers d'une forme de Pfaff completement integrable ou d'une fonction numerique (On the singular points of a completely integrable Pfaff form or of a numerical function), *Computes Rendus de l'Académic des Sciences*, Paris, **222**, 847–849.

Reech, M., 1858. Proprieté générale des surfaces fermées, Ecole, *Journal de l'Ecole Polytechnique*, **37**, 169–178.

Rezk-Salama, C., Engel, K., Bauer, M., Greiner, G., and Ertl, T., 2000. Interactive volume rendering on standard PC graphics hardware using multi-textures and multi-stage rasterization, In *Proceedings of Graphics Hardware 2000*, Interlaken, 109–118.

Rhyne, T.M., 1997. Going virtual with geographic information and scientific visualization, *Computers & Geosciences*, **23**, 489–491.

Rosenfeld, A., and Kak, A., 1982. *Digital Picture Processing*, Academic Press, San Diego, CA.

Roubal, J., and Poiker, T.K., 1985. Automated contour labelling and the contour tree, In *Proceedings of AUTOCARTO 7*, Washington, DC, 472–481.

Sadahiro, Y., 2001. Analysis of surface changes using primitive events, *International Journal of Geographical Information Science*, **15**, 523–538.

Salomon, D., 1998. *Data Compression: The Complete Reference*, Springer, New York.

Samet, H., 1990a. *Applications of Spatial Data Structures: Computer Graphics, Image Processing and Geographical Information Systems*, Addison-Wesley, Reading, MA.

Samet, H., 1990b. *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA.

Schalkoff, R., 1992. *Pattern Recognition – Statistical, Structural and Neural Approaches*, John Wiley & Sons, New York.

Schneider, B., 2001. On the uncertainty of local shape of lines and surfaces, *Cartography and Geographic Information Science*, **28**, 237–247.

Schneider, B., 2003, Surface networks: extension of the topology and extraction from bilinear surface patches, In *Proceedings of the 7th International Conference on GeoComputation*, Southampton (On CD-ROM).

Schulze, J.P., Kraus, M., Lang, U., and Ertl, T., 2003. Integrating pre-integration into the shear-warp algorithm, In *Proceedings of Volume Graphics 2003*, Tokyo, 109–118.

Scott, P.J., 1992. The mathematics of motif combination and their use for functional simulation, *International Journal of Machine Tools and Manufacture*, **32**, 69–73.

Scott, P.J., 1998. Foundations of topological characterization of surface texture, *International Journal of Machine Tools and Manufacture*, **38**, 559–566.

Scott, P.J., 2004. Pattern analysis and metrology: the extraction of stable features from observable measurements, In: *Proceedings of the Royal Society of London, Series A*, in press.

Serra, J., and Vincent, L., 1992. An overview of morphological filtering, *Circuits Systems Signal Process*, **11**, 47–108.

Shekhar, R., Fayyad, E., Yagel, R., and Cornhill, J.F., 1996. Octree-based decimation of marching cubes surfaces. In *Proceedings of IEEE Visualization '96*, San Jose, CA, 335–342.

Shepherd, I.D.H., 1995. Putting time on the map: dynamic displays in data visualization and GIS, In: P.F. Fisher (Ed.), *Innovations in GIS 2*, Taylor & Francis, London.

Sherratt, G.G., 1960. A model for general urban growth, *Management Science Models and Technique*, **2**, 147–159.

Shinagawa, Y., and Kunii, T.L., 1991. Constructing a Reeb graph automatically from cross sections, *IEEE Computer Graphics & Applications*, **11**, 44–51.

Shinagawa, Y., Kunii, T.L., and Kergosien, Y.L., 1991. Surface coding based on morse theory, *IEEE Computer Graphics & Applications*, **11**, 66–78.

Sibson, R., 1981. A brief description of natural neighbor interpolations, In: V. Barnett (Ed.), *Interpreting Multivariate Data*, John Wiley & Sons, Chichester.

Sircar, J.K., and Cerbrian, J.A., 1986. Application of image processing techniques to the automated labelling of raster digitized contours, In *Proceedings of the 2nd International Symposium on Spatial Data Handling*, Seattle, WA, 171–184.

