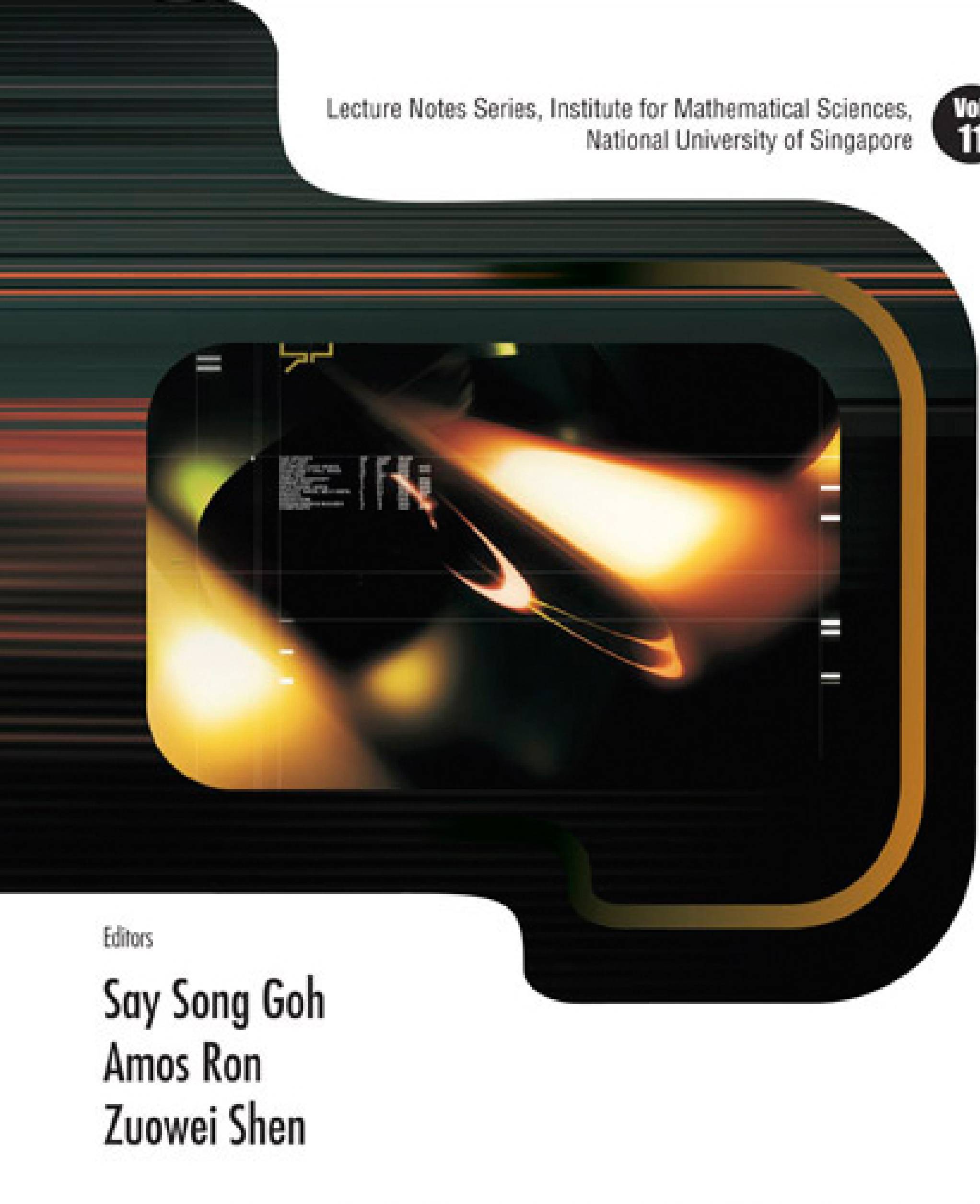
Lecture Notes Series, Institute for Mathematical Sciences,
National University of Singapore

Vol. 11



Editors

Say Song Goh
Amos Ron
Zuowei Shen

MATHEMATICS AND COMPUTATION IN IMAGING SCIENCE AND INFORMATION PROCESSING

MATHEMATICS AND COMPUTATION IN IMAGING SCIENCE AND INFORMATION PROCESSING

LECTURE NOTES SERIES
Institute for Mathematical Sciences, National University of Singapore

ISSN: 1793-0758

Published

Vol. 1 Coding Theory and Cryptology
edited by Harald Niederreiter

Vol. 2 Representations of Real and *p*-Adic Groups
edited by Eng-Chye Tan & Chen-Bo Zhu

Vol. 3 Selected Topics in Post-Genome Knowledge Discovery
edited by Limsoon Wong & Louxin Zhang

Vol. 4 An Introduction to Stein’s Method
edited by A. D. Barbour & Louis H. Y. Chen

Vol. 5 Stein's Method and Applications
edited by A. D. Barbour & Louis H. Y. Chen

Vol. 6 Computational Methods in Large Scale Simulation
edited by K.-Y. Lam & H.-P. Lee

Vol. 7 Markov Chain Monte Carlo: Innovations and Applications
edited by W. S. Kendall, F. Liang & J.-S. Wang

Vol. 8 Transition and Turbulence Control
edited by Mohamed Gad-el-Hak & Her Mann Tsai

Vol. 9 Dynamics in Models of Coarsening, Coagulation, Condensation and Quantization
edited by Weizhu Bao & Jian-Guo Liu

Vol. 10 Gabor and Wavelet Frames
edited by Say Song Goh, Zuowei Shen & Amos Ron

Vol. 11 Mathematics and Computation in Imaging Science and Information Processing
edited by Say Song Goh, Zuowei Shen & Amos Ron

Lecture Notes Series, Institute for Mathematical Sciences,
National University of Singapore

Vol.
11

MATHEMATICS AND COMPUTATION IN IMAGING SCIENCE AND INFORMATION PROCESSING

Editors

Say Song Goh
National University of Singapore, Singapore

Amos Ron
University of Wisconsin-Madison, USA

Zuowei Shen
National University of Singapore, Singapore

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

Library of Congress Cataloging-in-Publication Data
Mathematics and computation in imaging science and information processing /
editors Say Song Goh, Amos Ron, Zuowei Shen.
p. cm. -- (Lecture notes series, Institute for Mathematical Sciences, National University of Singapore ; v. 11)
Includes bibliographical references.
ISBN-13: 978-981-270-905-9 (hardcover : alk. paper)
ISBN-10: 981-270-905-3 (hardcover : alk. paper)
1. Image processing--Mathematical models. 2. Image processing--Data processing.
3. Electronic data processing--Mathematical models. I. Goh, Say Song. II. Ron, Amos.
III. Shen, Zuowei.

TA1637.M38 2007
621.36'7--dc22
2007019911

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

Printed in Singapore.

CONTENTS

This page intentionally left blank

FOREWORD

The Institute for Mathematical Sciences at the National University of Singapore was established on 1 July 2000. Its mission is to foster mathematical research, particularly multidisciplinary research that links mathematics to other disciplines, to nurture the growth of mathematical expertise among research scientists, to train talent for research in the mathematical sciences, and to provide a platform for interaction and collaboration between local and foreign mathematical scientists, in support of national development.

The Institute organizes thematic programs which last from one month to six months. The theme or themes of a program will generally be of a multidisciplinary nature, chosen from areas at the forefront of current research in the mathematical sciences and their applications, and in accordance with the scientific interests and technological needs in Singapore.

Generally, for each program there will be tutorial lectures on background material followed by workshops at the research level. Notes on these lectures are usually made available to the participants for their immediate benefit during the program. The main objective of the Institute's Lecture Notes Series is to bring these lectures to a wider audience. Occasionally, the Series may also include the proceedings of workshops and expository lectures organized by the Institute.

The World Scientific Publishing Company has kindly agreed to publish the Lecture Notes Series. This Volume, "Mathematics and Computation in Imaging Science and Information Processing", is the eleventh of this Series. We hope that through the regular publication of these lecture notes the Institute will achieve, in part, its objective of promoting research in the mathematical sciences and their applications.

April 2007

Louis H. Y. Chen
Ka Hin Leung
Series Editors

This page intentionally left blank

PREFACE

Rapid advances in communication, sensing and computational power have led to an explosion of data. The usefulness of this resource for human knowledge is determined by its accessibility and portability, which present fresh challenges to existing techniques in terms of transmission, storage, querying, display and numerical manipulation. As a result, much current research efforts focus on developing more advanced techniques for the representation, processing, analysis and interpretation of these data sets. This requires and gives rise to new theories and new methods in the areas of approximation, imaging science, information processing, mathematical modeling, scientific computing and statistics.

In view of these exciting developments, the program "Mathematics and Computation in Imaging Science and Information Processing" was held in Singapore at the Institute for Mathematical Sciences, National University of Singapore, from July to December 2003 and in August 2004 to promote multidisciplinary research on the mathematics in imaging science and information processing. In particular, the program emphasized on numerical methods in image and information processing, wavelet theory and its applications in image and signal processing, and time-frequency analysis and applications. Under the program, three conferences, six workshops, eleven tutorials and three public lectures were organized. A total of more than 340 participants took part in these activities including over 130 international attendees. We thank the Institute for Mathematical Sciences for its generous funding and efficient administrative support, without which the smooth implementation of the program would not be possible. We would also like to express our sincere appreciation to the authors for their contributions towards this volume.

The tutorials of the program, each comprising a series of lectures, were conducted by international experts, and they covered a wide spectrum of topics in the field of mathematical image, signal and information processing. This compiled volume contains survey articles by the tutorial speakers on subdivision in geometric modeling and computer graphics, high order numerical methods for time dependent Hamilton-Jacobi equations, variational methods in mathematical image processing, data hiding and image steganography, and the apriori algorithm in data mining. The accompanying volume is on Gabor analysis and wavelet theory, which are two fundamental mathematical tools in imaging science and information processing. It contains exposition articles by the tutorial speakers and also research papers. The two volumes collectively provide graduate students and researchers new to the field a comprehensive introduction to a number of important topics in mathematical image, signal and information processing. The chapters in each volume were written by specialists in their respective areas. The following outline the organization of this volume and highlight the topics presented.

During the last decade, subdivision surfaces based on arbitrary control meshes have become an important and quite pervasive tool for computer graphics and geometric modeling applications. Chapter 1 by D. Zorin provides an introduction to the algorithms and theory related to subdivision surfaces, covering both fundamentals and recent research developments. It surveys the basic concepts on subdivision surfaces and gives an overview of various important subdivision schemes for surfaces on arbitrary meshes, with focus on two of the most common schemes (Loop and Catmull-Clark). In addition, it reviews recent theoretical results on smoothness and approximation properties of subdivision surfaces on arbitrary meshes.

Hamilton-Jacobi equations have applications in many areas including image processing and computer vision. C.-W. Shu's Chapter 2 reviews several high order numerical methods for solving time dependent Hamilton-Jacobi equations. It begins with first order monotone schemes on structured rectangular meshes and unstructured meshes, which act as building blocks for high order schemes. The high order methods discussed include essentially non-oscillatory schemes for structured meshes, weighted essentially non-oscillatory schemes for both structured and unstructured meshes, and discontinuous Galerkin schemes for unstructured meshes.

Image deblurring is the recovery of a sharp image from its blurry observation. It appears in many different sectors of imaging science, including optical, medical and astronomical applications, and is a topic of active research in mathematical image processing. Regularization techniques are often used to handle this ill-posed inverse problem. Chapter 3, co-authored by T. F. Chan and J. Shen, presents a comprehensive account of image deblurring, highlighting its main modeling ideas and techniques. In particular, it describes various variational methods for image deblurring for both the situations of known and unknown point spread functions which model the blurs. Mathematical analysis is provided for the existence or uniqueness of solutions of these methods. The associated computational approaches are also developed.

With the increasing need to conceal information (for instance, secret data, copyright information and movie subtitles) within a host data set, data hiding is now a major research area in signal, image and video processing. It can be regarded as a game between the embedder/decoder and the attacker, who employ optimal data-hiding and attack strategies respectively. Chapter 4 by P. Moulin and R. Koetter reviews the fundamentals of the data-hiding problem. Various data-hiding algorithms are described, ranging from simple early codes to more modern codes based on information-theoretic binning concepts. The performance of these codes are analyzed in terms of probability of error and data-hiding capacity. As illustration of the theory, image watermarking examples are shown.

Related to data hiding is the topic of steganography, in which many new and powerful techniques have been developed in the last few years. Unlike cryptography which aims to make a message secure, the goal of steganographic techniques is to hide the presence of the message itself from an observer. Due to their high degree of redundancy present, digital images are common objects used as carriers of embedded messages. Chapter 5, co-authored by M. Kharrazi, H. T. Sencar and N. Memon, is a tutorial on image steganography and steganalysis. It first introduces some general concepts and ideas in the topic of steganography, with discussions on steganographic security and capacity. Then it focuses on image steganography and steganalysis, reviewing recent techniques for embedding messages and detecting presence of messages in images.

Finally, M. Hegland's Chapter 6 is on the field of data mining. Large amount of data sets are generated by day-to-day management in various sectors such as business, finance, administration and social services. Originated from market basket analysis, association rules are currently one of the most popular tools in data mining which is about extracting useful information from these data sets. Chapter 6 gives a tutorial on the apriori algorithm for efficient association rule discovery. It begins with fundamentals of association rule discovery, and the mathematical model derived provides a framework for the apriori algorithm. Then the apriori algorithm and several of its extensions are discussed in detail, giving much insight into this important approach of data mining.

Say Song Goh
National University of Singapore, Singapore

Amos Ron
University of Wisconsin-Madison, USA

Zuowei Shen
National University of Singapore, Singapore

SUBDIVISION ON ARBITRARY MESHES: ALGORITHMS AND THEORY

Denis Zorin
New York University
719 Broadway, 12th Floor, New York, USA
E-mail: dzorin@mrl.nyu.edu

Subdivision surfaces have become a standard geometric modeling tool for a variety of applications. This survey is an introduction to subdivision algorithms for arbitrary meshes and related mathematical theory; we review the most important subdivision schemes, the theory of smoothness of subdivision surfaces, and known facts about approximation properties of subdivision bases.

1. Introduction

This survey is based on a series of lectures presented at the IMS-IDR-CWAIP Joint Workshop on Data Representation at the National University of Singapore in August 2004.

Our primary goal is to present a brief introduction to the algorithms and theory related to subdivision surfaces from basic facts about subdivision to more recent research developments. This tutorial is intended for a broad audience of computer scientists and mathematicians. While not being comprehensive by any measure, it aims to provide an overview of what the author considers the most important aspects of subdivision algorithms and theory as well as provide references for further study.

A large variety of algorithms and a comprehensive theory exist for subdivision schemes on *regular grids*, which are only briefly mentioned in this survey. Subdivision on regular grids, being closely related to wavelet constructions, has an important applied role in many applications. However, ability to handle *arbitrary* control meshes was one of the primary reasons for the rapid increase in popularity of subdivision for computer graphics and geometric modeling applications during the last decade. This motivates our focus on schemes designed for such meshes.

We start with a brief survey of applications of subdivision in computer graphics and geometric modeling in Section 1. In Section 2, we introduce the basic concepts for both curve and surface subdivision. In the third section we review different types of subdivision rules focusing on the most commonly used in practice (Loop and Catmull-Clark subdivision).

In contrast to the regular case, fewer general theoretical results and tools are available for subdivision schemes on arbitrary meshes; in many aspects the theory is somewhat behind the practice. The most important theoretical results on smoothness and approximation properties of subdivision surfaces are reviewed in Sections 5 and 6.

Sections 2–5 are partially based on the notes for the SIGGRAPH course “Subdivision for Modeling and Animation” co-taught by the author in 1998-2000. There is a number of excellent books and review articles on subdivision which the author highly recommends for further reading: the monograph of Cavaretta et al. [11] on subdivision on regular grids, survey articles by Dyn and Levin [18,19], the book by Warren and Weiner [81], the articles by Sabin[67,66] and Schröder [71,72].

1.1. *Subdivision in computer graphics and geometric modeling*

The idea of constructing smooth surfaces from arbitrary meshes using recursive refinement was introduced in papers by Catmull and Clark [10] and Doo and Sabin [17] in 1978. These papers built on subdivision algorithms for regular control meshes, found in the spline literature, which can be traced back to late 40s when G. de Rham used “corner cutting” to describe smooth curves.

Wide adoption of subdivision techniques in computer graphics applications occurred in the mid-nineties: with an increase in complexity of the models, the need to extend traditional NURBS-based tools became apparent.

Constructing surfaces through subdivision elegantly addresses many issues with which computer graphics and computer-aided design practitioners are confronted. Most importantly, the need to handle control meshes of *arbitrary topology*, while maintaining surface smoothness and visual quality automatically. Subdivision surfaces easily admit multiresolution extensions, thus enabling efficient hierarchical representations of complex surfaces. At the same time, most popular subdivision schemes extend splines (and produce piecewise-polynomial surfaces for regular control meshes), thus maintaining continuity with previously used representations and inheriting some of the appealing qualities of splines. Another important advantage of subdivision surfaces is that simple local modifications of subdivision rules make it possible to introduce surface features of many different types [25,8]. Finally, subdivision surfaces can be extended to hierarchical representations either of wavelet [47], pyramid type [92], or related displaced subdivision surfaces [38].

Over the past few years, a number of crucial geometric algorithms were developed for subdivision surfaces and subdivision-based multiresolution representations. One of the important steps that enabled many practical applications was development of direct evaluation methods [75], that made it possible to evaluate, in constant time, recursively defined subdivision surfaces at arbitrary points. Algorithms were developed for trimming [43], performing boolean operations [7], filleting and blending [84,56], fitting [34], computing surface volumes [58], lofting [53,54,55,69] and other operations. Subdivision surfaces were demonstrated to be a useful tool for complex interactive surface editing [36,92,9,30].

Subdivision surfaces became a mature technology, used in a variety of applications. Examples of applications include representing and registering complex range scan data [2], face modeling [74,44] and three dimensional extensions of subdivision used in large-scale visualization [41,4].

As subdivision algorithms can be used to define bases on arbitrary mesh domains, they are a natural candidate for higher-order finite element calculations for engineering applications, shell problems in particular. First steps in this direction were made in [12,13]. Natural refinement structure of subdivision surfaces leads to adaptive hierarchal finite element constructions [35]. Subdivision-based mesh generation for FEM is explored in [39,40].

2. Basics

In this section we introduce the basic concepts of subdivision needed to define various subdivision schemes considered in Section 3.

2.1. *Subdivision curves*

The goal of this section is to introduce the basic concepts using subdivision curves as an example. The apparatus of subdivision matrices we introduce is not essential for curves, as the same formulas can be obtained by other means; however, it is indispensable for subdivision surfaces.

Subdivision algorithm. We can summarize the basic idea of subdivision as follows: subdivision defines a smooth curve or surface as the limit of successive refinements of an initial sequence of control points.

In this section, to simplify exposition, we only consider curves defined by infinite sequences of control points indexed by integers and only one type of refinement: a new control point is added to the sequence between two old control points and the positions of old points are recomputed (Figure 1).

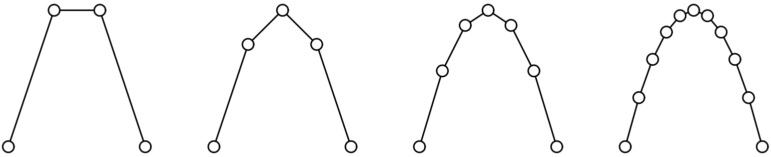


Fig. 1. Subdivision steps for a cubic spline.

The numbering for the refined sequence is chosen so that the point i in the original sequence has even number $2i$ in the new sequence. We use notation p^j for the sequence of control points after j subdivision steps.

The most general definition of a linear subdivision rule is that it is a collection of linear maps S^j, mapping p^j to p^{j+1}. In this survey we consider subdivision rules which satisfy two additional requirements: the rules are *stationary* and have finite support.

More formally, for the type of one-dimensional refinement described above, *stationary subdivision rules* can be specified by two sequences of coefficients $\{a^e_i, | i \in \mathbf{Z}\}$ and $\{a^o_i | i \in \mathbf{Z}\}$ which are usually referred to as even and odd masks. For a given sequence of control points $p = (p_i \in \mathbf{R}^n, i \in \mathbf{Z})$, a single subdivision step produces a new refined sequence p' of control points p'_i, defined by

$$\begin{aligned} p'_{2i} &= \sum_{j \in \mathbf{Z}} a^e_{i-j} p_j \\ p'_{2i+1} &= \sum_{j \in \mathbf{Z}} a^o_{i-j} p_j \end{aligned} \tag{1}$$

For our choice of numbering, the even-numbered points correspond to the repositioned original control points, and odd-numbered points are the newly added points. For stationary subdivision, the linear map from p^j to p^{j+1} does not depend on the level, i.e. there is a single linear operator S, such that $p^{j+1} = Sp^j$.

The rules have *finite support* if only a finite number of coefficients a_i^o and a_i^e are nonzero. The set of indices for which the mask coefficients are not zero is called *mask support.*

The most common subdivision scheme for uniform cubic B-splines has masks with nonzero entries $(1/8, 3/4, 1/8)$ with indices $(-1, 0, 1)$ and $(1/2, 1/2)$ with indices $(-1, 0)$, for even and odd control points respectively (Figure 1).

We can view the initial control points p^0 as values assigned to integer points in $\mathbf{R}$. It is natural to assign control points p^1 to half-integers, and in general control points p^j to points of the form $i/2^j$ in $\mathbf{R}$.

For each subdivision level j we then have a unique piecewise linear function $L[p^j]$, defined on $\mathbf{R}$ which interpolates the control points p^j: $L(i/2^j) = p_i^j$. We say that the subdivision scheme *converges* if for any initial control points p^0, the associated sequence of piecewise linear functions $L[p^{(j)}]$ converges pointwise.

In particular, for the cubic spline masks defined above, the limit curve is a cubic polynomial on each integer interval $[i, i+1]$. The reason for this is that this set of masks is derived from the well-known refinement relation for uniform cubic B-splines:

$$B(t) = \frac{1}{8}\left(B(2t-2) + 4B(2t-1) + 6B(2t) + 4B(2t+1) + B(2t+2)\right). \tag{2}$$

A cubic spline curve has the form $\sum_{i \in \mathbf{Z}} p_i B(t-i)$; applying the refinement relation (2) to $B(t-i)$ and collecting the terms, we obtain

$$\sum_{i \in \mathbf{Z}} p_i B(t-i) = \sum_{i \in \mathbf{Z}} p_i' B(2t-i)$$

with $p_{2i}' = (1/8)(p_{i-1} + 6p_i + p_{i+1})$ and $p_{2i+1}' = (1/2)(p_i + p_{i+1})$, i.e. with p_i' defined by the subdivision rules stated above. We conclude that sequences p_i and p_i' *define the same spline curve.* However, the refined control points p_i' correspond to scaled basis functions $B(2t)$ with smaller support and are spaced closer to each other. As we refine, we get control points for the same cubic curve $f(t)$ but split into shorter polynomial segments. One can show the piecewise linear functions, connecting the control points, converge to $f(t)$ pointwise.

While spline subdivision is a starting point for many subdivision constructions, deriving subdivision masks from spline refinement is not essential for obtaining convergent schemes or schemes producing smooth curves or surfaces. For example, one can replace the $(1/8, 3/4, 1/8)$ rule by three perturbed coefficients $1/8 - w, 3/4 + 2w, 1/8 - w)$, and still maintain convergence and tangent continuity of limit curves for sufficiently small w. However, the limit curves for the modified rules in general cannot be expressed in closed form.

Modified coefficients are usually chosen to meet a set of requirements necessary for desirable scheme behavior. The most basic requirement is

Affine invariance. If the points of sequence q are obtained by applying an affine transformation T to points of p, then $[Sq]_i = T[Sp]_i$, $i \in \mathbf{Z}$.

By considering translations by t, $q_i = p_i + t$, and substituting into the subdivision rules 1, we immediately obtain that the coefficients of masks should sum up to one:

$$\sum_{i \in \mathbf{Z}} a_i^e = 1, \quad \sum_{i \in \mathbf{Z}} a_i^o = 1.$$

In other words, the subdivision operator S should have a eigenvector with constant components $p_i = 1$, for all i, and eigenvalue 1. It can also be shown this is necessary (but not sufficient) condition for convergence.

Subdivision matrices. As we have seen above, a subdivision step can be represented by a linear operator acting on sequences. It is often useful to consider local subdivision matrices of finite dimension. Such matrices have an important role, both in practice and in theory, as they can be used for limit control point positions and tangent vectors and analysis of convergence and continuity. These local matrices are restrictions of the infinite subdivision matrices to *invariant neighborhoods* of points.

Fix an integer i; then the invariant neighborhood N_m of size m for i is the set of indices $\{i - m, \dots i + m\}$, such that the control points p_j^1, $j = 2i - m \dots 2i + m$, can be computed using only points p_i^0, for $i \in N_m$. The minimal size of the invariant neighborhood depends only on the support of the masks. For example, the minimal size m for the cubic B-spline subdivision rules is 1 because one can compute points p_{2i-1}^1, p_{2i}^1 and p_{2i+1}^1 given points p_{i-1}^0, p_i^0 and p_{i+1}^0.

We often need to consider invariant neighborhoods of larger size, such that the control points in the neighborhood define the curve completely on some interval containing the point of interest. For cubic splines, a curve segment, corresponding to an integer interval $[i, i+1]$, requires four control points. To obtain a part of the curve, containing i in the interior of its domain, we need to consider both $[i-1,i]$ and $[i,i+1]$ for a total of five points, which correspond to the neighborhood of size 2.

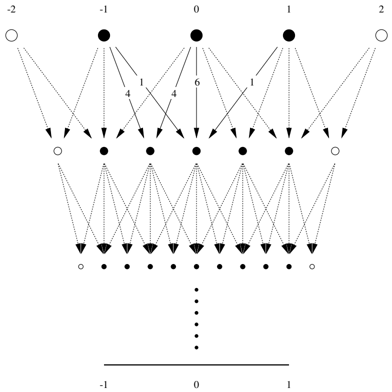


Fig. 2. In the case of cubic B-spline subdivision, the invariant neighborhood is of size 2. It takes 5 control points at the coarsest level to determine the behavior of the subdivision limit curve over the two segments adjacent to the origin. At each level, we need one more control point on the outside of the interval $t \in [-1,1]$ in order to continue on to the next subdivision level. 3 initial control points for example would not be enough.

The subdivision rules for computing five control points, centered at i, on level $j+1$ from five control points, centered at i on level j can be written as

$$\begin{pmatrix} p_{2i-2}^{j+1} \\ p_{2i-1}^{j+1} \\ p_{2i}^{j+1} \\ p_{2i+1}^{j+1} \\ p_{2i+2}^{j+1} \end{pmatrix} = \frac{1}{8} \begin{pmatrix} 1\,6\,1\,0\,0 \\ 0\,4\,4\,0\,0 \\ 0\,1\,6\,1\,0 \\ 0\,0\,4\,4\,0 \\ 0\,0\,1\,6\,1 \end{pmatrix} \begin{pmatrix} p_{i-2}^{j} \\ p_{i-1}^{j} \\ p_{i}^{j} \\ p_{i+1}^{j} \\ p_{i+2}^{j} \end{pmatrix}.$$

The 5 by 5 matrix in this expression is the *subdivision matrix.* If the same subdivision rules are used everywhere, this matrix does not depend on the choice of i.

The eigenvalues and eigenvectors of the subdivision matrix allow one to analyze how the control points in the invariant neighborhoods change from level to level.

Suppose an $n \times n$ subdivision matrix is non-defective, i.e. has n independent eigenvectors x_i, $i = 0, \ldots n-1$. Then, any vector of initial control points p can be written as a linear combination of eigenvectors of the matrix: $p = \sum_{i=0}^{n} a_i x_i$. The coefficients a_i can be computed using eigenvectors as

$$a_i = (l_i \cdot p),$$

using the dual basis of left eigenvectors l_i, $i = 0 \ldots n-1$, satisfying $(x_i \cdot l_k) = \delta_{ik}$. In this form, the result of applying the subdivision matrix j times, i.e. the control points on j-th subdivision level in the invariant neighborhood, can be written as

$$S^j p = \sum_{i=0}^{n} \lambda^j a_i x_i \tag{3}$$

where λ_i, $i = 0 \ldots n-1$, are the eigenvalues.

Limit positions. One can immediately observe that for convergence it is necessary that all eigenvalues of the matrix have magnitudes no greater than one. Furthermore, one can easily show that if there is more than one eigenvalue of magnitude one, the scheme does not converge either. At the same time, $\lambda_0 = 1$ is an eigenvalue corresponding to eigenvector $[1, 1, \ldots 1]$. The reason is that multiplying S by this eigenvector is equivalent to summing up the entries in each row, and by affine invariance, these entries sum up to one.

Next, we observe that for $i \geq 1$, $|\lambda_i| < 1$, all terms excluding the first on the right-hand side of (3) vanish, leaving only the term $a_0 x_0 = [a_0, a_0, \ldots a_0]$. This means that in the limit, all points in the invariant neighborhood approach a_0, i.e. a_0 is the value of the limit subdivision curve at the center of the invariant neighborhood.

Tangent vectors. If we further assume that $|\lambda_1| > |\lambda_2|$ and λ_2 is real and positive, consideration of the first two dominant terms in (3) makes it possible to compute the tangent to the curve under some additional conditions on the subdivision scheme, which will be considered in Section 5 for surfaces. Consider the vector of differences $S^j p - a_0 x_0$ between all points in the invariant neighborhood at level j and the center of the invariant neighborhood. if we scale this vector by $1/\lambda_1$, it converges to $a_1 x_1 = [a_1 x_1^1, a_1 x_1^2, \ldots a_1 x_1^n]$, i.e. all limit difference vectors are collinear and parallel to a_1. This suggests (but does not guarantee without additional assumptions, which hold for most common schemes) that $a_1 = (l_1 \cdot p)$ is a tangent vector to the curve.

The observations above show the left eigenvectors, corresponding to the eigenvalue 1 and the second largest eigenvalue λ_1, play a special role, defining the limit positions and tangents for a subdivision curve.

Example. The eigenvalues and eigenvectors of the subdivision matrix for cubic splines are

$$(\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4) = \left(1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right)$$

$$(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = \begin{pmatrix} 1 & -1 & 1 & 1 & 0 \\ 1 & -\frac{1}{2} & \frac{2}{11} & 0 & 0 \\ 1 & 0 & -\frac{1}{11} & 0 & 0 \\ 1 & \frac{1}{2} & \frac{2}{11} & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

The left eigenvectors of eigenvalue 1 and subdominant eigenvalue 1/2 are $[0, 1/6, 2/3, 1/6, 0]$ and $[0, -1, 0, 1, 0]$, which yield the formulas for the curve point and tangent

$$a_0 = \frac{1}{6}(p_{i-1} + 4p_i + p_{i+1}); \quad a_1 = p_{i+1} - p_{i-1},$$

which coincide with the formulas obtained by direct evaluation of cubic B-spline curves.

2.2. *Subdivision surfaces*

Most of the concepts we have introduced for subdivision curves can be extended to surfaces, but significant differences exist. While the control points for a curve have a natural ordering, this is no longer true for arbitrary meshes. Furthermore, for an arbitrary mesh, local mesh structure may vary: e.g. a vertex can share an edge with an arbitrary number of neighbors, rather than only one or two, as is the case for a curve and the polygonal faces of the mesh which may have different numbers of sides. A finer mesh can be obtained from a given coarser mesh in many different ways. Thus, in the case of subdivision for meshes, one needs to define *refinement rules*, which specify how the connectivity of the mesh is changed when it is refined, and *geometric rules*, which specify the way the control point positions are computed for the refined mesh.

Another important difference is that while the curves can always be considered to be functions on a domain in **R**, there is no simple natural domain for surfaces. To be able to define subdivision surfaces as a limit of refinement, we need to construct a suitable domain out of the control mesh of the surface. We start with a specific example, the Loop subdivision scheme, to motivate the formal constructions we need to introduce.

Refinement of triangular manifold meshes. This scheme uses *triangular manifold control meshes.* Such control mesh consists of a *complex* K, which is a triple (V, E, F) of sets of vertices, edges and faces, and control points p^0, associated with each vertex in V. We use notation $p^j(v)$ for a control point at refinement level j associated with vertex v. The sets of vertices, edges and faces satisfy the following constraints:

- each edge is a pair of distinct vertices;
- each face is a set of three distinct vertices;
- each pair of vertices of a face is an edge;
- the intersection of two faces is either empty or an edge;
- each edge belongs to exactly two faces;
- the link of a vertex v (the set of edges of all faces containing v, excluding the edges that contain v themselves) can be ordered cyclically such that each two sequential edges share a vertex.

Two complexes are *isomorphic* if between their vertices there is a one-to-one map, which maps faces to faces and edges to edges.

Similarly to the curve case, we define neighborhoods on meshes. A 1-neighborhood $N_1(v, K)$ of a vertex v is a set of faces, consisting of all triangles, containing v. A 1-neighborhood $N_1(G, K)$ of a set of faces G consists of all triangles of 1-neighborhoods of the vertices of G. An m-neighborhood $N_m(v, K)$ is defined recursively as 1-neighborhood of $m - 1$ neighborhood.

The most common refinement rule for such meshes is *face quadrisection.* The new mesh is formed as follows: all old vertices are retained; a new vertex is added for each edge, splitting it into two; each edge is replaced by two new edges and each face by four new faces. One can easily see that all new vertices inserted using this refinement rule have valence 6, and only the vertices of the original mesh may have a different valence. The vertices of valence 6 are called *regular*, and the vertices of other valences are called *extraordinary.*

The Loop subdivision scheme. To define how the control points are computed, we need to specify rules for updating the positions of existing control points and for computing newly inserted control points.

These rules for the Loop subdivision scheme are shown in Figure 3. The rule for a vertex, inserted on an edge e, uses the control points for two

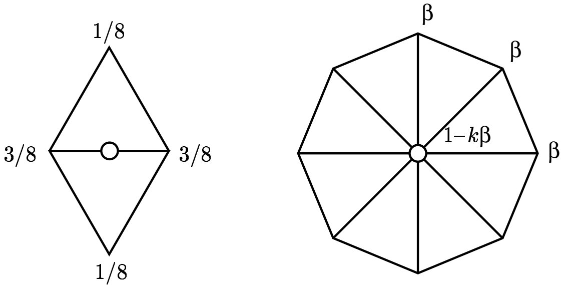


Fig. 3. Loop subdivision masks for new control points and updated positions of old control points. Vertices, for which control points are computed, are marked with circles.

triangles sharing e:

$$p^{j+1}(w) = \frac{3}{8}p^j(v_1) + \frac{3}{8}p^j(v_2) + \frac{1}{8}p^j(v_3) + \frac{1}{8}p^j(v_4),$$

where v_1, v_2 are edge endpoints, and v_3 and v_4 are the two remaining vertices of triangles sharing e.

The rule for updating positions of existing vertices is actually a parametric family of rules, with coefficients depending on the valence k of the vertex.

$$p^{j+1}(v) = (1 - k\beta) + \beta \sum_{v_i \in N_1(v)} p^j(v)$$

where β can be taken to be $3/8k$, for $k > 3$, and $\beta = 1/16$ for $k = 3$ (this is the simplest choice of β different choices of β are possible).

If the mesh is fully regular, i.e. all vertices have valence 6, these rules reduce to the subdivision rules for quartic box splines and can be derived from scaling relations similar to (2).

We note that these rules only depend on the local structure of the mesh, using only points within a fixed-size neighborhood of the point being computed: if we measure the neighborhood size in the refined mesh, both types of rules use level j control points, corresponding to vertices within the 2-neighborhood at level $j + 1$; this is the analog of finite support in the curve case.

Furthermore, we observe that the rules depend only on the mesh structure of the 1-neighborhood of the vertex (specifically, the number of adjacent vertices), not on the subdivision level, or vertex numbering. This is the analog of being stationary in the curve case. We will give a more precise definition below.

To reason about convergence of this scheme, we also need to define the piecewise linear interpolants similar to $L[p^j]$, defined for curves. Unfortunately, there is no natural way to map the vertices of an arbitrary mesh to points in the plane or some other standard domain, so one cannot use a similar simple construction. For mesh subdivision to be able to define the limit surfaces rigorously, we need to construct special domains for each complex; subdivision surfaces are defined as functions on these domains.

Domains for subdivision surfaces. The simplest construction of the domain for the subdivision surface requires an additional assumption. For triangular meshes, the control points p^0 in $\mathbf{R}^n$ can be used to define an geometric realization of a complex. Each face of K (i.e. a triple of vertices (u, v, w)) corresponds to the triangle in $\mathbf{R}^n$, defined by three control points $(p^0(u), p^0(v), p^0(w))$. We additionally require that no two control points coincide, and for any two triangles in $\mathbf{R}^n$, corresponding to faces of K, their intersection is either a control point, a triangle edge, empty, or, informally, the initial control mesh has no self-intersections. With this additional assumption, one can use the initial mesh as the domain on which the linear interpolants of control points at different levels of refinement are defined. We denote this domain $|K^0|$.

The initial control points p^0 are already associated with the points in the domain (the control points themselves). It remains to associate the control points on finer levels with points on the initial mesh. This can be done recursively. Suppose a vertex w of the refined complex K^{j+1} is inserted on the edge connecting vertices u and v of K^j. Suppose these vertices are already associated with points $t(u)$ and $t(v)$ on $|K^0|$, contained in the same triangle T of $|K^0|$. Then we associate w with the midpoint $(1/2)(t(u)+t(v))$, which, by convexity, is also contained in the same triangle T. It is easy to show that no two vertices can be assigned to the same point in the domain: the points obtained after j refinement steps form a regular grid on each triangle of $|K^0|$.

Now we can define the piecewise linear interpolants, similar to the ones used for curves. Fix a refinement level j and a triangle T of $|K^0|$. The vertices of K^j form a regular grid on T, with triangles corresponding to faces of K^j. For points of $|K^0|$ inside each subtriangle (u, v, w) of T, we define $L[p^j]$ to be the linear interpolant between $p^j(u)$, $p^j(v)$ and $p^j(w)$.

In this way, we obtain a sequence of functions $L[p^j]$ defined on $|K^0|$; the limit subdivision surface is a the pointwise limit of this sequence, if it exists. Thus the subdivision surface is defined as a function on $|K^0|$ with values in $\mathbf{R}^n$.

Stationary subdivision in 2D. The Loop subdivision scheme is an example of a stationary subdivision scheme. More generally, for any complex K and its refinements K^j, $K^0 = K^j$, a linear subdivision scheme gives a sequence of linear operators $S^j(K)$, mapping control points for vertices V^j to control points for vertices V^{j+1}. This means that for a given vertex w of K^{j+1},

$$p^{j+1}(w) = \sum_{v \in V} a_{vw} p^j(v) \tag{4}$$

We say that a scheme is *finitely supported* if there is an M, such that for any w and $v \notin N_M(w, K^{j+1})$, $a_{vw} = 0$. The *support* $\operatorname{supp} w$ of the mask of the scheme at w is the minimal subcomplex containing all vertices v of K^j such that $a_{vw} \neq 0$. We say that the scheme is stationary or invariant, with respect to isomorphisms, if the coefficients a_{vw} coincide for vertices, for which supports are isomorphic. More precisely, if there is an isomorphism $\iota : \operatorname{supp} w_1 \to \operatorname{supp} w_2$, and $\iota(w_1) = w_2$ then $a_{\iota(v)w_2} = a_{vw_1}$. The invariance can be also defined with respect to a restricted set of isomorphisms, e.g. if the mesh is tagged.

Subdivision matrices in 2D. The definition of invariant neighborhoods and the construction of subdivision matrices for subdivision on meshes is completely analogous to the curve case. However, the size of the matrix is variable and depends on the number of points in the invariant neighborhood. Another difference is related to the fact that invariant neighborhoods may not exist for a finite number of initial subdivision levels, as the mesh structure changes with each refinement. For a given neighborhood size m, however, after a sufficient number of subdivision steps, each extraordinary vertex v is surrounded by sufficiently many layers of regular vertices, and m-neighborhoods of v on different subdivision levels are similar.

For example, for the Loop scheme, the invariant neighborhood size is 2. For a vertex of valence k, it contains $3k + 1$ vertices. The subdivision matrix has the following general form:

$$\left(\begin{array}{c|ccc} 1 - k\beta & a_{01}^T & 0 & 0 \\ a_{10} & A_{11} & 0 & 0 \\ a_{20} & A_{21} & A_{22} & 0 \\ a_{30} & A_{32} & A_{32} & A_{33} \end{array}\right),$$

where all vectors a_{ij} are of length k and have constant elements ($a_{01} = \beta\mathbf{1}$, $a_{02} = (3/8)\mathbf{1}$, $a_{02} = (1/8)\mathbf{1}$, $a_{03} = (1/16)\mathbf{1}$. The blocks A_{ij} are cyclic $k \times k$, defined as follows,

$$A_{11} = \frac{1}{8}\text{Cyclic}(3, 1, 0, \ldots 0, 1),$$
$$A_{21} = \frac{1}{8}\text{Cyclic}(3, 3, 0 \ldots 0), \quad A_{22} = \frac{1}{8}\text{Cyclic}(1, 0, \ldots 0),$$
$$A_{31} = \frac{1}{16}\text{Cyclic}(10, 1, 0, \ldots 0, 1), \quad A_{32} = \frac{1}{16}\text{Cyclic}(1, 0, \ldots 0, 1),$$
$$A_{33} = \frac{1}{16}\text{Cyclic}(1, 0, \ldots 0).$$

Limit positions and tangent vectors in 2D. The computation of the limit positions for mesh subdivision scheme is the same as for curves: one needs to compute the dot product of the left eigenvector of eigenvalue 1 with the vector of control points in the invariant neighborhood.

The computation of tangent vectors is slightly different. Instead of a unique tangent vector, a smooth subdivision surface has at least two nonuniquely defined independent tangent vectors spanning the tangent plane. In the case of surfaces, we further assume that the eigenvalues of the subdivision matrix satisfy $1 = |\lambda_0| > |\lambda_1| \geq |\lambda_2| > |\lambda_3|$ and $\lambda_{1,2}$ are real. This is not necessary for tangent plane continuity, but this assumption commonly holds and greatly simplifies the exposition. In this case, again under some additional assumptions to be discussed in Section 5, one can compute the tangent vectors to the surface using right eigenvectors l_1 and l_2, corresponding to the eigenvalues λ_1 and λ_2.

For the Loop scheme, the masks for limit positions and tangent vectors are quite simple: both have supports in the 1-neighborhood of a vertex. The coefficients of the mask for the limit position, i.e. the entries of the left eigenvector l_0, have the same form as the vertex rule, with β replaced with $\beta_{limit} = 8\beta/(3 + 8k\beta)$. The two tangent masks l_1 and l_2 can be chosen to be $\cos 2\pi j/k$ and $\sin 2\pi j/k$ for the vertices of 1-neighborhood distinct from the center indexed by j. The coefficient for the center itself is 0. This choice is not unique: for the Loop scheme and most other commonly used schemes, $\lambda_1 = \lambda_2$, and any linear combination $c_1 l_1 + c_2 l_2$ is also a left eigenvector.

3. Overview of Subdivision Schemes

In this section we review a number of stationary subdivision schemes generating C^1-continuous surfaces on arbitrary meshes. Our discussion is not exhaustive even for stationary schemes. We discuss two most common schemes (Loop and Catmull-Clark) and their variations in considerable detail, and briefly several examples of other types of schemes; more detailed information on other schemes can be found in provided references.

3.1. *Classification of subdivision schemes*

Refinement rules. The variety of stationary subdivision schemes for surfaces is primarily due to the many possible ways to define refinement of complexes. Several classifications of refinement rules (e.g. [27,1,24]) were proposed; our discussion mostly follows [27].

Almost all refinement rules are extensions of refinement rules for periodic tilings of the plane. The principal reason is there is an extensive theory for analysis of subdivision on regular planar grids which can be used to analyze the surface, constructed from an arbitrary mesh everywhere excluding a set of isolated points.

A single refinement step typically maps a tiling to a finer tiling, which is obtained by scaling and optionally rotating the original tiling; however, some schemes may alternate between different tiling types.

All known schemes with one exception are based on refinements of *regular monohedral tilings*, for which all tiles are regular polygons. The 4-8 scheme [80,79], originally formulated using a tiling with right triangles, can be reformulated using regular quad tilings, i.e. it also fits into this category. There are only three regular tilings: triangular, quadrilateral and hexagonal. Hexagonal tilings are rarely used, and stationary schemes for such tilings were considered in detail only recently [14,85,57].

Once the tiling is fixed, there are still many ways to define how it is refined, even if we require that the refined tiling is of the same type. Dodgson [15] lists a set of heuristics that are typically used to limit the variety of possible refinement rules. Here we briefly review these heuristics and their motivation.

1 Refinement of regular tilings is used. While other tiling types, such as periodic (e.g. Laves or Archimedes tilings) or aperiodic (Penrose tilings) can be considered, all schemes proposed so far meet this requirement.

2/3 A refinement rule either maps all vertices of the original tiling to the vertices of the refined tiling, or it maps them to the face centers of the refined tiling. Again, it is possible to consider other types of rules, but all known schemes are in one of these categories.

4 If a point is a center of rotational symmetry of order k in the tiling (i.e. the rotations by $2\pi j/k$ around this vertex map the tiling to itself),

then in the refined tiling, it should be a center of rotational symmetry of at least the same order. If this requirement is not satisfied, one can show that the result of refinement depends on the way the vertices of a tiling are enumerated. Given the first 3 heuristics, this heuristic excludes refinement rules, mapping triangle vertices to centers, and hexagon centers to vertices.

5 For some number s, s times refined tiling is aligned with the original tiling, i.e. is obtained by uniform scaling. This is also justified by symmetry considerations, although, as pointed out in [15] is not strictly necessary. However, all schemes satisfying heuristic 7 in the stronger form that we use also satisfy this heuristic.