Slocum, T.A., Robeson, S.H., and Egbert, S.L., 1990. Traditional versus sequenced choropleth maps: an experimental investigation, *Cartographica*, **27**, 67–88.

Slocum, T., Yoder, S.C., Kessler, F.C., and Sluter, R.S., 2001. MapTime: software for exploring spatio-temporal data associated with point locations, *Cartographica*, **37**, 14–32.

Speight, J.G., 1976. Numerical classification of landform elements from air photo data, *Zeitschrift für Geomorphologie*, **25**, 154–168.

Takahashi, S., Ikeda, T., Shinagawa, Y., Kunii, T.L., and Ueda, M., 1995. Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data, *Computer Graphics Forum*, **14**, C181–C192.

Takahashi, S., Ohta, N., Nakamura, H., Takeshima, Y., and Fujishiro, I., 2002. Modeling surperspective projection of landscapes for geographical guide-map generation, *Computer Graphics Forum*, **21**, 259–268.

Takahashi, S., Shinagawa, Y., and Kunii, T.L., 1997. A feature-based approach for smooth surfaces, In *Proceedings of the 4th ACM Symposium on Solid Modeling and Applications*, ACM Press, New York.

Takahashi, S., Takeshima, Y., and Fujishiro, I., 2004. Topological volume skeletonization and its application to transfer function design, *Graphical Models*, **66**, 24–49.

Tang, L., 1992. Automatic extraction of specific geomorphological elements from contours, In *Proceedings of the 5th International Symposium on Spatial Data Handling*, Charleston, SC, 554–566.

Tarasov, S.P., and Vyalyi, M.N., 1998. Construction of contour trees in 3D in $O(n \log n)$, In *Proceedings of the 14th ACM Symposium on Computational Geometry*, Minneapolis, MN, 68–75.

Tobler, W.R., 1970. A computer movie simulating urban growth in the Detroit region, *Economic Geography*, **46**, 234–240.

Tufte, E., 1983. *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT.

Upson, C., and Kerlick, D., 1989. *Volumetric Visualization Techniques*, Association for Computing Machinery SIGGRAPH 1989 Course Notes, Tutorial No. 13, 1–86.

van Kreveld, M., 1996. Efficient methods for isoline extraction from a TIN, *Internatational Journal of Geographical Information Systems*, **10**, 523–540.

van Kreveld, M., van Oostrum, R., Bajaj, C., Pascucci, V., and Schikore, D., 1997. Contour trees and small seed sets for isosurface traversal, In *Proceedings of the 13th ACM Symposium on Computational Geometry*, Nice, 212–220.

van Oostrum, R., 1999. Geometric Algorithms for Geographic Information Systems, Ph.D. dissertation, Department of Computer Science, Utrecht University, Utrecht.

Vincent, L., and Soille, P., 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations, *IEEE Transactions on Pattern Analysis and Machine intelligence*, **13**, 583–598.

von Minusio, D.M., 2002. *Models and Experiments for Quality Handling in Digital Terrain Modelling*, Unpublished Ph.D. thesis, Department of Geography, University of Zurich, Zurich.

Wang, J., Robinson, G.J., and White, K., 2000a. Estimating surface net solar radiation by use of Landsat-5 TM and digital elevation models, *International Journal of Remote Sensing*, **21**, 31–43.

Wang, J., Robinson, G.J., and White, K., 2000b. Generating viewsheds without using sightlines, *Photogrammetric Engineering & Remote Sensing*, **66**, 87–90.

Ware, C., 2000. *Information Visualization: Perception for Design*, Morgan Kaufmann, San Francisco, CA.

Warntz, W., 1966. The topology of a socio-economic terrain and spatial flows, *Papers of the Regional Science Association*, **17**, 47–61.

Watson, G.S., 1972. Trend surface analysis and spatial correlation, *Geographical Society of America*, Special Paper, **146**, 39–46.

Watson, D.F., 1992. *Contouring: A Guide to the Analysis and Display of Spatial Data*, Pergamon, Oxford.