6 Triangle and quadrilateral schemes are generally useful but hexahedral schemes are more limited in their applications. One reason is that hexahedral tiling does not contain any multiple-edge straight lines, which can be used for meshes with boundaries and features.

7 Low *arity* (the ratio of the edge length of the refined tiling to the original tiling) is preferable. According to [15] arities higher than four are not likely to be useful. All practical and most known schemes, with exception of three recently proposed schemes, have arity two or less. As schemes of high arities result in very rapid decrease in the edge length, which is often undesirable, it is likely that only schemes of arity two or less will be used in applications.

These heuristics reduce the number of possible refinement rules to just six: four for quadrilateral tilings and two for triangle tilings (Figure 4).

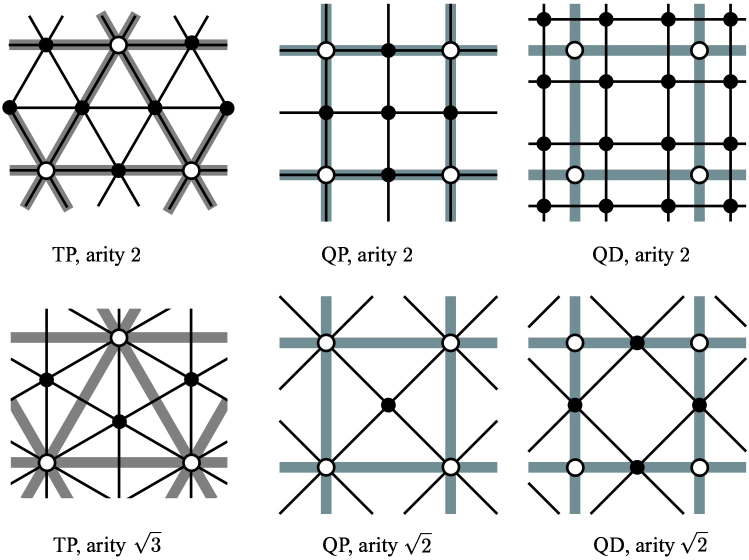
We note that classifications, based on considering various possible transformations of tilings, do not yield an immediate recipe for refinement purely in terms of mesh connectivity; generalization to arbitrary connectivity meshes is not automatic either.

The remaining six refinement rules are uniquely identified by three parameters:

Tiling. The tile can be triangle or quadrilateral.

Vertex mapping. Vertices are mapped to vertices (*primal*) or vertices are mapped to faces (*dual*). Dual triangle refinement are excluded by heuristics 4.

Arity. For triangle tilings can be 2 or $\sqrt{3}$; for quad meshes can be 2 or $\sqrt{2}$.



TP, arity 2

QP, arity 2

QD, arity 2

TP, arity $\sqrt{3}$

QP, arity $\sqrt{2}$

QD, arity $\sqrt{2}$

Fig. 4. Different refinement rules.

Each of the six refinement rules can be easily formulated in terms of mesh connectivity in such a way that the refinement can be applied to an arbitrary polygonal mesh. For ease of understanding, we provide a somewhat informal description. We only specify the set of new vertices and edges, with faces defined implicitly as loops of edges. For primal rules, old vertices are retained, and old edges are discarded. For dual rules, both old vertices and edges are discarded. For each rule, we list how many different types of geometric rules are necessary to construct a subdivision scheme for meshes without boundaries. To handle meshes with boundaries, additional special rules for boundary vertices are necessary.

While triangle-based refinement rules can be applied to any mesh, known geometric rules for such schemes are only formulated for triangle meshes.

Primal triangle rule (TP) of arity 2. This is the rule considered in Section 2: create new vertices for each old edge and split each old edge in two; for each old face connect new vertices inserted on edges of this face sequentially. Two geometric rules are necessary: one to update control points for old vertices (*vertex rule*) and another to compute positions of new control points (*edge rule*).

Primal triangle (TP) rule of arity $\sqrt{3}$. Create a new vertex for each face; connect old vertices with new vertices for each old face containing the old vertex; connect new vertices for adjacent old faces. Two similar geometric rules (vertex and edge) are needed.

Primal quad rule (QP) of arity 2. Create new vertices for each old edge and face; split old edges in two; for each old face, connect corresponding new vertex with new vertices inserted on edges. Three geometric rules are necessary: one for old vertices (*vertex rule*), one for new vertices corresponding to edges (*edge rule*), and one for new vertices corresponding to faces (*face rule*).

Primal quad rule (QP) of arity $\sqrt{2}$. Create a new vertex for each face; connect old vertices to new vertices for all adjacent faces. Two geometric rules are necessary, similar to the TP rules, the edge rule, and the face rule.

Dual quad rule (QD) of arity 2**.** For every face, create new vertices for every corner of the face and connect them into a face; connect new vertices corresponding to the same old vertex from adjacent faces. Only one geometric rule is necessary.

Dual quad rule (QD) of arity $\sqrt{2}$. Add a new vertex for each edge; for each face, connect new vertices on edges sequentially. Only one geometric rule is necessary.

The general property of the triangle rules is that it does not increase the number of non-triangular faces in the mesh. The general property of the quad rules is that they do not increase the number of non-quadrilateral faces. Moreover, both primal quad rules and the $\sqrt{3}$ triangle rule make all faces of a mesh triangular after one refinement step.

Classification. For each refinement rule type, there may be many different subdivision schemes depending on the choice of geometric rules. The geometric rules can be further classified by two characteristics: whether they are *approximating* or *interpolating*, and by their support size. Interpolating schemes do not alter the control points at vertices, inherited from the previous refinement level; approximating schemes do. The distinction between approximating and interpolating for schemes with arity no greater than two makes sense only for primal schemes.

With this criteria in place, we can classify most known schemes; in most cases, only one scheme of a given type is known. The reason for this is that only schemes with small support are practical, and additional symmetry considerations considerably reduce the number of degrees of freedom in coefficients. Maximizing smoothness of resulting surfaces on regular grids further restricts the choices, in most cases yielding a known parametric family of schemes.

The table below lists all schemes known to fit into our classification.

Refinement type	Approximating	Interpolating
TP, arity 2	Loop [45,25,8,46,62]	Butterfly [21,91]
TP, arity $\sqrt{3}$	$\sqrt{3}$,[33], composite $\sqrt{3}$ [57]	interpolatory $\sqrt{3}$,[37]
QP, arity 2	Catmull-Clark [10] iterated [90,76]	Kobbelt [31]
QP, arity $\sqrt{2}$	4-8 [80,79]	interpolating $\sqrt{2}$ [26]
QD, arity 2	Doo-Sabin [16,17], iterated [90]	—
QD, arity $\sqrt{2}$	Midedge [59,23]	—

Polygonal meshes with boundaries. The minimal number of geometric rules, ranging from one to three, is sufficient if we require the rules to be invariant with respect to isomorphisms of mask supports and assume the meshes do not have boundaries.

However, in practice it is not sufficient to consider only this class of meshes: in any practical application, the control mesh may have a boundary. Furthermore, the boundary may not be smooth everywhere: it may consist of several smooth pieces, jointed at *corners*. The definition of meshes with boundary is identical to the polygonal mesh definition in Section 2; the only differences are that an edge can be contained only in one face, and the link of a vertex is a chain of edges, with last vertex not connected to the first.

While a boundary edge or vertex is identified unambiguously, corner vertices on the boundary require tags. It turns out that depending on the type of corner (convex or concave); different rules need to be used, so at least two different tags are needed.

We have already seen that subdivision schemes defined on triangular meshes create new vertices only of valence 6 in the interior. On the boundary, the newly created vertices have valence 4. Similarly, on quadrilateral meshes both primal and dual schemes create only vertices of valence 4 in the interior and 3 on the boundary. Hence, after several subdivision steps, most vertices in a mesh will have one of these valences (6 in the interior, 4 on the boundary for triangular meshes, 4 in the interior, 3 on the boundary for quadrilateral). The vertices with these valences are called *regular*, and vertices of other valences are called *extraordinary*. Similarly, faces with

3 and 4 vertices are called regular for triangle and quadrilateral schemes respectively, and faces with a different number of vertices are called extraordinary.

Next, we consider several examples of subdivision schemes. We start with a detailed description of two schemes that are used in most applications: Loop and Catmull-Clark, which use TP and QP refinement rules of arity 2. Then we consider examples of interpolating schemes (Butterfly), dual schemes (Doo-Sabin) and non-arity 2 schemes (Midedge and 4-8 subdivision).

3.2. *Loop scheme*

The Loop scheme for meshes without boundary was already described in Section 2. The scheme is based on the *three-directional box spline*, which produces C^2-continuous surfaces on the regular meshes. The Loop scheme produces surfaces which are C^2-continuous everywhere except at extraordinary vertices, where they are C^1-continuous. C^1-continuity of this scheme for valences up to 100, including the boundary case, was proved by Schweitzer [73]. The proof for all valences can be found in [87]. In addition to already defined rules for interior vertices, it remains to specify rules for vertices on or near the boundary. The rules we define here were proposed in [8].

A common requirement for rules for boundary vertices is that the control points on level $j+1$ should only depend on boundary control points on level j. In the case of the Loop scheme, for compatibility with the regular case, we use the standard cubic spline rules, both for edge points and vertex points (Figure 5). If a vertex v is tagged as a corner vertex, a trivial interpolating rule is used: $p^{j+1}(v) = p^j(v)$.

Adding these rules formally completes the definition of the scheme for all possible cases; unfortunately, this set of rules is insufficient to produce limit surfaces which are C^1 continuous at the extraordinary boundary vertices or surfaces with concave corners on the boundary. To achieve this, spatial edge rules are applied at edge points *adjacent* to extraordinary boundary vertices.

For edge points, our algorithm consists of two stages, which, if desired, can be merged, but are conceptually easier to understand separately.

The first stage is a single iteration over the mesh during which we apply the vertex rules and compute initial control points for vertices inserted on edges. All rules used at this stage are shown in Figures 3 and Figure 5. The mask support is the same, but the coefficients are modified. The change in coefficient ensures that the surface is C^1 for boundary vertices. However, the scheme still cannot produce concave corners: The surface develops a "flip" at these vertices; the reason for this, informally, is that the invariant configuration defined by subdominant eigenvectors of subdivision matrix in this case does not have a concave corner; rather, it has a convex one.

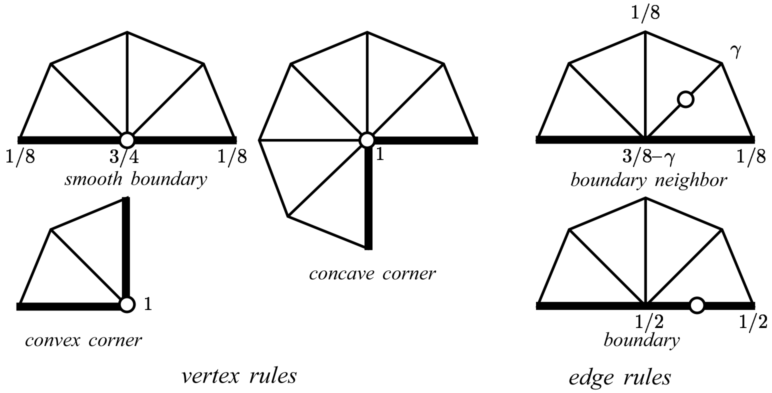


Fig. 5.

The γ is given in terms of parameter θ_k, defined differently for corner and boundary vertices:

$$\gamma\left(\theta_k\right) = 1/2 - 1/4 \cos\theta_k$$

For boundary vertices v not tagged as corners, we use $\theta_k = \pi/k$, where k is the number of polygons adjacent to v. For a vertex v tagged as a convex corner, we use $\theta_k = \alpha/k$, where $\alpha < \pi$, and for concave corner we choose $\alpha > \pi$. The parameter α can be either fixed (e.g. $\pi/2$ for convex and $3\pi/2$ for concave) or can be chosen depending on the angle between the vectors from $p^0(v)$ to adjacent boundary control points adjacent to v.

To ensure the correct behavior at the concave corner vertices, an additional step *flatness modification* is required which is defined as follows.

Flatness modification. To avoid the flip problem described above, one needs to ensure that the eigenvalues corresponding to a pair of "correct" eigenvectors, forming a concave corner, are subdominant. The following simple technique proposed in [8] achieves this. We introduce a *flatness parameter s* and modify the subdivision rule to scale all eigenvalues except λ_0 and $\lambda = \lambda_1 = \lambda_2$, corresponding to the desired eigenvectors, by factor

$1-s$. The vector of control points p after subdivision in a neighborhood of a point is modified as follows:

$$p^{\mathrm{new}} = (1-s)\,p + s\left(\mathbf{a}_0 x^0 + \mathbf{a}_1 x^1 + \mathbf{a}_2 x^2\right),$$

where, as before, $\mathbf{a}_i = (l^i \cdot p)$, and $0 \le s \le 1$. Geometrically, the modified rule blends between control point positions before the flatness modification and certain points in the tangent plane, which are typically close to the projection of the original control point. The limit position $\mathbf{a}_0$ of the center vertex remains unchanged.

The flatness modification is always applied at concave corner vertices; the default values for the flatness parameter is $s = 1 - (1/4)/\lambda_3$, where $\lambda_3 = (1/4)(\cos\pi/k) - \cos(\theta_k)) + 1/2$ (the largest eigenvalue $\neq 1$ of the subdivision matrix before the modification). The modification ensures that the surface is C^1 in this case. In other cases, s can be taken to be 0 by default.

The formulas for limit positions and tangents for all possible cases can be found in [8].

3.3. *Catmull-Clark scheme*

The Catmull-Clark scheme [10] probably is the most widely used subdivision scheme. One of the reasons is it extends tensor-product bicubic B-spline surfaces, the most commonly used type of spline surfaces. This scheme uses the QP refinement rule with arity 2. It produces surfaces that are C^2 everywhere, except at extraordinary vertices, where they are C^1. The tangent plane continuity of the scheme was analyzed in [5], and C^1-continuity in [60].

The masks are shown in Figure 6; for interior vertices, there are three types of masks: for new vertices inserted at edges and faces and for update of control points at old vertices.

If $k = 4$, the masks reduce to subdivision masks for bicubic B-splines. Similar to the Loop scheme, cubic spline rules are applied at the boundary, and at the corner boundary vertices, the trivial interpolating rule is used. Again, just as is the case for the Loop scheme, the minimal set of rules results in surfaces which lack smoothness at extraordinary boundary vertices. A similar technique is used for Catmull-Clark, with parameter γ computed as

$$\gamma(\theta_k) = 3/8 - 1/4\cos\theta_k.$$

The parameter θ_k is defined exactly in the same way as for the Loop scheme.

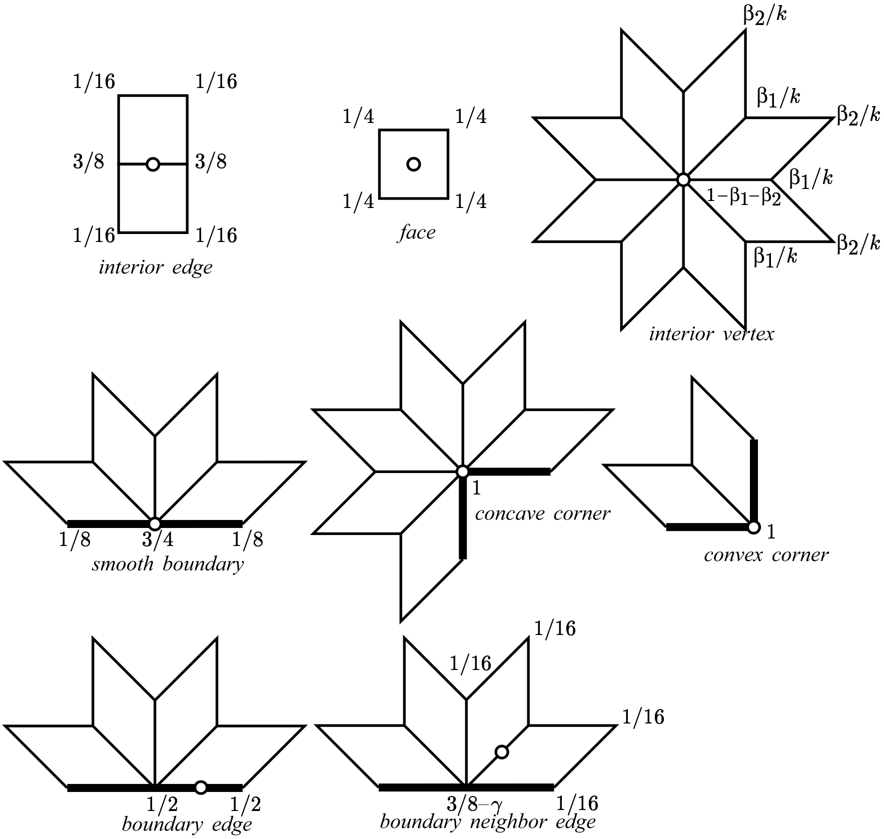


Fig. 6. Catmull-Clark subdivision. Catmull and Clark suggest the following coefficients for rules at extraordinary vertices: $\beta_1 = \frac{3}{2k}$ and $\beta_2 = \frac{1}{4k}$.

Finally, a similar extra step is used to ensure correct behaviour at concave corners:

$$p^{\text{new}} = (1-s)\,p + s\left(\mathbf{a}_0 x^0 + \mathbf{a}_1 x^1 + \mathbf{a}_2 x^2\right).$$

The limit position and tangent vector coefficients are listed in [8].

The geometric rules of the Catmull-Clark scheme are defined above for meshes with quadrilateral faces. Arbitrary polygonal meshes can be reduced to a quadrilateral mesh using a more general form of Catmull-Clark rules [10]:

- a face control point for an n-gon is computed as the average of the corners of the polygon;

- an edge control point is the average of the endpoints of the edge and newly computed face control points of adjacent faces;
- the vertex rule can be chosen in different ways; the original formula is

  $$p^{j+1}(v) = \frac{k-2}{k} p^j(v) + \frac{1}{k^2} \sum_{i=0}^{k-1} p^j(v_i) + \frac{1}{k^2} \sum_{i=0}^{k-1} p^{j+1}(v_i^f)$$

  where v_i are the vertices adjacent to v on level j, and v_i^f are face vertices on level $j+1$ corresponding to faces adjacent to v.

4. Modified Butterfly Scheme

The Butterfly scheme was proposed in [21]. Although the original Butterfly scheme is defined for arbitrary triangular meshes, the limit surface is not C^1-continuous at extraordinary points of valence $k = 3$ and $k > 7$ [87]. The scheme is C^1 on regular meshes.

Unlike approximating schemes based on splines, this scheme does not produce piecewise polynomial surfaces in the limit. In [91] a modification of the Butterfly scheme was proposed, which guarantees that the scheme produces C^1-continuous surfaces for arbitrary meshes as proved in [87]. The scheme is known to be C^1 but not C^2 on regular meshes. The masks for the scheme are shown in Figure 7.

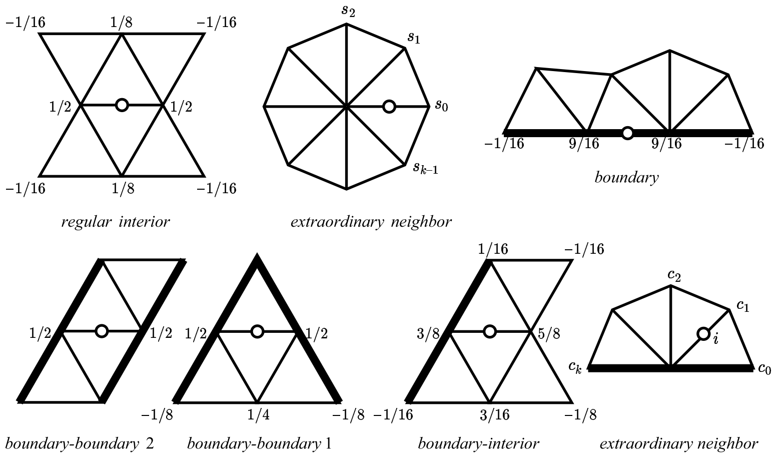


Fig. 7. Modified Butterfly subdivision. The coefficients s_i are $\frac{1}{k}\left(\frac{1}{4} + \cos\frac{2i\pi}{k} + \frac{1}{2}\cos\frac{4i\pi}{k}\right)$ for $k > 5$. For $k = 3$, $s_0 = \frac{5}{12}$, $s_{1,2} = -\frac{1}{12}$; for $k = 4$, $s_0 = \frac{3}{8}$, $s_2 = -\frac{1}{8}$, $s_{1,3} = 0$.

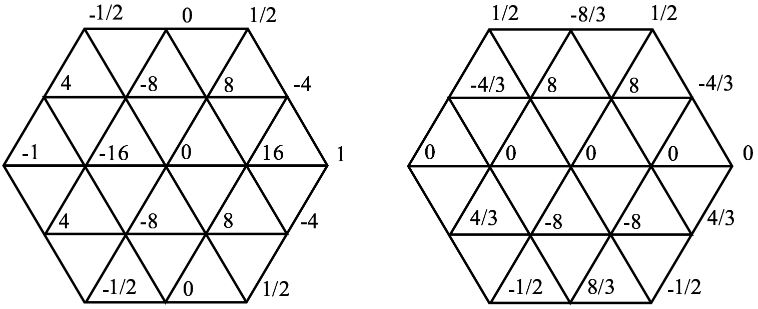


Fig. 8. Tangent masks for regular vertices (Butterfly scheme).

The tangent vectors at extraordinary interior vertices can be computed using the same rules as for the Loop scheme. For regular vertices, the formulas are more complex: in this case, we have to use control points in a 2-neighborhood of a vertex. The masks are shown in Figure 8.

Because the scheme is interpolating, no formulas are needed to compute the limit positions: all control points are on the surface. On the boundary, the four point subdivision scheme is used [20]. To achieve C^1-continuity on the boundary, special coefficients have to be used.

Boundary rules. The rules extending the Butterfly scheme to meshes with boundary are somewhat more complex, because the stencil of the Butterfly scheme is relatively large. A complete set of rules for a mesh with boundary (up to head-tail permutations), includes 7 types of rules: regular interior, extraordinary interior, regular interior-boundary, regular boundary-boundary 1, regular boundary-boundary 2, boundary, and extraordinary boundary neighbor; see Figures 7. To put it all into a system, the main cases can be classified by the types of head and tail vertices of the edge on which we add a new vertex. The following table shows how the type of rule to be applied

Head	**Tail**	**Rule**
regular interior	regular interior	standard rule
regular interior	regular crease	regular interior-crease
regular crease	regular crease	regular crease-crease 1 or 2
extraordinary interior	extraordinary interior	average two extraordinary rules
extraordinary interior	extraordinary crease	same
extraordinary crease	extraordinary crease	same
regular interior	extraordinary interior	interior extraordinary
regular interior	extraordinary crease	crease extraordinary
extraordinary interior	regular crease	interior extraordinary
regular crease	extraordinary crease	crease extraordinary

for computing a *non-boundary* vertex is determined from the valence of the adjacent vertices, and whether they are on the boundary or not. The only case when additional information is necessary, is when both neighbors are regular crease vertices.