Weibel, R., and Dutton, G., 1999. Generalising spatial data and dealing with multiple representations, In: P.A. Longley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind (Eds.), *Geographical Information Systems: Volume 1 – Principles and Technical Issues*, John Wiley & Sons, New York.

Werner, C., 1988. Formal analysis of ridge and channel patterns in maturely eroded terrain, *Annals of the Association of American Geographers*, **78**, 253–270.

Westermann, R., and Ertl, T., 1998. Efficiently using graphics hardware in volume rendering applications, In *Proceedings of SIGGRAPH '98*, Orlando, FL, 169–177.

Westermann, R., Johnson, C., and Ertl, T., 2000. A level-set method for flow visualization, In *Proceedings of IEEE Visualization 2000*, Salt Lake City, UT, 147–154.

Westermann, R., Kobbelt, L., and Ertl, T., 1999. Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces, *The Visual Computer*, **15**, 100–111.

Whitten, E.H.T., 1974. Scale and directional field and analytical data for spatial variability studies, *Mathematical Geology*, **6**, 183–198.

Wilhelms, J., and van Gelder, A., 1992. Octrees for faster isosurface generation, *ACM Transactions on Graphics*, **11**, 201–227.

Williams, L., 1983. Pyramidal parametrics, *ACM Computer Graphics*, **17**, 1–11.

Wilson, J.P., and Gallant, J.C. (Eds.), 2000. *Terrain Analysis: Principles and Applications*, John Wiley & Sons, New York, Chichester.

Witkin, A.P., 1983. Scale-space filtering, In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, 1019–1022.

Wolf, G.W., 1984. A mathematical model of cartographic generalization, *Geo-Processing*, **2**, 271–286.

Wolf, G.W., 1988a. Weighted surface networks and their application to cartographic generalization, In: W. Barth (Ed.), *Visualisierungstechniken und Algorithmen*, Springer-Verlag, Heidelberg.

Wolf, G.W., 1988b. *Generalisierung topographischer Karten mittels Oberflaechengraphen*, Dissertation, Department of Geography, University of Klagenfurt, Klagenfurt, 250.

Wolf, G.W., 1989. A practical example of cartographic generalization using weighted surface networks, In: F. Dollinger, and J. Strobl (Eds.), *Angewandte Geographische Informationstechnologie*, Department of Geography, University of Salzburg, Salzburg.

Wolf, G.W., 1990. Metric surface networks, In *Proceedings of the 4th International Symposium on Spatial Data Handling*, Zurich, 844–856.
Wolf, G.W., 1991a. A FORTRAN subroutine for cartographic generalization, *Computers & Geosciences*, **17**, 1359–1381.
Wolf, G.W., 1991b. Characterization of functions representing topographic surfaces, In *Proceedings of AUTOCARTO 10*, Baltimore, MD, 186–204.
Wolf, G.W., 1992. Hydrologic applications of weighted surface networks, In *Proceedings of the 5th International Symposium on Spatial Data Handling*, Charleston, SC, 567–579b.
Wolf, G.W., 1993. Data structures for the topological characterization of topographic surfaces, In: D. Pumain (Ed.), Systèmes d'information géographique et systèmes experts, *Sixième Colloque européen de géographie théorique et quantitative*, GIP RECLUS, Montpellier.
Wood, J., 1996a. *The Geomorphological Characterisation of Digital Elevation Models*, Ph.D. thesis, Department of Geography, University of Leicester, Leicester.
Wood, J., 1996b. *Scale-based characterisation of Digital Elevation Models*, In: D. Parker (Ed.), Innovations in GIS 3, Taylor & Francis, London.
Wood, J., 1998. Modelling the continuity of surface form using digital elevation models, In *Proceedings of the 8th International Symposium on Spatial Data Handling*, Vancouver, 725–736.
Wood, J., 1999. Visualisation of scale dependencies in surface models, In *Proceedings of the International Cartographic Association Conference*, Ottawa.
Wood, J., and Rana, S., 2000. Constructing weighted surface networks for the representation and analysis of surface topology, In *Proceedings of the 5th International Conference on Geocomputation*, Chatham, UK (On CD-ROM).
Young, M., 1978. *Terrain Analysis: Program Documentation, Statistical Characterisation of Altitude Matrices by Computer*, Report No. 5 on Grant DA-ERO-591-73-G0040, University of Durham, Durham.

WEBSITE REFERENCES

URL #1 US Geological Survey, *DEM of the Crater Lake Area*, Oregon, 30 metres cell resolution [WWW document] http://seamless.usgs.gov/ (accessed 1 April 2003).
URL #2 McCool, M., *Sparse Texture Storage for Graphics Accelerators*, Technical Talk [WWW document] http://www.cgl.uwaterloo.ca/Projects/rendering/Talks/sparse/slides.pdf (accessed 1 April 2003).
URL #3 Center for International Earth Science Information Network. AIDS Data Animation Project [WWW document] http://www.ciesin.org/datasets/cdc-nci/cdc-nci.html (accessed 16 March 2002).
URL #4 Census Dissemination Unit. Surfpop v2.0: Background to census surface models [WWW document] http://census.ac.uk/cdu/software/surpop/background.html (accessed 17 March 2003).
URL #5 Discreet. 3D Studio Max [WWW document] http://www.discreet.com/products/3dsmax/ (accessed 16 March 2002).
URL #6 Wood, J., LandSerf: visualisation and analysis of terrain models [WWW document] http://www.soi.city.ac.uk/~jwo/landserf/ (accessed 16 March 2002).
URL #7 Dykes, J., and Rana, S., Augmenting visualization of dynamic raster surfaces [WWW document] http://www.soi.city.ac.uk/~jad7/snv/ (accessed 17 March 2003).
URL #8 Blok, C., and Kobben, B., A Web Cartography Forum: an evaluation site for visualization tools [WWW document] http://www.itc.nl/~carto/research/webcartoforum/ (accessed 16 March 2002).
URL #9 Johnson, C.K., Crystallographic topology: the topology of crystallographic groups and simple crystal structures [WWW document] http://www.ornl.gov/ortep/topology.html (accessed 1 April 2003).

Index

Topological Data Structures for Surfaces – An Introduction to Geographical Information Science. Edited by Sanjay Rana

ISBN: 0-470-85151-1

Plate 1 *Hakone area: (a) ridge and course lines (Reproduced from Takahashi, S., Ikeda, T., Shinagawa, Y., Kunii, T.L., and Ueda, M., (1995). Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data, Computer Graphics Forum,* **14***3, 181––192, Blackwell publishers, by permission of Blackwell publishers (b) the Reeb graph (Source: Takahashi, S., Ikeda, T., Shinagawa, Y., Kunii, T.L., and Ueda, M., (1995). Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data, Computer Graphics Forum,* **14***3, 181––192. Blackwell publishers), and (c) a surperspective guide-map image (Source: Takahashi, S., Ohta, N., Nakamura, H., Takeshima, Y., and Fujishiro, I., (2002). Modeling surperspective projection of landscapes for geographical guide-map generation, Computer Grpahics Forum,* **21***3, 259––268. Blackwell publishers)*

Plate 2 *Two terrain models (a) and (c) and their Reeb graph representations (b) and (d). The models in (a) and (c) are freely available at http://www.geographx.co.nz/*

Plate 3 *The population surface in Toyama-shi*

Plate 4 *Visualisation of a pion collision simulation. (top row) Grayscale colour map of the density field. (second row) Same field rendered with a hue-based colour map, which improves the visualisation of the data. (third row) Further enhancement of the visualisation with overlapping topology diagram. (bottom row) Reduced topology diagram highlighting the important structures of the data*

Plate 5 *SNV: surface network visualiser. The software application to visualise and interact with a sequence of surface networks*

Plate 6 *Grit from a grinding wheel* 1.0×1.0 *mm (a) initial critical points; (b) hill segmentation after 3% Wolf pruning; (c) after 5% Wolf pruning; and (d) 10% Wolf pruning*

Plate 7 *Anodised extruded aluminium* 0.5×0.5 *mm (a) initial critical points and (b) hill segmentation after Wolf pruning*