The extraordinary crease rule (Figure 7) uses coefficients c_{ij}, $j = 0 \ldots k$, to compute the vertex number i in the ring, when counted from the boundary. Let $\theta_k = \pi/k$. The following formulas define c_{ij} :

$$c_0 = 1 - \frac{1}{k}\left(\frac{\sin\theta_k \sin i\theta_k}{1-\cos\theta_k}\right)$$

$$c_{i0} = -c_{ik} = \frac{1}{4}\cos i\theta_k - \frac{1}{4k}\left(\frac{\sin 2\theta_k \sin 2\theta_k i}{\cos\theta_k - \cos 2\theta_k}\right)$$

$$c_{ij} = \frac{1}{k}\left(\sin i\theta_k \sin j\theta_k + \frac{1}{2}\sin 2i\theta_k \sin 2j\theta_k\right)$$

4.1. ***Doo-Sabin scheme***

The Doo-Sabin subdivision is quite simple conceptually: a single mask is sufficient to define the scheme. Special rules are required only for the boundaries, where the limit curve is a quadratic spline. It was observed by Doo that this can also be achieved by replicating the boundary edge, i.e., creating a quadrilateral with two coinciding pairs of vertices. Nasri [52] describes other ways of defining rules for boundaries. The rules for the Doo-Sabin scheme are shown in Figure 9. C^1-continuity for schemes similar to the Doo-Sabin schemes was analyzed in [60].

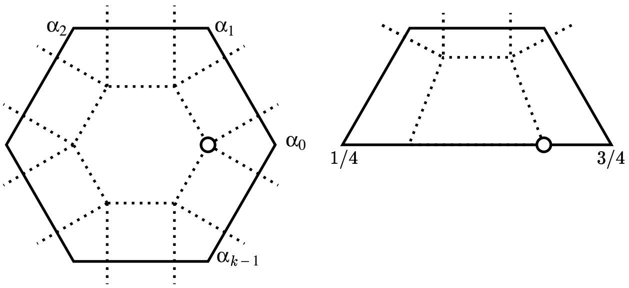


Fig. 9. The Doo-Sabin subdivision. The coefficients are defined by the formulas $\alpha_0 = 1/4 + 5/4k$ and $\alpha_i = (3 + 2\cos(2i\pi/k))/4k$, for $i = 1 \ldots k-1$.

4.2. *Midedge scheme and other non-integer arity schemes*

A scheme described in [59] is an arity $\sqrt{2}$ QD scheme; two steps of refinement of this type result in Doo-Sabin type scheme.

The rules for the simplest version of this scheme are very straightforward: the point inserted on an edge is the average of the endpoints. While the limit surface is smooth for this rule, the quality of the surface is not good for extraordinary faces; the rules can be modified to improve surface quality.

An example of a QP scheme of arity $\sqrt{2}$ is the 4-8 scheme [80,79]. While originally defined in terms of 4-8 refinement, it can be easily reinterpreted in terms of regular quadrilateral grid refinement as shown in Figure 10.

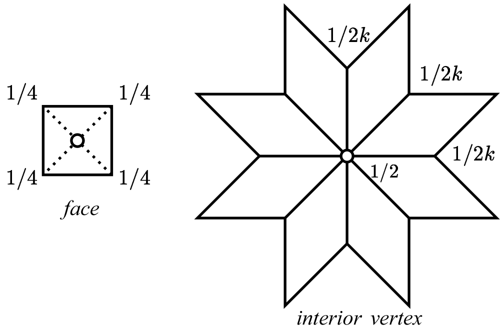


Fig. 10. The 4-8 subdivision scheme rules refinement. As the edges are not refined, only face and vertex rules are necessary.

It should be noted that for quadrilateral schemes of non-integer arity; there appears to be no natural treatment for the boundaries: as each quad for the refined mesh has vertices from two quads sharing an edge, it is impossible to construct quads in the same way on the boundary. One needs to introduce special refinement rules on the boundary and corresponding special geometric rules. A set of such rules is described in [80]. The rules are quite complex (six different rules are needed), in contrast to the rules for interior vertices.

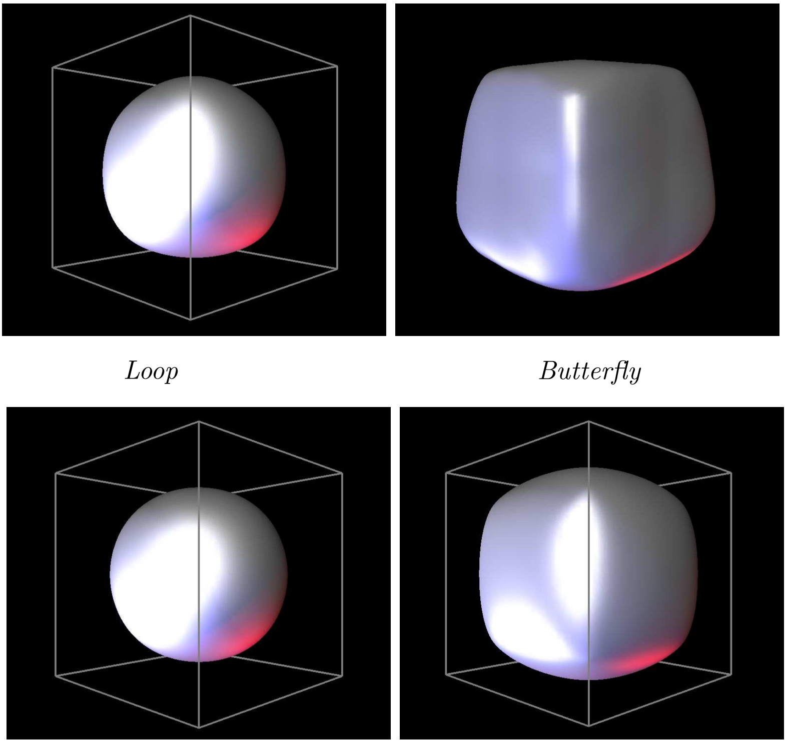
On regular grids this scheme produces surfaces of high smoothness (C^4) despite its small support, but at extraordinary vertices, it is still only C^1.

The first TP scheme of arity $\sqrt{3}$ was described in [33]; other schemes were considered in [57].

4.3. *Comparison*

We conclude our survey of subdivision schemes with some comparisons. For sufficiently smooth and fine control meshes, the results for most common schemes are indistinguishable visually. We use relatively simple meshes to demonstrate the differences in clear form; for most meshes used in applications, the differences are less apparent. In our comparison, we consider Loop, Catmull-Clark, Modified Butterfly and Doo-Sabin subdivision.

Figure 11 shows the surfaces obtained by subdividing a cube. Loop and Catmull-Clark subdivision produce surfaces of higher visual quality, as these schemes reduce to C^2 splines on a regular mesh. As all faces of the cube are quads, Catmull-Clark yields the nicest surface; the surface generated by the Loop scheme is more asymmetric because the cube had to be triangulated



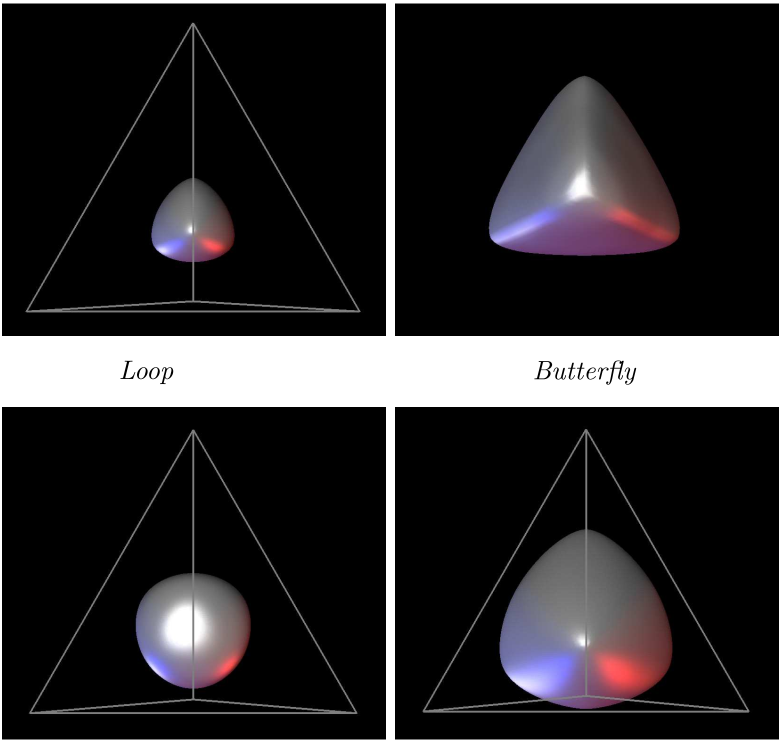
Loop

Butterfly

Catmull-Clark

Doo-Sabin

Fig. 11. Results of applying various subdivision schemes to the cube. For triangular schemes (Loop and Butterfly) the cube was triangulated first.



Loop

Butterfly

Catmull-Clark

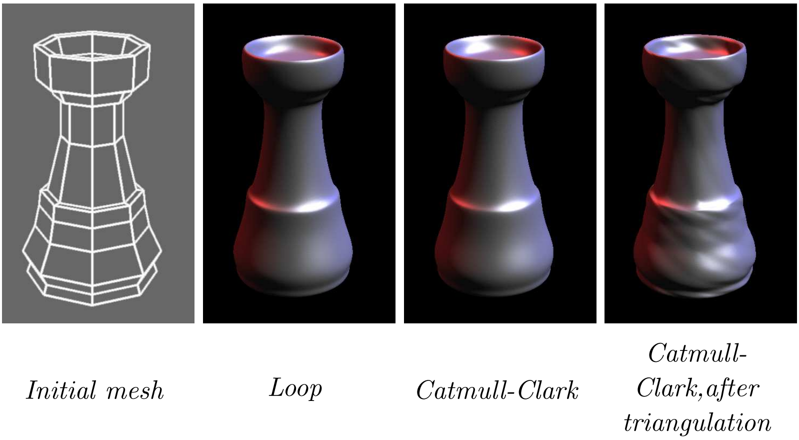
Doo-Sabin

Fig. 12. Results of applying various subdivision schemes to a tetrahedron.

before the scheme is applied. At the same time, Doo-Sabin and Modified Butterfly reproduce the shape of the cube more closely. The surface quality is worst for the Modified Butterfly scheme, which interpolates the original mesh. We observe that there is a tradeoff between interpolation and surface quality: the closer the surface is to interpolating, the lower the surface quality.

Figure 12 shows the results of subdividing a tetrahedron. Similar observations hold in this case. In addition, we observe extreme shrinking for the Loop and Catmull-Clark subdivision schemes.

Overall, Loop and Catmull-Clark appear to be the best choices for most applications, which do not require exact interpolation of the initial mesh. The Catmull-Clark scheme is most appropriate for meshes with a significant fraction of quadrilateral faces. It might not perform well on certain types



Initial mesh

Loop

Catmull-Clark

Catmull-Clark, after triangulation

Fig. 13. Applying Loop and Catmull-Clark subdivision schemes to a model of a chess rook. The initial mesh is shown on the left. Before the Loop scheme was applied, the mesh was triangulated. Catmull-Clark was applied to the original quadrilateral model and to the triangulated model; note the substantial difference in surface quality.

of meshes, most notably triangular meshes obtained by triangulation of a quadrilateral mesh (see Figure 13). The Loop scheme performs reasonably well on any triangular mesh, thus, when triangulation is not objectionable, this scheme might be preferable.

More in-depth studies of subdivision surface behavior focusing on curvature can be found in [68,61,29]. Ways to improve surface appearance using coefficient tuning were explored in [6].

5. Smoothness of Subdivision Surfaces

In this section we review the theory of smoothness of surfaces generated using stationary subdivision. Smoothness is the focus of most of the work in theory of subdivision. The standard goal is to establish conditions on masks of subdivision schemes that ensure that the limit surfaces, for almost all configurations of control points, are in a smoothness class. Most commonly, the classes C^r, for integer values of r are considered.

In the regular case, powerful analysis tools exist. (see e.g. a recent survey [19] or the book [11] as well as [28] for further references). In most cases, subdivision schemes for surfaces are constructed by generalizing relatively simple schemes for regular grids, for which smoothness analysis is relatively straightforward.

Due to locality of subdivision rules, this ensures surfaces are smooth away from isolated points, corresponding to vertices or face centers of the initial meshes. To complete the analysis for arbitrary meshes, one needs to analyze behaviour near such points; in this section we concentrate on this topic.

To be able to formulate the criteria for surface smoothness, we precisely define the limit subdivision surfaces and review tangent plane continuity and C^r-continuous surfaces.

5.1. *C^r-continuity and tangent plane continuity*

There are many different equivalent or nearly equivalent ways to define C^r-surfaces for integer r. A standard approach in differential geometry is to define C^r manifolds, and then define C^r surfaces in $\mathbf{R}^n$ as C^r-continuous *immersions* or *embeddings* of C^r manifolds. However, this approach is not the most convenient for our purposes, as no *a priori* smooth structure exists on the domain of subdivision surfaces. Thus, we take a somewhat different but equivalent approach. We do not require a smooth structure and say that a surface defined on a domain, for which only topological structure exists, is C^r if there is a C^r-continuous local reparameterization for a neighborhood of any point. More formally, we use the following definition.

Definition 1: A surface $f : M \to \mathbf{R}^n$, where M is a topological 2D manifold, is C^r**-continuous**, for $r \geq 1$, if for every point $x \in M$ there exists an open neighborhood U_x in M of x, and a regular parameterization $\pi : D \to f(U_x)$ of $f(U_x)$ over an open unit disk D in the plane, A **regular parameterization** π is one that is r-times continuously differentiable, one-to-one, and has a Jacobi matrix of maximum rank, i.e. if (s, t) is a choice of coordinates on D $\partial_s \pi$ and $\partial_t \pi$ for any choice of coordinates on D are independent.

We call a subdivision scheme C^r continuous if for any complex K and almost any choice of control points p for vertices of this complex, resulting limit surfaces are C^r-continuous. In practice, however, it is difficult to prove this for arbitrary complexes, and additional restrictions have to be imposed.

The condition that the Jacobi matrix of p has maximum rank is necessary to make sure that there no degeneracies, i.e., f represents a surface, not a curve or point.

In our constructions, it is useful to consider a weaker definition of surface smoothness at a point. This definition captures the intuitive idea that the tangent plane to a surface changes continuously, and is applicable only for an isolated point, i.e. we assume that the surface is C^r-continuous everywhere excluding a point. We first define a tangent plane continuous surface in $\mathbf{R}^3$. Note that if the surface is C^1-continuous in $\mathbf{R}^3$ in a neighborhood of a point, there is a well-defined normal at that point given for a choice of coordinates (s,t) by $\partial_s\pi \times \partial_t\pi$.

Definition 2: A surface $f : M \to \mathbf{R}^3$ is **tangent plane continuous** at $x \in M$ if and only if it is C^1-continuous in a neighborhood of x, and there exists a limit of normals at x.

An example of a surface which is tangent plane continuous but not C^1-continuous is $(x = s^2 - t^2, y = 2st, z = s^3)$.

We will also need the definition of tangent plane continuity in higher dimensions; for $n > 3$, the appropriate generalization of the cross product is the exterior (wedge) product, $\mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}^{n(n-1)/2}$; for two vectors v, w, their product $v \wedge w$ has components $v_i w_j - v_j w_i$, $0 \le i < j \le n$. The exterior product is linear in each argument and antisymmetric ($v \wedge w = -w \wedge v$). From antisymmetry, it follows that $v \wedge v = 0$. For $n = 3$, the exterior product is identical to the cross product. The exterior product $v \wedge w$ defines a plane in n dimensions spanned by vectors v and w just as normal $v \times w$ defines the plane in 3D. In higher dimensions, the definition of tangent plane continuity is identical to 3D, with exterior product $\partial_s \wedge \partial_t$ considered instead of the normal.

The following fact can be easily proved: if a surface is tangent plane continuous at a point and the projection of the surface onto the tangent plane at that point is one-to-one for a neighborhood of the point, the surface is C^1.

The definition of tangent plane continuity for a subdivision scheme is similar to the definition of C^r-continuity.

5.2. *Universal surfaces*

We present an approach to establishing smoothness criteria for subdivision schemes described in [88]. We do not derive the necessary and sufficient conditions in full generality, as required algebraic machinery is relatively complicated and obscures the main ideas. Instead, we derive conditions similar to Reif's originally proposed sufficient condition [63]. We use the more general approach based on the universal surfaces over Reif's original derivation since in author's view it provides better geometric intuition for tangent plane continuity and C^1 continuity. Most statements are presented without proof. For more complete analysis and proofs, we refer the reader to [65,86,88].

It is intuitively clear that to verify that a subdivision scheme with finitely supported masks produces smooth surfaces for almost all configurations of control points, it is sufficient to consider behavior of a part of the surface on a 1-neighborhood of an extraordinary vertex v $|N_1(v)|$. We further assume that the control mesh for $|N_1(v)|$ contains a single extraordinary vertex and is an invariant neighborhood. This is, in fact, a limiting assumption; however, all known analysis techniques rely on this assumption, as verification of smoothness of subdivision schemes in a more general setting so far is not possible. This problem is discussed in greater detail in [88].

In this restricted setting, we can regard a regular k-gon U centered at zero in $\mathbf{R}^2$, as the domain of the patch of the subdivision surface in which we are interested. Let S be the subdivision matrix, and p^j vectors of control points for U at subdivision levels j, $p^{j+1} = Sp^j$. Let N be the number of points in p. An important observation following from the construction of the limit subdivision surface is that $p^1 = Sp$ is the vector of control points for the scaled domain $(1/2)U$, and in general, $p^j = S^j p^0$ is the vector of control points for $(1/2^j)U$; in other words, the limit function $f[p]$ evaluated on $(1/2)U$ satisfies

$$f[p](y/2) = f[S^j p](y) \tag{5}$$

Consider a basis $e_1, \ldots e_N$; then $p = \sum_i p_i e_i$. By linearity of subdivision, we can write $f[p] = f[\sum_i p_i e_i] = \sum_i p_i f[e_i]$. We introduce the map $\psi : U \to \mathbf{R}^N$, defined as $f = (f[e_1], f[e_2], \ldots f[e_N])$. This surface (*the universal surface*)defined by this map is defined uniquely up to a nonsingular linear transformation.

For any vector of control points p, we can regard the subdivision surface $f[p]$ as a linear map of the universal surface to three-dimensional space give by

$$f[p](y) = (p \cdot \psi(y))$$

It immediately follows from (5) that ψ satisfies

$$\psi(y/2) = S^T \psi(y) \tag{6}$$

Furthermore, we can verify by direct computation that the normal to the subdivision surface $f[p](y)$ at points y where it is f is differentiable can be computed as

$$\partial_1 f[p] \wedge \partial_2 f[p] = N(y) = ((p^y \wedge p^z) \cdot w(y), (p^z \wedge p^x) \cdot w(y), (p^x \wedge p^y) \cdot w(y)) \tag{7}$$

where $w(y) = \partial_1 \psi(y) \wedge \partial_2 \psi(y)$, i.e. the analog of the normal for the universal surface. We also note that f is differentiable everywhere on U except at edges of triangles of U. Furthermore, one-sided derivative limits exist at edges, excluding the center of U, i.e. zero. One can show that using one-sided limits of derivatives on either side of the edge yields the same vector $w(y)$, so it is defined everywhere.

This surface has the following important property.

Theorem 3: A subdivision scheme is tangent plane continuous (C^r) at vertices of a given valence if and only if the universal surface for this valence is tangent plane continuous, assuming that the universal surface is C^1-continuous away from zero. The universal surface is C^r-continuous if and only if the subdivision scheme is C^r continuous.

This theorem allows us to replace analysis of all possible surfaces generated using a subdivision scheme with analysis of a single surface in higher-dimensional space for each valence. The assumption of the theorem about C^1-continuity away from zero typically follows from the analysis of the regular case and *the characteristic map* as explained below.

To analyze whether the universal surface is tangent plane continuous, we need to look at the behavior of the vectors $w(y)$ (the generalized normals) as $y \to 0$. For any linear transform A, $Aw \wedge Av = (\Lambda A)(w \wedge v)$ defines a natural extension of A to the space of exterior products. Thus, taking derivatives and wedge products, we obtain

$$w(y/2) = \partial_1 \psi(y/2) \wedge \partial_2 \psi(y/2) = 4(\Lambda S^T) \partial_1 \psi(y) \wedge \partial_2 \psi(y) = 4(\Lambda S^T) w(y) \tag{8}$$

i.e. the vector $w(y)$ satisfies a scaling relation, but with a different matrix.

As a result, our problem is reduced to the following: under which conditions does the direction of $A^j w(y)$, where $A = 4\Lambda S^T$, converge to a unique limit for $j \to \infty$ and for any choice of y?

5.3. *Sufficient smoothness criteria*

So far, our discussion has been completely general. Without any assumptions on the matrix S, the conditions for convergence to a unique limit can be quite complex and require analysis of the Jordan normal form of the subdivision matrix. The main ideas can be easily understood if we consider the special case when S satisfies the conditions of Section 2: the matrix has a basis of eigenvectors, λ_1 and λ_2 are real positive, $\lambda_0 = 1 > \lambda_1 \geq \lambda_2 > |\lambda_3|$, if the eigenvalues are ordered by magnitude, and each is repeated once for each of its eigenvectors.

In the case of such matrices, the matrix ΛS also has a simple structure. First, we observe that if x_i and x_j are independent eigenvectors of S, with eigenvalues λ_i and λ_j, then $\Lambda S(x_i \wedge x_j) = Sx_i \wedge Sx_j = \lambda_i \lambda_j x_i \wedge x_j$, i.e. $x_i \wedge x_j$ is an eigenvector with eigenvalue $\lambda_i \lambda_j$. There are $N(N-1)/2$ such eigenvectors, and these eigenvectors are independent. We conclude that ΛS also has a complete system of eigenvectors, with eigenvalues equal to $\lambda_i \lambda_j$, with $i < j$.

This observation allows us to understand the behavior of $A^j w(y)$. Suppose $w(y) = \sum_i \alpha_i x_i$ where x_i are eigenvectors of A. Then, the direction of $A^j w(y)$ converges to the direction of x_i, where x_i is the eigenvector with largest eigenvalue such that $\alpha_i \neq 0$.

We observe that we can define $\psi = \sum c_i f_i$, where $f_i = f[x_i]$ is the eigenbasis function corresponding to eigenvalue λ_i; in particular, by affine invariance, f_0 is a constant.

$$w(y) = \sum_{i<j} (c_i \wedge c_j)(\partial_1 f_i \partial_2 f_j - \partial_2 f_i \partial_1 f_j) = \sum_{i<j} (c_i \wedge c_j) J[f_i, f_j]$$

where $J[f_i, f_j]$ denotes the Jacobian of two functions.

We note that the terms corresponding to $i = 0$ vanish because f_0 is a constant. Thus, the largest eigenvalue which may have a nonzero term in this decomposition is $\lambda_1 \lambda_2$.

If we assume that for any y, $J[f_i, f_j](y) \neq 0$, we see that the limit direction of $A^j w(y) = w(y/2^j)$ is always $c_1 \wedge c_2$, i.e. all these sequences converge to the same limit. With more careful analysis, one can easily establish that the limit is the same for any sequence $w(y_j)$, with $y_j \to 0$.

We obtain the following *sufficient* condition for tangent plane continuity:

Theorem 4: Suppose for a valence k, the subdivision matrix is non-defective and has eigenvalues satisfying $\lambda_0 = 1 > \lambda_1 \geq \lambda_2 > |\lambda_3|$, when ordered in non-increasing order, each eigenvalue repeated according to its multiplicity. Suppose the eigenbasis functions corresponding to eigenvalues λ_1 and λ_2 satisfy $J[f_1, f_2] \neq 0$ everywhere on the regular k-gon $U \setminus \{0\}$. Then, the scheme produces tangent plane continuous surfaces on U for almost any choice of control points.

The pair of functions (f_1, f_2) defines a map $U \to \mathbf{R}^2$. This planar map is called the *characteristic map*[a].

This condition is not necessary, even given the assumptions on the scheme: e.g. the Jacobian of f_1 and f_2 can be zero everywhere, but the scheme can still be tangent plane continuous if e.g. the Jacobian of f_1 and f_3 does not vanish. Theorem 4 is a weaker form of Reif's criterion; note that we do not obtain C^1 continuity, only tangent plane continuity. However, we note that the stronger C^1-continuity criterion immediately follows from combining Theorem 4 with the observation from Section 5.1 that C^1 continuity is equivalent to tangent plane continuity and injectivity of projection to the tangent plane.

We observe that in the coordinate system with basis vectors c_i, the projection of the universal surface to the tangent plane is equivalent to simply discarding all components except f_1 and f_2; this projection is one-to-one, if the map $(f_1, f_2) : U \to \mathbf{R}^2$ is one-to-one, i.e. the characteristic map is injective. This yields the following criterion.

Corollary 5: *(Reif's criterion) If the assumptions of Theorem 4 are satisfied, and in addition the characteristic map is injective, the scheme produces C^1-continuous surfaces on U for almost any choice of control points p.*

Higher-order smoothness. The general conditions for higher order smoothness have quite elaborate form and are beyond the scope of this tutorial. We only state a necessary and sufficient condition for C^2-continuity, which are of greatest practical relevance, for a limited class of schemes since the conditions have simple and intuitive form:

Proposition 6: *Suppose a scheme satisfies conditions of Corollary 5 and has equal subdominant eigenvalues $\lambda = \lambda_1 = \lambda_2$. Then the scheme produces C^2 continuous surfaces if and only if for any eigenvalue $\mu \neq \lambda$, $\mu \neq 1$, either $|\mu| < \lambda^2$ or $\mu = \lambda_2$, and the corresponding eigenbasis function is a homogeneous quadratic function of f_1 and f_2.*

This condition shows a serious limitation of stationary subdivision: the simplest approach to constructing C^2 schemes is to ensure that all non-subdominant eigenvalues are sufficiently small. This can be easily achieved

[a] Reif's original definition is somewhat different: only the restriction of (f_1, f_2) to an annular region around zero is included.

by manipulation of coefficients, as was shown in [8,62], but results in surfaces, which have zero quadratic approximants at extraordinary vertices, i.e. zero curvature. To obtain non-zero curvature, we need to satisfy a much more difficult condition on the eigenbasis functions. In fact, it was demonstrated in [64] that this is impossible to achieve for schemes based on low degree splines.

6. Approximation Properties of Subdivision Surfaces

While smoothness of subdivision surfaces with arbitrary control meshes has received a lot of attention, much less is known about approximation properties: to the best of our knowledge, there is a single published work on the topic [3]. Given that subdivision bases for refined grids coincide with spline surfaces almost everywhere, one would expect similar approximation behavior. However, available estimates do not fully confirm this. At the same time, subdivision surfaces are used by many authors as a practical approximation tool [48,42,70,77,50,2] with good results, which highlights the need for more thorough theoretical exploration of this aspect of subdivision.

In this section we review the main concepts of approximation of surfaces and state the estimates obtained in [3] as well as some results of [89].

6.1. *Functional spaces on surfaces*

A typical form of the approximation estimates for finite element and splines spaces is

$$\|g - \tilde{g}\|_{H^s(\Omega)} < Ch^{t-s}\|g\|_{H^t(\Omega)},$$

where $g \in H^t$ is the approximated function, $\tilde{g}$ is the best approximation (the closest point from the approximating space), h is a parameter characterizing the approximation space (e.g. element size for finite elements, or support size of individual basis functions), and $H^s(\Omega)$ and $H^t(\Omega)$ are Sobolev spaces on the domain Ω.

Our goal is to derive similar estimates for subdivision surfaces. The task is complicated by the fact that the domains on which subdivision bases are defined (polygonal complexes) do not have an intrinsic smoothness structure, and it is impossible to define Sobolev spaces on these domains without introducing such structure.

On the other hand, one can observe that subdivision itself can be used to introduce a smoothness structure on the domain using the characteristic maps. Before we explain the construction, we review the needed general concepts: $C^{r,1}$ manifolds and Sobolev spaces on these manifolds in the context of polygonal complexes.

First, we recall the definition of spaces $H^s(\Omega)$ for an open domain Ω in $\mathbf{R}^n$. Consider a function f in $C_0^\infty(\Omega)$, the space of compactly supported smooth functions on Ω. For an integer s, define the seminorm $|f|_{H^s(\Omega)}$ as the L_p norm of the s-th differential of f on Ω (recall that the s-th differential is a multilinear form in s variables with coefficients equal to partial derivatives of total order s). The norm $\|f\|_{H^s(\Omega)}$ is defined as $\|f\|_{L_p(\Omega)} + |f|_{H^s(\Omega)}$.

A $C^{r,1}$ 2D manifold structure on a subset M of Euclidean space is an *atlas*, which is a collection of *charts* (χ_i, Ω_i), $\chi_i : \Omega_i \to M$, and Ω_i is an open domain in $\mathbf{R}^2$. Charts satisfy several conditions: (1) the union of $\chi_i(\Omega_i)$ is M; (2) the *transition maps* $\chi_j^{-1} \circ \chi_i$ are of smoothness class $C^{r,1}$, i.e. of r-times differentiable functions with Lipshitz r-th derivatives. We need to consider $C^{r,1}$ smoothness structures, rather than simply C^r because they allow us to construct a broader range of functional spaces on manifolds with smoothness structure defined by subdivision.

Let ρ_i be a partition of unity subordinated to the atlas χ_i, that is, the support of each ρ_i is contained in the range of χ_i (note that the support of a function is defined to be the closure of the set where the function is not zero, and therefore, the distance from $\operatorname{supp} \rho_i$ to $\partial\Omega$ is positive.) For a $C^{r,1}$ and $s \leq r+1$ function $f : M \to \mathbf{R}$ define the norm

$$\|f\|_{H^s(\Omega_i)} = \sum_i \|(\rho_i f) \circ \chi_i\|_{H^s \Omega_i}$$

The space $H^s(M)$ is the completion of $C^{k,1}(M)$ with respect to this norm. It is straightforward to show that the norms defined with respect to different atlases or partitions of unity M are equivalent; thus, the definition of the space $H^s(M)$ does not depend on the atlas or the partition of unity. The fact that $f \circ g$ is in H^s if f is in H^s and g is in $C^{s-1,1}$ is necessary to prove invariance [51].

In this definition, the norm is not invariant with respect to the change of atlas: while the spaces stay the same, the norms may change.

These definitions allow us to formulate approximation estimates for bases defined on manifolds. It remains to define a sufficiently smooth structure on the domain of subdivision surfaces.

6.2. *Manifold structure defined by subdivision*

As in the previous section, we restrict our attention to TP schemes of arity 2, i.e. schemes similar to the Loop scheme.

Suppose for regular grids a subdivision scheme yields functions which are $C^{r,1}$, for example it is based on B-splines of degree $r+1$. We note that these splines reproduce polynomials of degree $r+1$, and therefore, have approximation order $r+2$ for functions from the space H^{r+2}.

We also assume that the characteristic map is regular, in the sense defined in the previous section, and one-to-one. For each vertex v of a complex K, consider the inverse of the composition of a piecewise linear map from $|N_1(K)|$ to the regular k-gon U_k, with the characteristic map $\Phi_k : U_k \to \mathbf{R}^2$. We denote this map χ_v. We use the interior of images $\Phi_k(U_k)$ as the domains of the charts, and χ_v as the chart maps for out atlas.

One can easily show that for any two adjacent vertices v and w for any interior point of $|N_1(v)| \cap |N_2(w)|$, the composition $\chi_v^{-1} \circ \chi_w$ is $C^{r,1}$, i.e. the structure defined by these charts is $C^{r,1}$.

We conclude that for a scheme producing $C^{r,1}$-continuous surfaces for regular control grid and with regular and one-to-one characteristic maps, we can define smoothness spaces up to order $k+1$ on arbitrary meshes. This result is somewhat unsatisfactory as we cannot consider functions of higher smoothness $k+2$, for which the scheme has the best approximation rate in the regular case.

Now we can state the result obtained in [3].

Theorem 7: Consider bases defined by the Loop subdivision scheme on complexes K^0, K^1, $\ldots K^j$ $\ldots$, obtained by quadrisection refinement of the initial complex $K^0 = K$. Then, the $C^{2,1}$ manifold structure and Sobolev spaces $H^t(|K|)$ for $t \leq 3$ are defined on $|K|$ as described above, and the best approximation $\tilde{f}$ by subdivision basis functions of a function $f \in H^t(|K|)$ satisfies

$$\|f - \tilde{f}\|_{H^s(|K|)} < C\lambda_{max}^{t-s}\|f\|_{H^t(|K|)}$$

for any $s \leq 2$, where λ_{max} is the maximal subdominant eigenvalue for all valences of vertices in K.

Comparing with what is known for quartic box splines, on which Loop scheme is based, we see that this statement is limiting in several ways. First, the maximal exponent is 3 rather than 4; this is due to the limited smoothness of the chosen manifold structure on $|K|$. The order of approximation is further reduced by having to use λ_{max}, which can be as high as 5/8 instead of 1/2, as the scale parameter.

Finally, the norm on the left-hand side can be at most H^2. This is due to the fact that the basis functions are C^1, and therefore, are in H^2 but not higher smoothness spaces.

Given that the basis produced by subdivision almost everywhere coincides with the basis in the regular case, one expects that better estimates should be possible. Indeed, one can show that by choosing a different smoothness structure on $|K|$, one can obtain the following estimate [89]

$$\|f - \tilde{f}\|_{L_2(|K|)} < C(1/2)^t \|f\|_{H^t(|K|)}$$

for $t \leq 4$. While exactly matching splines for $L_2 = H^2$ norms on the right-hand side, the choice of smoothness structure results in the loss of estimates for H^1 and H^2.

The optimal choice of structure for estimates of this type remains open.

7. Conclusions

The survey we have presented is far from exhaustive. We did not discuss many important theoretical and algorithmic topics related to stationary subdivision on meshes. There are a few important extensions. Examples include variational subdivision [32] and PDE-based schemes [82,83,81], subdivision schemes in higher dimensions [4,49] and schemes for arbitrary mesh refinement [22]. Multiresolution surfaces based on subdivision were not considered either.

While there was a rapid progress in subdivision theory in the late 90s, few questions were resolved conclusively. While smoothness criteria exist and are well established, applying these criteria remains difficult, especially for parametric families of subdivision schemes and requires extensive computations. There are no criteria directly relating smoothness of limit surfaces to easy-to-verify conditions on mask coefficients; although, recent work by Prautzsch and Umlauf [78] is a promising step in this direction. Analysis approximation properties and fairness of limit surfaces is even further from completion.

In contrast, an increasing number of applications use subdivision as the surface representation of choice, and applications appear in other areas (e.g. subdivision-based finite elements). We hope that the needs of practical applications will encourage further theoretical advances.

References

1. M. Alexa. Refinement operators for triangle meshes. *Computer Aided Geometric Design*, 19(3):169–172, March 2002.
2. B. Allen, B. Curless, and Z. Popovic. Articulated body deformation from range scan data. In *SIGGRAPH '02: 29th International Conference on Computer Graphics and Interactive Techniques*, volume 21, pages 612–19. 2002.
3. G. Arben. *Approximation Properties of Subdivision Surfaces*. PhD thesis, University of Washnigton, 2001.
4. C. Bajaj, S. Schaefer, J. Warren, and G. Xu. A subdivision scheme for hexahedral meshes. *Visual Computer*, 18(5-6):343–56, 2002.
5. A. A. Ball and D. J. T. Storry. Conditions for tangent plane continuity over recursively generated B-spline surfaces. *ACM Transactions on Graphics*, 7(2):83–102, 1988.
6. L. Barthe and L. Kobbelt. Subdivision scheme tuning around extraordinary vertices. *Computer Aided Geometric Design*, 21(6):561–583, 2004.
7. H. Biermann, D. Kristjansson, and D. Zorin. Approximate boolean operations on free-form solids. In *Proceedings of SIGGRAPH 2001*, pages 185–94. 2001.
8. H. Biermann, A. Levin, and D. Zorin. Piecewise smooth subdivision surfaces with normal control. In *Proceedings of SIGGRAPH'00: 27th International Conference on Computer Graphics and Interactive Techniques Conference*, pages 113–20. 2000.
9. H. Biermann, I. Martin, F. Bernardini, and D. Zorin. Cut-and-paste editing of multiresolution surfaces. In *SIGGRAPH '02: 29th International Conference on Computer Graphics and Interactive Techniques*, volume 21, pages 312–21. 2002.
10. E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978.
11. A. S. Cavaretta, W. Dahmen, and C. A. Micchelli. Stationary subdivision. *Mem. Amer. Math. Soc.*, 93(453):vi+186, 1991.
12. F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: A new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 47(12):2039–72, 2000.
13. F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, and P. Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Computer Aided Design*, 34(2):137–48, 2002.
14. J. Claes, K. Beets, and F. Van Reeth. A corner-cutting scheme for hexagonal subdivision surfaces. In *Proceedings SMI. Shape Modeling International 2002*, pages 13–20. 2002. 17-22 May 2002.
15. N. A. Dodgson. An heuristic analysis of the classification of bivariate subdivision schemes. Technical Report 611, University of Cambridge Computer Laboratory, December 2004.
16. D. Doo. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Proceedings on Interactive Techniques in Computer Aided Design*, pages 157–165, Bologna, 1978.
17. D. Doo and M. Sabin. Analysis of the behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, 1978.

18. N. Dyn and D. Levin. The subdivision experience. *Wavelets, images, and surface fitting (Chamonix-Mont-Blanc, 1993)*, pages 229–244, 1994.
19. N. Dyn and D. Levin. Subdivision schemes in geometric modelling. *Acta Numerica*, 11:73–144, 2002.
20. N. Dyn, D. Levin, and J. A. Gregory. A 4-point interpolatory subdivision scheme for curve design. *Computer-Aided Geometric Design*, 4(4):257–68, 1987.
21. N. Dyn, D. Levin, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160–9, 1990.
22. I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pages 325–334, August 1999.
23. A. Habib and J. Warren. Edge and vertex insertion for a class of C^1 subdivision surfaces. *Computer-Aided Geometric Design*, 16(4):223–47, 1999.
24. B. Han. Classification and construction of bivariate subdivision schemes. In A. Cohen, J.-L. Merrien, and L. L. Schumaker, editors, *Curves and Surfaces*, pages 187–197, Saint-Malo, 2003.
25. H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, pages 295–302, July 1994.
26. I. Ivrissimtzis, N. Dodgson, M. Hassan, and M. Sabin. On the geometry of recursive subdivision. *International Journal Shape Modeling*, 8(1):23–42, June 2002.
27. I. Ivrissimtzis, N. Dodgson, and M. Sabin. A generative classification of mesh refinement rules with lattice transformations. *Computer Aided Geometric Design*, 21(1):99–109, 2004.
28. Q. Jiang and P. Oswald. Triangular $\sqrt{3}$-subdivision schemes: the regular case. *Journal of Computational and Applied Mathematics*, 156(1):47–75, 2003.
29. K. Karciauskas, J. Peters, and U. Reif. Shape characterization of subdivision surfaces – Case studies. *Computer-Aided Geometric Design*, 21(6):601–614, July 2004. http://authors.elsevier.com/sd/article/S0167839604000627.
30. A. Khodakovsky and P. Schröder. Fine level feature editing for subdivision surfaces. In *Proceedings of ACM Solid Modeling '99*, pages 203–211, 1999.
31. L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *European Association for Computer Graphics 17th Annual Conference and Exhibition. EUROGRAPHICS '96*, volume 15, pages C409–20, C485. 1996.
32. L. Kobbelt. A variational approach to subdivision. *Computer-Aided Geometric Design*, 13(8):743–761, 1996.
33. L. Kobbelt. square root 3-subdivision. In *Proceedings of SIGGRAPH'00: 27th International Conference on Computer Graphics and Interactive Techniques Conference*, pages 103–12. 2000.

34. L. Kobbelt, J. Vorsatz, U. Labsik, and H.-P. Seidel. A shrink wrapping approach to remeshing polygonal surfaces. In *European Association for Computer Graphics 20th Annual Conference. EUROGRAPHICS'99*, volume 18, pages C119–29, C405. 1999.
35. P. Krysl, E. Grinspun, and P. Schröder. Natural hierarchical refinement for finite element methods. *International Journal for Numerical Methods in Engineering*, 56(8):1109–24, 2003.
36. T. Kurihara. Interactive surface design using recursive subdivision. In *Proceedings of Computer Graphics International '93*, pages 228–43. 1993.
37. U. Labsik and G. Greiner. Interpolatory square root 3-subdivision. In *European Association for Computer Graphics. 21st Annual Conference. EUROGRAPHICS*, volume 19, pages C131–8, 528. 2000.
38. A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *Proceedings of SIGGRAPH'00: 27th International Conference on Computer Graphics and Interactive Techniques Conference*, pages 85–94. 2000.
39. C. K. Lee. Automatic metric 3d surface mesh generation using subdivision surface geometrical model. 1.construction of underlying geometrical model. *International Journal for Numerical Methods in Engineering*, 56(11):1593–614, 2003.
40. C. K. Lee. Automatic metric 3d surface mesh generation using subdivision surface geometrical model. 2. mesh generation algorithm and examples. *International Journal for Numerical Methods in Engineering*, 56(11):1615–46, 2003.
41. L. Linsen, V. Pascucci, M. A. Duchaineau, B. Hamann, and K. I. Joy. Hierarchical representation of time-varying volume data with $\sqrt{2}^4$ subdivision and quadrilinear B-spline wavelets. In *Proceedings 10th Pacific Conference on Computer Graphics and Applications*, pages 346–55. 2002.
42. N. Litke, A. Levin, and P. Schröder. Fitting subdivision surfaces. In *Proceedings VIS 2001. Visualization 2001*, pages 319–568. 2001.
43. N. Litke, A. Levin, and P. Schröder. Trimming for subdivision surfaces. *Computer-Aided Geometric Design*, 18(5):463–81, 2001.
44. Y.-J. Liu, M. M.-F. Yuen, and S. Xiong. A feature-based approach for individualized human head modeling. *Visual Computer*, 18(5-6):368–81, 2002.
45. C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.
46. C. Loop. Bounded curvature triangle mesh subdivision with the convex hull property. *Visual Computer*, 18(5-6):316–25, 2002.
47. M. Lounsbery, T. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, 1997.
48. W. Ma, X. Ma, and S.-K. Tso. A new algorithm for loop subdivision surface fitting. In *Proceedings of Computer Graphics Imaging*, pages 246–51. 2001.
49. R. MacCracken and K. I. Joy. Free-form deformations with lattices of arbitrary topology. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pages 181–188, August 1996.
50. M. Marinov and L. Kobbelt. Optimization techniques for approximation with

subdivision surfaces. In *ACM Symposium on Solid Modeling and Applications*, pages 113–122, 2004.

51. V. G. Maz'ja. *Sobolev spaces.* Springer Series in Soviet Mathematics. Springer-Verlag, Berlin, 1985. Translated from the Russian by T. O. Shaposhnikova.
52. A. H. Nasri. Polyhedral subdivision methods for free-form surfaces. *ACM Transactions on Graphics*, 6(1):29–73, January 1987.
53. A. H. Nasri. Curve interpolation in recursively generated B-spline surfaces over arbitrary topology. *Computer-Aided Geometric Design*, 14(1):13–30, 1997.
54. A. H. Nasri. An algorithm for interpolating intersecting curves by recursive subdivision surfaces. In *Proceedings Shape Modeling International '99. International Conference on Shape Modeling and Applications*, pages 130–7, 274. 1999.
55. A. H. Nasri. Interpolating an unlimited number of curves meeting at extraordinary points on subdivision surfaces. *Computer Graphics Forum*, 22(1):87–97, 2003.
56. R. Ohbuchi, Y. Kokojima, and S. Takahashi. Blending shapes by using subdivision surfaces. *Computers & Graphics*, 25(1):41–58, 2001.
57. P. Oswald and P. Schröder. Composite primal/dual $\sqrt{3}$-subdivision schemes. *Computer-Aided Geometric Design*, 20(3):135–164, 2003.
58. J. Peters and A. H. Nasri. Computing volumes of solids enclosed by recursive subdivision surfaces. In *EUROGRAPHICS 97. The European Association for Computer Graphics 18th Annual Conference*, volume 16, pages C89–94. 1997.
59. J. Peters and U. Reif. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics*, 16(4):420–31, 1997.
60. J. Peters and U. Reif. Analysis of algorithms generalizing *b*-spline subdivision. *SIAM Journal on Numerical Analysis*, 35(2):728–748 (electronic), 1998.
61. J. Peters and U. Reif. Shape characterization of subdivision surfaces – Basic principles. *Computer-Aided Geometric Design*, 21(6):585–599, July 2004.
62. H. Prautzsch and G. Umlauf. A G^1 and a G^2 subdivision scheme for triangular nets. *International Journal of Shape Modeling*, 6(1):21–35, 2000. Habilitation.
63. U. Reif. A unified approach to subdivision algorithms near extraordinary vertices. *Computer-Aided Geometric Design*, 12(2):153–74, 1995.
64. U. Reif. A degree estimate for subdivision surfaces of higher regularity. *Proc. Amer. Math. Soc.*, 124(7):2167–2174, 1996.
65. U. Reif. Analyse und konstruktion von subdivisionsalgorithmen für freiformflächen beliebiger topologie, 1998.
66. M. A. Sabin. Interrogation of subdivision surfaces. *Handbook of computer aided geometric design*, pages 327–341, 2002.
67. M. A. Sabin. Subdivision surfaces. *Handbook of computer aided geometric design*, pages 309–325, 2002.
68. M. A. Sabin, N. A. Dodgson, M. F. Hassan, and I. P. Ivrissimtzis. Curvature behaviours at extraordinary points of subdivision surfaces. *Computer-Aided Design*, 35(11):1047–1051, September 2003.

69. S. Schaefer, J. Warren, and D. Zorin. Lofting curve networks using subdivision surfaces. In *Eurographics/SIGGRAPH Symposium on Geometry Processing*, pages 103–114, 2004.
70. V. Scheib, J. Haber, M. C. Lin, and H.-P. Seidel. Efficient fitting and rendering of large scattered data sets using subdivision surfaces. In *23rd Annual Conference of the Eurographics Association*, volume 21, pages 353–62, 630. 2002.
71. P. Schröder. Subdivision, multiresolution and the construction of scalable algorithms in computer graphics. *Multivariate approximation and applications*, pages 213–251, 2001.
72. P. Schröder. Subdivision as a fundamental building block of digital geometry processing algorithms. In *15th Toyota Conference: Scientific and Engineering Computations for the 21st Century - Methodologies and Applications*, volume 149, pages 207–19. 2002.
73. J. E. Schweitzer. *Analysis and Application of Subdivision Surfaces*. PhD thesis, University of Washington, Seattle, 1996.
74. S. Skaria, E. Akleman, and F. I. Parke. Modeling subdivision control meshes for creating cartoon faces. In *Proceedings International Conference on Shape Modeling and Applications*, pages 216–25. 2001.
75. J. Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Proceedings of SIGGRAPH 98: 25th International Conference on Computer Graphics and Interactive Techniques*, pages 395–404. 1998.
76. J. Stam. On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree. *Computer-Aided Geometric Design*, 18(5):383–96, 2001.
77. H. Suzuki, S. Takeuchi, and T. Kanai. Subdivision surface fitting to a range of points. In *Proceedings. Seventh Pacific Conference on Computer Graphics and Applications*, pages 158–67, 322. 1999.
78. G. Umlauf. A technique for verifying the smoothness of subdivison schemes. In M.L. Lucian and M. Neamtu, editors, *Geometric Modeling and Computing*, Seattle, 2003.
79. L. Velho. Quasi 4-8 subdivision. *Computer-Aided Geometric Design*, 18(4):345–57, 2001.
80. L. Velho and D. Zorin. 4-8 subdivision. *Computer Aided Geometric Design*, 18(5):397–427, June 2001.
81. J. Warren and H. Weimer. *Subdivision Methods for Geometric Design: A Constructive Approach*. Morgan Kaufmann, 2001.
82. H. Weimer and J. Warren. Subdivision schemes for thin plate splines. In *EUROGRAPHICS '98. 19th Annual Conference*, volume 17, pages C303–13, C392. 1998.
83. H. Weimer and J. Warren. Subdivision schemes for fluid flow. In *Proceedings of SIGGRAPH 99: 26th International Conference on Computer Graphics and Interactive Techniques*, pages 111–20. 1999.
84. Z. Xu and K. Kondo. Fillet operations with recursive subdivision surfaces. In *Proceedings of 6th IFIP Working Conference on Geometric Modelling: Fundamentals and Applications*, pages 269–84. 2001.

85. H. Zhang and G. Wang. Honeycomb subdivision. *Journal of Software*, 13(7):1199–207, 2002.
86. D. Zorin. *Subdivision and Multiresolution Surface Representations*. PhD thesis, Caltech, Pasadena, 1997.
87. D. Zorin. A method for analysis of C^1-continuity of subdivision surfaces. *SIAM Journal on Numerical Analysis*, 37(5):1677–708, 2000.
88. D. Zorin. Smoothness of stationary subdivision on irregular meshes. *Constructive Approximation*, 16(3):359–397, 2000.
89. D. Zorin. Approximation on manifolds by bases locally reproducing polynomials. Unpublished manuscript, 2003.
90. D. Zorin and P. Schröder. A unified framework for primal/dual quadrilateral subdivision schemes. *Computer-Aided Geometric Design*, 18(5):429–54, 2001.
91. D. Zorin, P. Schröder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'96)*, pages 189–92. 1996.
92. D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pages 259–268, August 1997.

HIGH ORDER NUMERICAL METHODS FOR TIME DEPENDENT HAMILTON-JACOBI EQUATIONS

Chi-Wang Shu
Division of Applied Mathematics, Brown University
Providence, Rhode Island 02912, USA
E-mail: shu@dam.brown.edu

In these lectures we review a few high order accurate numerical methods for solving time dependent Hamilton-Jacobi equations. We will start with a brief introduction of the Hamilton-Jacobi equations, the appearance of singularities as discontinuities in the derivatives of their solutions hence the necessity to introduce the concept of viscosity solutions, and first order monotone numerical schemes on structured and unstructured meshes to approximate such viscosity solutions, which can be proven convergent with error estimates. We then move on to discuss high order accurate methods which are based on the first order monotone schemes as building blocks. We describe the Essentially Non-Oscillatory (ENO) and Weighted Essentially Non-Oscillatory (WENO) schemes for structured meshes, and WENO schemes and Discontinuous Galerkin (DG) schemes for unstructured meshes.

1. Introduction and Properties of Hamilton-Jacobi Equations

In these lectures we review high order accurate numerical methods for solving time dependent Hamilton-Jacobi equations

$$\varphi_t + H(\varphi_{x_1}, ..., \varphi_{x_d}) = 0, \qquad \varphi(x,0) = \varphi^0(x), \tag{1}$$

where H is a (usually nonlinear) function which is at least Lipschitz continuous. H could also depend on φ, x and t in some applications, however the main difficulty for numerical solutions is the nonlinear dependency of H on the gradient of φ.

Hamilton-Jacobi equations appear often in many applications. One important application of Hamilton-Jacobi equations is the area of image processing and computer vision, which is the main theme of this program at the Institute for Mathematical Sciences (IMS) of the National University of Singapore. Other application areas include, e.g. control and differential games.

It is easy to verify that global C^1 solution does not exist for (1) in the generic situation, regardless of the smoothness of the initial condition $\varphi^0(x)$. Singularities in the form of discontinuities in the derivatives of φ would appear at a finite time in most situations, thus the solutions would be Lipschitz continuous but no longer C^1. This could be verified, at least in the one dimensional case, by observing the equivalence between the Hamilton-Jacobi equation

$$\varphi_t + H(\varphi_x) = 0, \qquad \varphi(x,0) = \varphi^0(x) \tag{2}$$

and the hyperbolic conservation law

$$u_t + H(u)_x = 0, \qquad u(x,0) = u^0(x) \tag{3}$$

if we identify $u = \varphi_x$. Singularities for the conservation law (3) are in the form of discontinuities in the solution u, thus u is bounded, with a bounded total variation, but is not continuous. The study of singularities for (3) can be performed using characteristics, see for example [23,39,25]. Such results can be directly translated to that for the Hamilton-Jacobi equation (2) by integrating u once. Discontinuities in u then become discontinuities for the derivative of φ.

This lack of global smoothness of the solution φ in (1) makes it necessary to define a "weak" solution for the PDE (1), that is, a solution φ which may not satisfy the PDE (1) pointwise at every point. In particular, we would only require that φ satisfies the PDE (1) at any point where φ has continuous first derivatives. At those points where the first derivatives of φ are not continuous, a different requirement is needed for the solution φ to be an acceptable weak solution. For the hyperbolic conservation law (3), the requirements at the discontinuities of u include the so-called Rankine-Hugoniot jump condition, which relates the moving speed of the discontinuity with its strength and is derived from an integral version of the PDE (3), and an entropy condition which singles out a unique, physically relevant weak solution from many candidates. For the Hamilton-Jacobi equation (2) or in general (1), the requirements at the discontinuities of the derivatives of φ are characterized by certain inequalities which single out the unique, physically relevant "viscosity solution" of the Hamilton-Jacobi equation. To be more precise, φ is called a viscosity sub-solution of (1) if, for any smooth function ψ, at each local maximum point $(\bar{x}, \bar{t})$ of $\varphi - \psi$, we have the inequality

$$\psi_t(\bar{x},\bar{t}) + H(\psi_{x_1}(\bar{x},\bar{t}),...,\psi_{x_d}(\bar{x},\bar{t})) \leq 0.$$

Similarly, φ is called a viscosity super-solution of (1) if, for any smooth function ψ, at each local minimum point $(\bar{x},\bar{t})$ of $\varphi - \psi$, we have the inequality

$$\psi_t(\bar{x},\bar{t}) + H(\psi_{x_1}(\bar{x},\bar{t}),...,\psi_{x_d}(\bar{x},\bar{t})) \geq 0.$$

ψ is called the viscosity solution to (1) if it is both a viscosity sub-solution and a viscosity super-solution of (1). For more details, see for example [13].

For the purpose of numerical approximations to the Hamilton-Jacobi equation (1), we would need to pay special attention to the following properties of its viscosity solution φ:

- The viscosity solution φ may contain discontinuous derivatives. In applications, most solutions we encounter are piecewise smooth.
- The weak solution φ may not be unique. There are extra requirements at the discontinuities of the derivatives of φ to make it the unique, physically relevant viscosity solution.

For simplicity of notations we shall mostly concentrate on the two dimensional case, namely $d = 2$ in (1). In this case we will use x, y instead of x_1 and x_2. The equation (1) is then rewritten as

$$\varphi_t + H(\varphi_x, \varphi_y) = 0, \qquad \varphi(x,y,0) = \varphi^0(x,y). \tag{4}$$

2. First Order Monotone Schemes

In this section we will briefly describe first order monotone schemes for solving the Hamilton-Jacobi equation (4), both on structured meshes and on unstructured meshes. These first order monotone schemes will be used as building blocks for high order schemes to be described in the following sections.

2.1. *Monotone schemes on structured rectangular meshes*

We first consider monotone schemes on structured rectangular meshes. For simplicity of notations we will assume that the mesh is uniform in x and y. This simplification is not essential: all of the discussions below can be applied to non-uniform Cartesian meshes with obvious modifications. We denote by Δx and Δy the mesh sizes in x and y respectively, and denote by $\varphi_{i,j}$ the numerical approximation to the viscosity solution of (4), $\varphi(x_i, y_j, t) = \varphi(i\Delta x, j\Delta y, t)$. We also use the standard notations

$$\Delta_\pm^x \varphi_{i,j} = \pm\left(\varphi_{i\pm1,j} - \varphi_{i,j}\right), \qquad \Delta_\pm^y \varphi_{i,j} = \pm\left(\varphi_{i,j\pm1} - \varphi_{i,j}\right).$$

First order monotone schemes [14] are defined as schemes of the form

$$\frac{d}{dt}\varphi_{i,j} = -\hat{H}\left(\frac{\Delta_-^x \varphi_{i,j}}{\Delta x}, \frac{\Delta_+^x \varphi_{i,j}}{\Delta x}; \frac{\Delta_-^y \varphi_{i,j}}{\Delta y}, \frac{\Delta_+^y \varphi_{i,j}}{\Delta y}\right) \tag{5}$$

where $\hat{H}$ is called a numerical Hamiltonian, which is a Lipschitz continuous function of all four arguments and is consistent with the Hamiltonian H in the PDE (4):

$$\hat{H}(u, u; v, v) = H(u, v).$$

A monotone numerical Hamiltonian $\hat{H}$ is one which is monotonically non-decreasing in the first and third arguments and monotonically non-increasing in the other two. This can be symbolically represented as

$$\hat{H}\left(\uparrow, \downarrow; \uparrow, \downarrow\right).$$

The scheme (5) with a monotone numerical Hamiltonian is called a monotone scheme. We give here the semi-discrete (continuous in time) form of the monotone scheme. The fully discrete scheme can be obtained by using forward Euler in time. It is also called a monotone scheme.

It is proven in [14] that monotone schemes have the following favorable properties:

- Monotone schemes are stable in the L^∞ norm;
- Monotone schemes are convergent to the viscosity solution of (4);
- The error between the numerical solution of a monotone scheme and the exact viscosity solution of (4), measured in the L^∞ norm, is at least half order $O(\sqrt{\Delta x})$.

The low half order error estimate is not a particular concern for viscosity solutions containing kinks (discontinuities in the first derivatives). In fact, it can be shown that for many cases, this half order error estimate is optimal. However, it is an unfortunate fact that monotone schemes cannot be higher than first order accurate *for smooth solutions.* This is indeed a serious concern, as we would hope the scheme to be high order accurate for smooth solutions, or in smooth regions of non-smooth solutions. Monotone schemes would not be able to achieve this.

The importance of monotone schemes is that they are often used as building blocks for high order schemes. All the high order schemes discussed in these lectures are built upon first order monotone schemes. Thus it is important to know a few typical monotone schemes and their relative merits.

The simplest monotone flux is the Lax-Friedrichs flux [14,32]:

$$\begin{aligned}\hat{H}^{LF}\left(u^-,u^+;v^-,v^+\right) = H\left(\frac{u^-+u^+}{2},\frac{v^-+v^+}{2}\right)\\ -\frac{1}{2}\alpha^x\left(u^+-u^-\right)-\frac{1}{2}\alpha^y\left(v^+-v^-\right)\end{aligned} \qquad (6)$$

where

$$\alpha^x = \max_{\substack{A\le u\le B\\ C\le v\le D}} \left|H_1(u,v)\right|, \qquad \alpha^y = \max_{\substack{A\le u\le B\\ C\le v\le D}} \left|H_2(u,v)\right|. \qquad (7)$$

Here $H_i(u,v)$ is the partial derivative of H with respect to the i-th argument, or the Lipschitz constant of H with respect to the i-th argument. It can be easily shown that $\hat{H}^{LF}$ is monotone for $A \le u \le B$ and $C \le v \le D$.

Another slightly different Lax-Friedrichs flux is

$$\begin{aligned}\hat{H}^{LF}\left(u^-,u^+;v^-,v^+\right) = \frac{1}{4}\left(H(u^-,v^-)+H(u^+,v^-)+H(u^-,v^+)+\right.\\ \left.H(u^+,v^+)\right)-\frac{1}{2}\alpha^x\left(u^+-u^-\right)-\frac{1}{2}\alpha^y\left(v^+-v^-\right)\end{aligned} \qquad (8)$$

where α^x and α^y are chosen the same way as before by (7). This flux is also monotone for $A \le u \le B$ and $C \le v \le D$.

The Godunov type monotone flux is defined as [5]:

$$\hat{H}^G\left(u^-,u^+;v^-,v^+\right) = \text{ext}_{u\in I(u^-,u^+)}\,\text{ext}_{v\in I(v^-,v^+)}\ H(u,v) \qquad (9)$$

where

$$I(a,b) = [\min(a,b), \max(a,b)]$$

and the function ext is defined by

$$\text{ext}_{u\in I(a,b)} = \begin{cases}\min_{a\le u\le b} & \text{i}f \quad a\le b,\\ \max_{b\le u\le a} & \text{i}f \quad a>b.\end{cases}$$

As pointed out in [5], since in general

$$\min_u \max_v H(u,v) \neq \max_v \min_u H(u,v),$$

we will generally obtain different versions of the Godunov type fluxes $\hat{H}^G$ by changing the order of the min and the max.

The local Lax-Friedrichs flux is defined as

$$\hat{H}^{LLF}\left(u^-,u^+;v^-,v^+\right) = H\left(\frac{u^-+u^+}{2},\frac{v^-+v^+}{2}\right) - \frac{1}{2}\alpha^x(u^-,u^+)\left(u^+-u^-\right) - \frac{1}{2}\alpha^y(v^-,v^+)\left(v^+-v^-\right) \qquad (10)$$

where

$$\alpha^x(u^-,u^+) = \max_{\substack{u\in I(u^-,u^+)\\ C\le v\le D}} \left|H_1(u,v)\right|,$$
$$\alpha^y(v^-,v^+) = \max_{\substack{A\le u\le B\\ v\in I(v^-,v^+)}} \left|H_2(u,v)\right|. \qquad (11)$$

It is proven in [32] that the local Lax-Friedrichs flux $\hat{H}^{LLF}$ is monotone for $A \le u \le B$ and $C \le v \le D$. The local Lax-Friedrichs flux $\hat{H}^{LLF}$ has smaller dissipation than the (global) Lax-Friedrichs flux $\hat{H}^{LF}$.

It would seem that a more local Lax-Friedrichs flux could be

$$\hat{H}^{LLLF}\left(u^-,u^+;v^-,v^+\right) = H\left(\frac{u^-+u^+}{2},\frac{v^-+v^+}{2}\right) - \frac{1}{2}\alpha^x(u^-,u^+;v^-,v^+)\left(u^+-u^-\right) - \frac{1}{2}\alpha^y(u^-,u^+;v^-,v^+)\left(v^+-v^-\right)$$

where

$$\alpha^x(u^-,u^+;v^-,v^+) = \max_{\substack{u\in I(u^-,u^+)\\ v\in I(v^-,v^+)}} \left|H_1(u,v)\right|,$$
$$\alpha^y(u^-,u^+;v^-,v^+) = \max_{\substack{u\in I(u^-,u^+)\\ v\in I(v^-,v^+)}} \left|H_2(u,v)\right|.$$

This would be easier to compute and also would have even smaller dissipation than the local Lax-Friedrichs flux $\hat{H}^{LLF}$ defined in (7). Unfortunately, it is shown in [32] that $\hat{H}^{LLLF}$ is *not* a monotone flux.

Another very useful monotone flux is the Roe flux with entropy fix [32]:

$$\hat{H}^{RF}\left(u^-,u^+;v^-,v^+\right) = \begin{cases} H(u^*,v^*) & \text{Case 1;}\\ H\left(\frac{u^-+u^+}{2},v^*\right) - \frac{1}{2}\alpha^x(u^-,u^+)\left(u^+-u^-\right) & \text{Case 2;}\\ H\left(u^*,\frac{v^-+v^+}{2}\right) - \frac{1}{2}\alpha^y(v^-,v^+)\left(v^+-v^-\right) & \text{Case 3;}\\ \hat{H}^{LLF}\left(u^-,u^+;v^-,v^+\right) & \text{Case 4.} \end{cases} \qquad (12)$$

where Case 1 refers to the situation when $H_1(u,v)$ and $H_2(u,v)$ do not change signs in the region $u \in I(u^-,u^+)$ and $v \in I(v^-,v^+)$; Case 2 refers to the remaining situations and when $H_2(u,v)$ does not change sign in the region $A \le u \le B$ and $v \in I(v^-, v^+)$; Case 3 refers to the remaining situations and when $H_1(u,v)$ does not change sign in the region $u \in I(u^-, u^+)$ and $C \le v \le D$; and finally Case 4 refers to all remaining situations. Here u^* and v^* are defined by upwinding

$$u^* = \begin{cases} u^-, & \text{if} \quad H_1(u,v) \ge 0; \\ u^+, & \text{if} \quad H_1(u,v) \le 0; \end{cases} \qquad v^* = \begin{cases} v^-, & \text{if} \quad H_2(u,v) \ge 0; \\ v^+, & \text{if} \quad H_2(u,v) \le 0. \end{cases}$$

This Roe flux with local Lax-Friedrichs entropy fix is easy to code and has almost as small a numerical viscosity as the (much more complicated) Godunov flux, hence it is quite popular.

All the monotone fluxes considered above apply to a general Hamiltonian H. There are also simple monotone fluxes which apply to H of certain specific forms. The most noticeable example is the Osher-Sethian flux [31], which applies to Hamiltonians of the form $H(u,v) = f(u^2, v^2)$ where f is a monotone function of each argument:

$$\hat{H}^{OS}\left(u^-, u^+, v^-, v^+\right) = f(\bar{u}^2, \bar{v}^2) \tag{13}$$

where $\bar{u}^2$ and $\bar{v}^2$ are implemented by

$$\bar{u}^2 = \begin{cases} (\min(u^-,0))^2 + (\max(u^+,0))^2, & \quad \text{if} \quad f(\downarrow, \cdot) \\ (\min(u^+,0))^2 + (\max(u^-,0))^2, & \quad \text{if} \quad f(\uparrow, \cdot) \end{cases}$$

$$\bar{v}^2 = \begin{cases} (\min(v^-,0))^2 + (\max(v^+,0))^2, & \quad \text{if} \quad f(\cdot, \downarrow) \\ (\min(v^+,0))^2 + (\max(v^-,0))^2, & \quad \text{if} \quad f(\cdot, \uparrow)\,. \end{cases}$$

This numerical Hamiltonian is purely upwind and easy to program, hence it should be used whenever possible. However, we should point out that not all Hamiltonians H can be written in the form $f(u^2, v^2)$ with a monotone f. For example, $H(u,v) = \sqrt{au^2 + cv^2}$ is of this form for constants a and c, hence we can use the Osher-Sethian flux for it, but $H(u,v) = \sqrt{au^2 + 2buv + cv^2}$ is not of this form, hence Osher-Sethian flux does not apply and we must program a Godunov type monotone flux if we would like a purely upwind flux.

2.2. *Monotone schemes on unstructured meshes*

In many situations it is more convenient and efficient to use an unstructured mesh rather than a structured one described in the previous section. We can similarly define the concept of monotone schemes on unstructured meshes, which again serve as building stones for higher order schemes. In this section we only present the Lax-Friedrichs type monotone scheme on unstructured meshes of Abgrall [2]. Other monotone schemes can also be defined on unstructured meshes.

The equation (4) is solved in a general domain Ω, which has a triangulation $\mathcal{T}_h$ consisting of triangles. The nodes are named by their indices $0 \le i \le N$, with a total of $N+1$ nodes. For every node i, we define the k_i+1 angular sectors $T_0, \cdots, T_{k_i}$ meeting at the point i; they are the inner angles at node i of the triangles having i as a vertex. The indexing of the angular sectors is ordered counterclockwise. $\vec{n}_{l+\frac{1}{2}}$ is the unit vector of the half-line $D_{l+\frac{1}{2}} = T_l \bigcap T_{l+1}$, and θ_l is the inner angle of sector T_l, $0 \le l \le k_i$; see Figure 1.

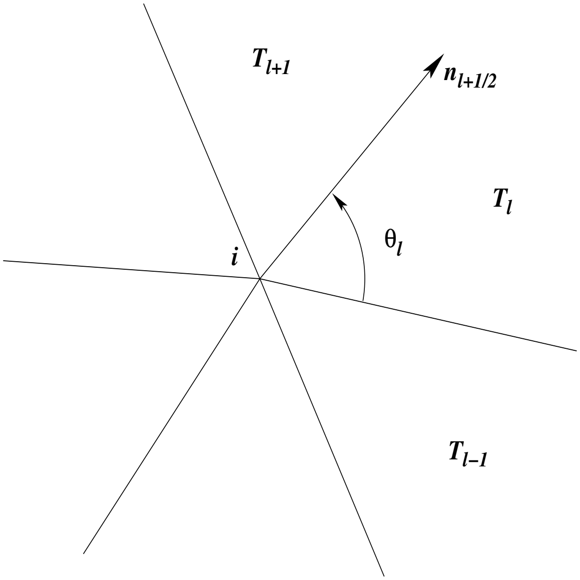


Fig. 1. Node i and its angular sectors.

We denote by φ_i the numerical approximation to the viscosity solution of (4) at node i. $(\nabla\varphi)_0, \cdots, (\nabla\varphi)_{k_i}$ will respectively represent the numerical approximation of $\nabla\varphi$ at node i in each angular sector $T_0, \cdots, T_{k_i}$.

The Lax-Friedrichs type monotone Hamiltonian for arbitrary triangulations developed by Abgrall in [2] is a generalization of the Lax-Friedrichs monotone Hamiltonian for Cartesian meshes described in the previous section. This monotone Hamiltonian is given by

$$\hat{H}((\nabla\varphi)_0, \cdots, (\nabla\varphi)_{k_i}) = H\left(\frac{\sum_{l=0}^{k_i} \theta_l (\nabla\varphi)_l}{2\pi}\right)$$

$$-\frac{\alpha}{\pi}\sum_{l=0}^{k_i} \beta_{l+\frac{1}{2}} \left(\frac{(\nabla\varphi)_l + (\nabla\varphi)_{l+1}}{2}\right) \cdot \vec{n}_{l+\frac{1}{2}} \qquad (14)$$

where

$$\beta_{l+\frac{1}{2}} = \tan\left(\frac{\theta_l}{2}\right) + \tan\left(\frac{\theta_{l+1}}{2}\right)$$
$$\alpha = \max\{\max_{\substack{A\le u\le B\\ C\le v\le D}} |H_1(u,v)|, \max_{\substack{A\le u\le B\\ C\le v\le D}} |H_2(u,v)|\}.$$

Here H_1 and H_2 are again the partial derivatives of H with respect to φ_x and φ_y, respectively, or the Lipschitz constants of H with respect to φ_x and φ_y, if H is not differentiable. $[A, B]$ is the value range for $(\varphi_x)_l$, and $[C, D]$ is the value range for $(\varphi_y)_l$, over $0 \le l \le k_i$ for the local Lax-Friedrichs Hamiltonian, and over $0 \le l \le k_i$ and $0 \le i \le N$ for the global Lax-Friedrichs Hamiltonian.

The $\hat{H}$ in (14) defines a monotone Hamiltonian. It is Lipschitz continuous in all arguments and is consistent with H, i.e., $\hat{H}(\nabla\varphi, \cdots, \nabla\varphi) = H(\nabla\varphi)$. It is proven in [2] that the numerical solution of the monotone scheme using this numerical Hamiltonian converges to the viscosity solution of (4), with the same half order convergence rate in the L^∞ norm for regular triangulations, namely for such triangulations where the ratio between the radii of the smallest circle outside a triangle and the largest circle inside the triangle stays bounded during mesh refinement.

3. High Order ENO and WENO Schemes on Structured Rectangular Meshes

In this section we describe the high order ENO (essentially non-oscillatory) and WENO (weighted ENO) schemes on structured rectangular meshes for solving the two dimensional Hamilton-Jacobi equations (4). Schemes for higher spatial dimensions are similar. We will only consider spatial discretizations in this section. Time discretization will be described in section 6.

We first explain the meaning of “high order” when the solution contains possible discontinuities for its derivatives. In such situations high order accuracy refers to a formal high order truncation error in smooth regions of the solution. Thus in general we can only expect high order accuracy in smooth regions away from derivative singularities. However, typically high order methods also have a sharper resolution for the derivative singularities. Thus high order methods are also referred to as “high resolution” schemes, especially when applied to conservation laws.

3.1. *High order ENO schemes*

High order ENO schemes for solving Hamilton-Jacobi equations were developed in [31] for the second order case and in [32] for the more general cases, based on ENO schemes for solving conservations laws [17,37,38]. We refer to the lecture notes of Shu [36] for more details of ENO and WENO schemes.

The key idea of ENO schemes is an adaptive stencil interpolation procedure, which automatically obtains information from the locally smoothest region, and hence yields a uniformly high-order essentially non-oscillatory approximation for piecewise smooth functions.

We first summarize the ENO interpolation procedure, which is used for building ENO schemes to solve the Hamilton-Jacobi equations (4). Given point values $f(x_j)$, $j = 0, \pm 1, \pm 2, ...$ of a (usually piecewise smooth) function $f(x)$ at discrete nodes x_j, we associate an r-th degree polynomial $P^{f,r}_{j+1/2}(x)$ with each interval $[x_j, x_{j+1}]$, constructed inductively as follows:

(1) We start with a first degree polynomial interpolating at the two boundary nodes of the target interval $[x_j, x_{j+1}]$ and denote the left-most point in its stencil by $k^1_{\min}$:

$$P^{f,1}_{j+1/2}(x) = f[x_j] + f[x_j, x_{j+1}](x - x_j), \qquad k^1_{\min} = j;$$

(2) If $k^{m-1}_{\min}$ and $P^{f,m-1}_{j+1/2}(x)$ are both defined, then let

$$a^{(m)} = f[x_{k^{m-1}_{\min}}, ..., x_{k^{m-1}_{\min}+m}], \qquad b^{(m)} = f[x_{k^{m-1}_{\min}-1}, ..., x_{k^{m-1}_{\min}+m-1}],$$

and

(a) If $|a^{(m)}| \geq b^{(m)}$, then $c^{(m)} = b^{(m)}$, $k^m_{\min} = k^{m-1}_{\min} - 1$; otherwise $c^{(m)} = a^{(m)}$, $k^m_{\min} = k^{m-1}_{\min}$,

(b) The ENO polynomial of the next higher degree is defined by

$$P^{f,m}_{j+1/2}(x) = P^{f,m-1}_{j+1/2}(x) + c^{(m)} \prod_{i=k^{m-1}_{\min}}^{k^{m-1}_{\min}+m-1} (x - x_i).$$

In the procedure above, $f[\,\cdot\,,\,\cdots\,,\,\cdot\,]$ are the standard Newton divided differences defined inductively as

$$f[x_i] = f(x_i); \qquad f[x_i, ..., x_{i+m}] = \frac{f[x_{i+1}, ..., x_{i+m}] - f[x_i, ..., x_{i+m-1}]}{x_{i+m} - x_i}.$$

Note that we start from the first degree polynomial $P^{f,1}$ with a stencil of two points, which would generate a first order monotone scheme in the procedure below.

Clearly, the ENO interpolation procedure starts with a base stencil containing 2 grid points, then adaptively adds one point to the stencil at each stage, which is either the left neighboring point or the right neighboring point to the current stencil depending on which would yield a smaller (in magnitude) divided difference together with points in the current stencil.

It can be shown that this ENO interpolation procedure can generate high order approximation yet avoids spurious oscillations, in the sense of yielding a total variation of the interpolant being at most $O(\Delta x^r)$ larger than the total variation of the piecewise smooth function $f(x)$ being interpolated. Thus the ENO procedure is especially suited for problems with singular but piecewise smooth solutions, such as solutions to conservation laws or Hamilton-Jacobi equations.

High order ENO schemes use monotone fluxes described in section 2.1 as building blocks and the ENO interpolation procedure described above to compute high order approximations to the left and right derivatives. The algorithm can be summarized as follows:

(1) At any node (x_i, y_j), fix j to compute along the x-direction, by using the ENO interpolation procedure, to obtain

$$u^{\pm}_{i,j} = \frac{d}{dx} P^{\varphi,r}_{i\pm 1/2,j}(x_i). \tag{15}$$

(2) Similarly, at the node (x_i, y_j), fix i to compute along the y-direction, by using the ENO interpolation procedure, to obtain

$$v^{\pm}_{i,j} = \frac{d}{dy} P^{\varphi,r}_{i,j\pm 1/2}(y_j). \tag{16}$$

(3) Form the semi-discrete r-th order ENO scheme

$$\frac{d}{dt}\varphi_{i,j} = -\hat{H}(u^-_{i,j}, u^+_{i,j}; v^-_{i,j}, v^+_{i,j}). \tag{17}$$

This semi-discrete ENO scheme will be discretized in time by the high order strong stability preserving Runge-Kutta time discretizations, to be described in section 6.

Numerical results obtained with these ENO schemes can be found in [31] and [32] and will be not be presented here.

3.2. *High order WENO schemes*

WENO schemes are designed based on ENO schemes. Both ENO and WENO schemes use the idea of adaptive stencils in the interpolation procedure based on the local smoothness of the numerical solution to automatically achieve high order accuracy and a non-oscillatory property near discontinuities. ENO uses just one (optimal in some sense) out of many candidate stencils when doing the interpolation, as is described in the previous section, while WENO uses a convex combination of all the candidate stencils, each being assigned a nonlinear weight which depends on the local smoothness of the numerical solution based on that stencil. WENO improves upon ENO in robustness, better smoothness of fluxes, better steady state convergence, better provable convergence properties, and more efficiency. For more details regarding WENO schemes, we again refer to the lecture notes [36].

High order WENO schemes for solving Hamilton-Jacobi equations were developed in [20], based on WENO schemes for solving conservations laws [30,21]. The framework of WENO schemes for solving Hamilton-Jacobi equations is similar to that of ENO schemes described in the previous section. The only difference is the interpolation procedure to obtain $u_{i,j}^{\pm}$ and $v_{i,j}^{\pm}$.

Let us look at the fifth order WENO interpolation procedure to obtain $u_{i,j}^{-}$ as an example. When the third order ENO interpolation procedure (see the previous section) is used, we can easily work out the algebra to obtain the three possible interpolations to $u_{i,j}^{-}$:

$$\begin{aligned}
u_{i,j}^{-,0} &= \frac{1}{3}\frac{\Delta_x^+\varphi_{i-3,j}}{\Delta x} - \frac{7}{6}\frac{\Delta_x^+\varphi_{i-2,j}}{\Delta x} + \frac{11}{6}\frac{\Delta_x^+\varphi_{i-1,j}}{\Delta x}, \\
u_{i,j}^{-,1} &= -\frac{1}{6}\frac{\Delta_x^+\varphi_{i-2,j}}{\Delta x} + \frac{5}{6}\frac{\Delta_x^+\varphi_{i-1,j}}{\Delta x} + \frac{1}{3}\frac{\Delta_x^+\varphi_{i,j}}{\Delta x}, \\
u_{i,j}^{-,2} &= \frac{1}{3}\frac{\Delta_x^+\varphi_{i-1,j}}{\Delta x} + \frac{5}{6}\frac{\Delta_x^+\varphi_{i,j}}{\Delta x} - \frac{1}{6}\frac{\Delta_x^+\varphi_{i+1,j}}{\Delta x},
\end{aligned} \tag{18}$$

depending on which of the three possible stencils

$$\{x_{i-3}, x_{i-2}, x_{i-1}, x_i\}, \qquad \{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}, \qquad \{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$$

(where y_j is omitted in the stencil as it is the same for all three stencils) are chosen by the ENO stencil choosing procedure based on the magnitudes of the divided differences. Recall that $\Delta_x^+ \varphi_{i,j} = \varphi_{i+1,,j} - \varphi_{i,j}$ is the standard forward difference operator in x. If the third order ENO scheme is used, one of the $u_{i,j}^{-;m}$ for $m = 0, 1$ or 2 is used as $u_{i,j}^-$. The WENO procedure however uses a convex combination of all three $u_{i,j}^{-;m}$ for the final approximation $u_{i,j}^-$:

$$u_{i,j}^- = w_0 u^{-,0} + w_1 u^{-,1} + w_2 u^{-,2} \tag{19}$$

where $w_s \geq 0$ are the nonlinear weights obeying $w_0 + w_1 + w_2 = 1$. The weights w_s are chosen to satisfy the following two properties:

(1) In smooth regions, $\{w_0, w_1, w_2\}$ should be very close to the so-called optimal linear weights $\{0.1, 0.6, 0.3\}$:

$$w_0 = 0.1 + O(\Delta x^2), \qquad w_1 = 0.6 + O(\Delta x^2), \qquad w_2 = 0.3 + O(\Delta x^2),$$

which makes $u_{i,j}^-$ defined by (19) fifth order accurate in approximating $\frac{\partial \varphi}{\partial x}(x_i, y_j)$ in smooth regions;

(2) When stencil s contains a singularity (discontinuity in the x derivative) of φ, the corresponding weight w_s should be very close to zero, so that the approximation $u_{i,j}^-$ emulates an ENO approximation where "bad" stencils make no contributions. In the choice of weights in [20] $w_s = O(\Delta x^4)$ when stencil s contains a singularity.

The key ingredient in designing a nonlinear weight to satisfying the two properties listed above is a smoothness indicator, which is a measurement of how smooth the function being interpolated is inside the interpolation stencil. The recipe used in [20] is similar to that in [21] for conservation laws, namely the smoothness indicator is a scaled sum of the squares of the L^2 norms of the second and higher derivatives of the interpolation polynomial on the target interval. These smoothness indicators work out to be

$$\begin{aligned} IS_0 &= 13(a-b)^2 + 3(a-3b)^2, \\ IS_1 &= 13(b-c)^2 + 3(b+c)^2, \\ IS_2 &= 13(c-d)^2 + 3(3c-d)^2, \end{aligned}$$

where

$$a = \frac{\Delta_x^2 \varphi_{i-2,j}}{\Delta x}, \qquad b = \frac{\Delta_x^2 \varphi_{i-1,j}}{\Delta x}, \qquad c = \frac{\Delta_x^2 \varphi_{i,j}}{\Delta x}, \qquad d = \frac{\Delta_x^2 \varphi_{i+1,j}}{\Delta x} \tag{20}$$

are the second order differences, defined by $\Delta_x^2 \varphi_{i,j} = \varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}$. With these smoothness indicators, the nonlinear weights are then defined by

$$w_0 = \frac{\tilde{w}_0}{\tilde{w}_0 + \tilde{w}_1 + \tilde{w}_2}, \qquad w_1 = \frac{\tilde{w}_1}{\tilde{w}_0 + \tilde{w}_1 + \tilde{w}_2}, \qquad w_2 = \frac{\tilde{w}_2}{\tilde{w}_0 + \tilde{w}_1 + \tilde{w}_2},$$

with

$$\tilde{w}_0 = \frac{1}{(\varepsilon + IS_0)^2}, \qquad \tilde{w}_1 = \frac{6}{(\varepsilon + IS_1)^2}, \qquad \tilde{w}_2 = \frac{3}{(\varepsilon + IS_2)^2},$$

where ε is a small number to prevent the denominator to become zero and is typically chosen as $\varepsilon = 10^{-6}$. Finally, after some algebraic manipulations, we obtain the fifth order WENO approximation to $u_{i,j}^-$ as

$$\begin{aligned} u_{i,j}^- = \frac{1}{12}\left(-\frac{\Delta_x^+ \varphi_{i-2,j}}{\Delta x} + 7\frac{\Delta_x^+ \varphi_{i-1,j}}{\Delta x} + 7\frac{\Delta_x^+ \varphi_{i,j}}{\Delta x} - \frac{\Delta_x^+ \varphi_{i+1,j}}{\Delta x}\right) \\ -\Phi^{WENO}(a,b,c,d) \end{aligned}$$

where

$$\Phi^{WENO}(a,b,c,d) = \frac{1}{3} w_0 \,(a - 2b + c) + \frac{1}{6}\left(w_2 - \frac{1}{2}\right)(b - 2c + d)$$

with a, b, c, d defined by (20).

By symmetry, the approximation to the right derivative $u_{i,j}^+$ is given by

$$\begin{aligned} u_{i,j}^+ = \frac{1}{12}\left(-\frac{\Delta_x^+ \varphi_{i-2,j}}{\Delta x} + 7\frac{\Delta_x^+ \varphi_{i-1,j}}{\Delta x} + 7\frac{\Delta_x^+ \varphi_{i,j}}{\Delta x} - \frac{\Delta_x^+ \varphi_{i+1,j}}{\Delta x}\right) \\ +\Phi^{WENO}(e,d,c,b) \end{aligned}$$

with b, c, d defined by (20) and e defined by

$$e = \frac{\Delta_x^2 \varphi_{i+2,j}}{\Delta x}.$$

The procedure to obtain $v_{i,j}^{\pm}$ is similar. Finally, we can form the semi-discrete fifth order WENO scheme as

$$\frac{d}{dt}\varphi_{i,j} = -\hat{H}(u_{i,j}^-, u_{i,j}^+; v_{i,j}^-, v_{i,j}^+). \tag{21}$$

This semi-discrete WENO scheme will be discretized in time by the high order strong stability preserving Runge-Kutta time discretizations, to be described in section 6. WENO schemes of different orders of accuracy can be defined along the same lines. For example, the third order WENO scheme is given by (21) with $u_{i,j}^-$ on the left-biased stencil $\{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$ defined by

$$\begin{aligned} u_{i,j}^- = &\frac{1}{2}\left(\frac{\Delta_x^+\varphi_{i-1,j}}{\Delta x} + \frac{\Delta_x^+\varphi_{i,j}}{\Delta x}\right) \\ &-\frac{w_-}{2}\left(\frac{\Delta_x^+\varphi_{i-2,j}}{\Delta x} - 2\frac{\Delta_x^+\varphi_{i-1,j}}{\Delta x} + \frac{\Delta_x^+\varphi_{i,j}}{\Delta x}\right) \end{aligned}$$

where

$$w_- = \frac{1}{1+2r_-^2}, \qquad r_- = \frac{\varepsilon + (\Delta_x^2\varphi_{i-1,j})^2}{\varepsilon + (\Delta_x^2\varphi_{i,j})^2}.$$

By symmetry, the approximation to $u_{i,j}^+$ on the right-biased stencil $\{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$ is defined by

$$\begin{aligned} u_{i,j}^+ = &\frac{1}{2}\left(\frac{\Delta_x^+\varphi_{i-1,j}}{\Delta x} + \frac{\Delta_x^+\varphi_{i,j}}{\Delta x}\right) \\ &-\frac{w_+}{2}\left(\frac{\Delta_x^+\varphi_{i+1,j}}{\Delta x} - 2\frac{\Delta_x^+\varphi_{i,j}}{\Delta x} + \frac{\Delta_x^+\varphi_{i-1,j}}{\Delta x}\right) \end{aligned}$$

where

$$w_+ = \frac{1}{1+2r_+^2}, \qquad r_+ = \frac{\varepsilon + (\Delta_x^2\varphi_{i+1,j})^2}{\varepsilon + (\Delta_x^2\varphi_{i,j})^2}.$$

Numerical results obtained with these WENO schemes can be found in [20] and will be not be presented here.

4. High Order WENO Schemes on Unstructured Meshes

In this section we describe high order WENO schemes for solving the two dimensional Hamilton-Jacobi equations (4) on unstructured triangular meshes. We will concentrate on the third order WENO scheme in [42]. For the fourth order WENO schemes, see [42] for details. We again use the first order monotone flux described in section 2.2 as building blocks.

The semi-discrete high order WENO scheme is given by:

$$\frac{d}{dt}\varphi_i(t) + \hat{H}((\nabla\varphi)_0, \cdots, (\nabla\varphi)_{k_i}) = 0 \qquad (22)$$

where $\hat{H}$ is the monotone flux described in section 2.2. The WENO procedure to obtain approximations to the sectional derivatives $(\nabla\varphi)_0$, ..., $(\nabla\varphi)_{k_i}$ will be described in detail below. The semi-discrete scheme (22) will be discretized in time by the high order strong stability preserving Runge-Kutta time discretizations, to be described in section 6.

First we discuss how to construct a high-order approximation to $\nabla\varphi$ in every angular sector of every node, see Figure 1. Let P^k denote the set of two-dimensional polynomials of degree less than or equal to k. We use Lagrange interpolations as follows: given a smooth function φ, and a triangulation with triangles $\{\triangle_0, \triangle_1, \ldots, \triangle_M\}$ and nodes $\{0, 1, 2, \ldots, N\}$, we would like to construct, for each triangle $\triangle_i$, a polynomial $p(x, y) \in P^k$, such that $p(x_l, y_l) = \varphi(x_l, y_l)$, where (x_l, y_l) are the coordinates of the three nodes of the triangle $\triangle_i$ and a few neighboring nodes. $p(x, y)$ would thus be a $(k+1)$th-order approximation to φ on the cell $\triangle_i$.

Because a kth degree polynomial $p(x, y)$ has $K = \frac{(k+1)(k+2)}{2}$ degrees of freedom, we need to use the information of at least K nodes. In addition to the three nodes of the triangle $\triangle_i$, we may take the other $K-3$ nodes from the neighboring cells around triangle $\triangle_i$. We rename these K nodes as $S_i = \{M_1, M_2, \ldots, M_K\}$, S_i is called a big stencil for the triangle $\triangle_i$. Let (x_i, y_i) be the barycenter of $\triangle_i$. Define $\xi = (x - x_i)/h_i$, $\eta = (y - y_i)/h_i$, where $h_i = \sqrt{|\triangle_i|}$ with $|\triangle_i|$ denoting the area of the triangle $\triangle_i$, then we can write $p(x, y)$ as:

$$p(x,y) = \sum_{j=0}^{k} \sum_{s+r=j} a_{sr} \xi^s \eta^r.$$

Using the K interpolation conditions:

$$p(M_l) = \varphi(M_l), \qquad l = 1, 2, \cdots, K,$$

we get a $K \times K$ linear system for the K unknowns a_{sr}. The normalized variables ξ, η are used to make the condition number of the linear system independent of mesh sizes.

It is well known that in two and higher dimensions such interpolation problem is not always well defined. The linear system can be very ill-conditioned or even singular, in such cases we would have to add more nodes to the big stencil S_i from the neighboring cells around triangle $\triangle_i$ to obtain an over-determined linear system, and then use the least-square method to solve it. We remark that this ill-conditioning may come from both the geometric distribution of the nodes, for which we could do nothing other than changing the mesh, and from the choice of basis functions in the interpolation. For higher order methods, a closer to orthogonal basis rather than $\xi^s \eta^r$ would be preferred, such as the procedure using barycentric coordinates in [1] and [3]. However, for third and fourth order cases, $\xi^s \eta^r$ can be used for simplicity.

After we have obtained the approximation polynomial $p(x,y)$ on the triangle $\triangle_i$, ∇p will be a kth-order approximation for $\nabla\varphi$ on $\triangle_i$. Hence we get the high-order approximation $\nabla p(x_l, y_l)$ to $\nabla\varphi(x_l, y_l)$, for any one of the three vertices (x_l, y_l) of the triangle $\triangle_i$, in the relevant angular sectors.

A scheme is called linear if it is linear when applied to a linear equation with constant coefficients. We need a third-order approximation for $\nabla\varphi$ to construct a third-order linear scheme, hence we need a cubic polynomial interpolation. A cubic polynomial p^3 has 10 degrees of freedom. We will use some or all of the nodes shown in Figure 2 to form our big stencil. For extremely distorted meshes the number of nodes in Figure 2 may be less than the required 10. In such extreme cases we would need to expand the choice for the big stencil, see [42] for details. For our target triangle $\triangle_0$, which has three vertices i, j, k and the barycenter G, we need to construct a cubic polynomial p^3, then ∇p^3 will be a third-order approximation for $\nabla\varphi$ on $\triangle_0$, and the values of ∇p^3 at points i, j and k will be third-order approximations for $\nabla\varphi$ at the angular sector $\triangle_0$ of nodes i, j and k. We label the nodes of the neighboring triangles of triangle $\triangle_0$ as follows: nodes 1, 2, 3 are the nodes (other than i, j, k) of neighbors of $\triangle_0$, nodes 4, 5, 6, 7, 8, 9 (other than 1, 2, 3, i, j, k) are the nodes of the neighbors of the three neighboring triangles of $\triangle_0$. Notice that the points 4, 5, 6, 7, 8, 9 do not have to be six distinct points. For example the points 5 and 9 could be the same point.

The interpolation points are nodes taken from a sorted node set. An ordering is given in the set so that, when the nodes are chosen sequentially from it to form the big stencil S_0, the target triangle $\triangle_0$ remains central to avoid serious downwind bias which could lead to linear instability. Referring to Figure 2, the interpolation points for the polynomial p^3 include nodes i, j, k and the nodes taken from the sorted set: $W = \{1,2,3,4,5,6,7,8,9\}$. The detailed procedure to determine the big stencil S_0 for the target triangle $\triangle_0$ is given below.

Procedure 1: The choice of the big stencil for the third-order scheme.

(1) Referring to Figure 2, we form a sorted node set: $W = \{1,2,3,4,5,6,7,8,9\}$. In extreme cases when this set does not contain enough distinct points, we may need to add more points from the next layer of neighbors.

(2) To start with, we take $S_0 = \{i,j,k,1,2,3,4,5,6,7\}$. Use this stencil S_0 to form the 10×10 interpolation coefficient matrix A.

(3) Compute the reciprocal condition number c of A. This is provided by

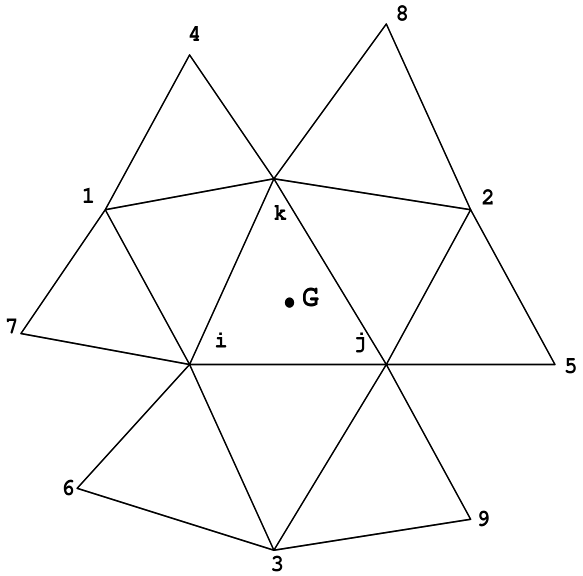


Fig. 2. The nodes used for the big stencil of the third-order scheme.

most linear solvers. If $c \geq \delta$ for some threshold δ, we have obtained the final stencil S_0. Otherwise, add the next node in W (i.e. node 8) to S_0. Use the 11 nodes in S_0 as interpolation points to get the 11×10 least square interpolation coefficient matrix A. Judge the reciprocal condition number c again. Continue in doing this until $c \geq \delta$ is satisfied. It seems that $\delta = 10^{-3}$ is a good threshold after extensive numerical experiments [42]. Notice that, since we have normalized the coordinates, this threshold does not change when the mesh is scaled uniformly in all directions. For all the triangulations tested in [42], at most 12 nodes are needed in S_0 to reach the condition $c \geq \delta$.

We now have obtained the big stencil S_0 and its associated cubic polynomial p^3. For each node (x_l, y_l) in $\triangle_0$, $\nabla p^3(x_l, y_l)$ is a third-order approximation to $\nabla\varphi(x_l, y_l)$. In order to construct a high-order WENO scheme, an important step is to obtain a high-order approximation using a linear combination of lower order approximations. We will use a linear combination of second-order approximations to get the same third-order approximation to $\nabla\varphi(x_l, y_l)$ as $\nabla p^3(x_l, y_l)$, i.e., we require

$$\frac{\partial}{\partial x}p^3(x_l, y_l) = \sum_{s=1}^{q} \gamma_{s,x} \frac{\partial}{\partial x} p_s(x_l, y_l), \qquad \frac{\partial}{\partial y}p^3(x_l, y_l) = \sum_{s=1}^{q} \gamma_{s,y} \frac{\partial}{\partial y} p_s(x_l, y_l) \tag{23}$$

where p_s are quadratic interpolation polynomials, and $\gamma_{s,x}$ and $\gamma_{s,y}$ are the linear weights for the x-directional derivative and the y-directional derivative respectively, for $s = 1, \cdots, q$. The linear weights are constants depending only on the local geometry of the mesh. The equalities in (23) should hold for any choices of the function φ.

Notice that to get a second-order approximation for the derivatives $\nabla\varphi(x_l, y_l)$, we need a quadratic interpolation polynomial. According to the argument in [19], the cubic polynomial $p^3(x, y)$ has four more degrees of freedom than each quadratic polynomial $p_s(x, y)$, namely x^3, x^2y, xy^2, y^3. For the six degrees of freedom $1, x, y, x^2, xy, y^2$, if we take $\varphi = 1, \varphi = x, \varphi = y, \varphi = x^2, \varphi = xy$ and $\varphi = y^2$, the equalities in (23) will hold for all these cases under only one constraint each on $\gamma_{s,x}$ and $\gamma_{s,y}$, namely $\sum_{s=1}^{q} \gamma_{s,x} = 1$ and $\sum_{s=1}^{q} \gamma_{s,y} = 1$, because p^3 and p_s all reproduce these functions exactly. Hence we should only need $q \geq 5$. $q = 5$ is taken in the scheme below.

We now need $q = 5$ small stencils $\Gamma_s, s = 1, \cdots, 5$ for the target triangle $\triangle_0$, satisfying $S_0 = \bigcup_{s=1}^{5} \Gamma_s$, and every quadratic polynomial p_s is associated with a small stencil Γ_s. In the third-order scheme, the small stencils will be the same for both directions x, y and all three nodes i, j, k in $\triangle_0$. However the linear weights $\gamma_{s,x}, \gamma_{s,y}$ can be different for different nodes i, j, k and different directions x, y. Because each quadratic polynomial has six degrees of freedom, the number of nodes in Γ_s must be at least six. To build a small stencil Γ_s, we start from several candidates $\Gamma_s^{(r)}, r = 1, 2, \cdots, n_s$. These candidates are constructed by first taking a point $A_s^{(r)}$ as the "center", then finding at least six nodes from S_0 which have the shortest distances from $A_s^{(r)}$ and can generate the interpolation coefficient matrix with a good condition number, using the method of Procedure 1. We then choose the best Γ_s among $\Gamma_s^{(r)}, r = 1, \cdots, n_s$ for every $s = 1, \cdots, 5$. Here "best" means that by using this group of small stencils, the linear weights $\gamma_{s,x}, \gamma_{s,y}, s = 1, \cdots, 5$ for all three nodes i, j, k are either all positive or have the smallest possible negative values in magnitude. The details of the algorithm is described in the following procedure.

Procedure 2: The third-order linear scheme.

For every triangle $\triangle_l, l = 1, \cdots, N$, do Steps 1 to 6:

(1) Follow Procedure 1 to obtain the big stencil S_l for $\triangle_l$.

(2) For $s = 1, \cdots, 5$, find the set $W_s = \{\Gamma_s^{(r)}, r = 1, 2, \cdots, n_s\}$, which are the candidate small stencils for the s-th small stencil. We use the following method to find the $\Gamma_s^{(r)}$ in W_s: first, nodes i, j, k are always included in every $\Gamma_s^{(r)}$; then we take a point $A_s^{(r)}$ as the center of $\Gamma_s^{(r)}$, detailed below, and find at least 3 additional nodes other than i, j, k from S_l which satisfy the following two conditions: 1) they have the shortest distances from $A_s^{(r)}$; and 2) taking them and the nodes i, j, k as the interpolation points, we will obtain the interpolation coefficient matrix A with a good condition number, namely the reciprocal condition number c of A satisfies $c \geq \delta$ with the same threshold $\delta = 10^{-3}$. For the triangulations tested in [42], at most 8 nodes are used to reach this threshold value. Finally, the center of the candidate stencils $A_s^{(r)}, r = 1, \cdots, n_s; s = 1, \cdots, 5$ are taken from the nodes around $\triangle_l$ (see Figure 2) as follows:

- $A_1^{(1)}$ = point G, $n_1 = 1$;
- $A_2^{(1)}$ = node 1, $A_2^{(2)}$ = node 4, $A_2^{(3)}$ = node 7, $n_2 = 3$;
- $A_3^{(1)}$ = node 2, $A_3^{(2)}$ = node 5, $A_3^{(3)}$ = node 8, $n_3 = 3$;
- $A_4^{(1)}$ = node 3, $A_4^{(2)}$ = node 6, $A_4^{(3)}$ = node 9, $n_4 = 3$;
- $\{A_5^{(r)}\}_{r=1}^{9}$ = nodes $4, 5, 6, 7, 8, 9$ and the middle points of nodes 4 and 8, 5 and 9, 6 and 7. $n_5 \leq 9$.

(3) By taking one small stencil $\Gamma_s^{(r_s)}$ from each $W_s, s = 1, \cdots, 5$ to form a group, we obtain $n_1 \times n_2 \times \cdots \times n_5$ groups of small stencils. We eliminate the groups which contain the same small stencils, and also eliminate the groups which do not satisfy the condition

$$\bigcup_{s=1}^{5} \Gamma_s^{(r_s)} = S_l$$

According to every group $\{\Gamma_s^{(r_s)}, s = 1, \cdots, 5\}$ of small stencils, we have 5 quadratic polynomials $\{p_s^{(r_s)}\}_{s=1}^{5}$. We evaluate $\frac{\partial}{\partial x} p_s^{(r_s)}$ and $\frac{\partial}{\partial y} p_s^{(r_s)}$ at points i, j, k, to obtain second-order approximation values for $\nabla \varphi$ at the three vertices of the triangle $\triangle_l$. We remark that for practical implementation, we do not use the polynomial itself, but compute a series of constants $\{a_l\}_{l=1}^{m}$ which depend on the local geometry only, such that:

$$\frac{\partial}{\partial x} p_s^{(r_s)}(x_n, y_n) = \sum_{l=1}^{m} a_l \varphi_l \qquad (24)$$

where every constant a_l corresponds to one node in the stencil $\Gamma_s^{(r_s)}$ and m is the total number of nodes in $\Gamma_s^{(r_s)}$. For every vertex (x_n, y_n) of triangle $\triangle_l$, we obtain a series of such constants. And for the y directional partial derivative, we compute the corresponding constants too.

(4) For every group $\{\Gamma_s^{(r_s)}, s = 1, \cdots, 5\}$, we form linear systems and solve them to get a series of linear weights $\gamma_{s,x}^{(r_s)}$ and $\gamma_{s,y}^{(r_s)}$ satisfying the equalities (23), for the three vertices i, j, k. Using the previous argument for combining low-order approximations to get high-order approximation, we form the linear system for $\gamma_{s,x}^{(r_s)}$ at a vertex (ξ_n, η_n) as follows (note that we use normalized variables): take $\varphi = \xi^3, \xi^2\eta, \xi\eta^2, \eta^3$ respectively, the equalities are:

$$\sum_{s=1}^{5} \gamma_{s,x}^{(r_s)} \frac{\partial}{\partial \xi} p_s^{(r_s)}(\xi_n, \eta_n) = \frac{\partial}{\partial \xi} \varphi(\xi_n, \eta_n) \tag{25}$$

where $p_s^{(r_s)}$ is the quadratic interpolation polynomial for φ, using stencil $\Gamma_s^{(r_s)}$. Again, in practical implementation, we will not use $p_s^{(r_s)}$ itself, instead we use the constants computed in the last step and equation (24) to compute the approximation for the derivatives of φ. Together with the requirement

$$\sum_{s=1}^{5} \gamma_{s,x}^{(r_s)} = 1, \tag{26}$$

we obtain a 5×5 linear system for $\gamma_{s,x}^{(r_s)}$. For $\gamma_{s,y}^{(r_s)}$, the same argument can be applied. Note that we need to compute the reciprocal condition number c for every linear system again. If $c \geq \delta$ for the same threshold $\delta = 10^{-3}$, we will accept this group of stencils as one of the remaining candidates. Otherwise, the linear system is considered to be ill-conditioned and its corresponding group of small stencils $\{\Gamma_s^{(r_s)}, s = 1, \cdots, 5\}$ is eliminated from further consideration.

(5) For each of the remaining groups $\Lambda_l = \{\Gamma_s^{(r_s)}, s = 1, \cdots, 5\}$, find the minimum value γ_l of all these linear weights $\gamma_{s,x}^{(r_s)}, \gamma_{s,y}^{(r_s)}$ of the three vertices i, j, k. Then find the group of small stencils whose γ_l is the biggest, and take this group as the final 5 small stencils for triangle $\triangle_l$. Denote them by $\Gamma_s, s = 1, \cdots, 5$. For every final small stencil $\Gamma_s, s = 1, 2, \cdots, 5$, we store the index numbers of the nodes in Γ_s, the constants in the linear combinations of node values to approximate values of $\nabla\varphi$

at points i, j, k, and the linear weights $\gamma_{s,x}$, $\gamma_{s,y}$ of the three points i, j, k.

(6) Now we have set up the necessary constants which only depend on the mesh for all triangles. To form the final linear scheme, we compute the third-order approximations $(\nabla\varphi)_0, \cdots, (\nabla\varphi)_{k_l}$ for all mesh nodes l, by the linear combinations of second-order approximations, using the prestored constants and linear weights. Then we can form the scheme (22).

We now describe the construction of WENO schemes based on non-linear weights.

We only discuss the case of WENO approximation for the x-directional derivative at vertex i of the target cell $\triangle_l$. Other cases are similar. In order to compute the non-linear weights, we need to compute the smoothness indicators first.

For a polynomial $p(x, y)$ defined on the target cell $\triangle_0$ with degree up to k, we take the smoothness indicator β as:

$$\beta = \sum_{2\le|\alpha|\le k} \int_{\triangle_0} |\triangle_0|^{|\alpha|-2} \left(D^\alpha p(x,y)\right)^2 dxdy \tag{27}$$

where α is a multi-index and D is the derivative operator. The smoothness indicator measures how smooth the function p is on the triangle $\triangle_0$: the smaller the smoothness indicator, the smoother the function p is on $\triangle_0$. The scaling factor in front of the derivatives renders the smoothness indicator self-similar and invariant under uniform scaling of the mesh in all directions. The smoothness indicator (27) is the same as that used for the structured mesh case discussed in the previous section.

Now we define the non-linear weights as:

$$\omega_j = \frac{\widetilde{\omega}_j}{\sum_m \widetilde{\omega}_m}, \qquad \widetilde{\omega}_j = \frac{\gamma_j}{(\varepsilon + \beta_j)^2} \tag{28}$$

where γ_j is the jth linear weight (e.g. the $\gamma_{s,x}$ in the linear schemes), β_j is the smoothness indicator for the jth interpolation polynomial $p_j(x, y)$ (the p_s in equation (23) for the third-order case) associated with the jth small stencil, and ε is again a small positive number to avoid the denominator to become 0 and is usually taken as $\varepsilon = 10^{-6}$. The final WENO approximation for the x-directional derivative at vertex i of target cell $\triangle_l$ is given by

$$(\varphi_x)_i = \sum_{j=1}^{q} \omega_j \frac{\partial}{\partial x} p_j(x_i, y_i) \tag{29}$$

where (x_i, y_i) are the coordinates of vertex i and $q = 5$ for the third-order schemes.

In the WENO schemes, the linear weights $\{\gamma_j\}_{j=1}^{q}$ depend on the local geometry of the mesh and can be negative. If $\min(\gamma_1, \cdots, \gamma_q) < 0$, we can adopt the splitting technique of treating negative weights in WENO schemes developed by Shi, Hu and Shu [34]. We omit the details of this technique and refer the readers to [34].

Again, we remark that the smoothness indicator (27) is a quadratic function of function values on nodes of the small stencil, so in practical implementation, to compute the smoothness indicator β_j for the j-th small stencil by equation (27), we do not need to use the interpolation polynomial itself, instead we use a series of constants $\{a_{rt}, r = 1, \cdots, t; t = 1, \cdots, m\}$, which can be precomputed and they depend on the mesh only, such that

$$\beta_j = \sum_{t=1}^{m} \varphi_t \left(\sum_{r=1}^{t} a_{rt}\varphi_r\right), \tag{30}$$

where m is the total number of nodes in the j-th small stencil. These constants for all smoothness indicators should be precomputed and stored once the mesh is generated.

We summarize the algorithm for the third-order WENO schemes as follows:

Procedure 3: The third-order WENO schemes.

(1) Generate a triangular mesh.
(2) Compute and store all constants which only depend on the mesh and the accuracy order of the scheme. These constants include the node index numbers of each small stencil, the coefficients in the linear combinations of function values on nodes of small stencils to approximate the derivative values and the linear weights, following Procedure 2 for the third-order case, and the constants for computing smoothness indicators in equation (30).
(3) Using the prestored constants, for each angular sector of every node i, compute the low-order approximations for $\nabla\varphi$ and the nonlinear weights, then compute the third order WENO approximation (29). Finally, form the scheme (22).

Numerical examples using the third and fourth order WENO schemes on unstructured meshes can be found in [42] and will not be presented here.

5. High Order Discontinuous Galerkin Schemes on Unstructured Meshes

Discontinuous Galerkin methods have become very popular in recent years to solve hyperbolic conservation laws because of their distinctive features, among which are the easy design of the methods with any order of accuracy and their minimal requirement on the mesh structures [12]. Adapted from these methods for conservation laws, a discontinuous Galerkin method for solving the Hamilton-Jacobi equations (1) was developed by Hu and Shu in [18] based on the equivalence between Hamilton-Jacobi equations and hyperbolic conservation laws [22,29]. See also [24]. In [18,24], the Hamilton-Jacobi equations (1) were first rewritten as a system of conservation laws

$$(w_i)_t + (H(\mathbf{w}))_{x_i} = 0, \text{in } \Omega \times [0,T], \quad \mathbf{w}(x,0) = \nabla\varphi^0(x), \tag{31}$$

where $\mathbf{w} = \nabla\varphi$. With piecewise polynomial space as the solution space, the usual discontinuous Galerkin formulation could be obtained for (31) [8,10]. Notice that $w_i, \quad i = 1, \cdots, n$ are not independent due to the restriction $\mathbf{w} = \nabla\varphi$. A least square procedure was applied in each time step (or each time stage depending on the particular time discretization used) to enforce this restriction in [18,24].

In a recent preprint by Li and Shu [27], we have given a reinterpretation and simplified implementation of the discontinuous Galerkin method for Hamilton-Jacobi equations developed in [18,24]. This was based on a recent work by Cockburn et al [9] and by Li and Shu [26], where the locally divergence-free discontinuous Galerkin methods were developed for partial differential equations with divergence-free solutions. Compared with traditional ways to solve this type of equations, the piecewise divergence-free polynomial space, which is a subspace of the standard piecewise polynomial space, is used. With minimal change in the scheme formulation (only the solution and test space is changed to a smaller space), the computational cost is reduced, the stability and the order of accuracy of the scheme are maintained. For specific applications such as the Maxwell equations [9] and the ideal magnetohydrodynamics (MHD) equations [26], this new method even improves over the traditional discontinuous Galerkin method in terms of stability and/or accuracy while saving computational costs. The idea of this approach could be applied to more general situations, by using piecewise solution space in which functions satisfy certain properties of the exact solutions (divergence-free, or curl-free, ...). The general approximation theory can guarantee no loss of accuracy when such smaller solution space is used. This observation leads to a reinterpretation and simplified implementation of the discontinuous Galerkin method for Hamilton-Jacobi equations developed in [18,24].

In this section we describe the discontinuous Galerkin method for solving the Hamilton-Jacobi equations developed in [18,24], using the reinterpretation in [27]. There are other similar or related types of discretizations for Hamilton-Jacobi equations on unstructured meshes, e.g. the schemes of Augoula and Abgrall [4] and that of Barth and Sethian [6], which will not be described in this section because of space limitations.

Starting with a regular triangulation $\mathcal{T}_h = \{K\}$ of Ω (edges denoted by e), the general discontinuous Galerkin formulation of (31) is: find $\mathbf{w} = (w_1, \cdots, w_n) \in \mathbf{V}^k$, such that

$$\frac{d}{dt}\int_K w_i v_i dx = \int_K H(\mathbf{w})(v_i)_{x_i} dx - \sum_{e\in\partial K}\int_e \hat{H}_{i,e,K} v_i ds, \quad \forall K, i = 1, \cdots, n \tag{32}$$

holds for all $\mathbf{v} = (v_1, \cdots, v_n) \in \mathbf{V}^k$, where $\mathbf{V}^k$ is the solution space which will be specified later, and $\hat{H}_{i,e,K}$ is the monotone numerical flux described in section 2.2. The strong stability preserving Runge-Kutta time discretization, to be described in section 6, could be used in time direction. Notice (32) is the formulation for the derivatives of φ in (1). To recover the missing constant in φ (e.g. the cell average of φ in each element), there are two different strategies developed in [18,24] which can be used:

(1) By requiring that

$$\int_K (\varphi_t + H(\varphi_x, \varphi_y))\, v\, dxdy \;=\; 0, \tag{33}$$

for all $v \in V_h^0$ and for all $K \in \mathcal{T}_h$, that is,

$$\int_K (\varphi_t + H(\varphi_x, \varphi_y))\; dxdy \;=\; 0, \qquad \forall K \in \mathcal{T}_h\,; \tag{34}$$

(2) By using (34) to update only one (or a few) elements, e.g., the corner element(s), then use

$$\varphi(B,t) = \varphi(A,t) + \int_A^B (\varphi_x\, dx + \varphi_y\, dy) \tag{35}$$

to determine the missing constant. The path should be taken to avoid crossing a derivative discontinuity, if possible.

We refer the readers to [18,24] for more details.

Before finalizing the scheme, we introduce the following spaces,

$$\mathbf{V}_1^k = \{(v_1, \cdots, v_n) : v_i|_K \in P^k(K), i = 1, \cdots, n, \forall K \in \mathcal{T}_h\}, \tag{36}$$

$$\mathbf{V}_2^k = \{(v_1, \cdots, v_n) : \mathbf{v}|_K = \nabla\varphi, \varphi \in P^{k+1}(K), \forall K \in \mathcal{T}_h\}, \qquad (37)$$

where $P^k(K)$ denotes the space of polynomials in K of degree at most k. It is easy to see that $\mathbf{V}_2^k \subset \mathbf{V}_1^k$. Two formulations are obtained if $\mathbf{V}^k$ in (32) is specified as follows:

- *Formulation I*: $\mathbf{V}^k = \mathbf{V}_1^k$. A single polynomial $\varphi \in P^{k+1}(K)$, up to a constant, is recovered from $\mathbf{w}$ in each element by the following least square procedure

$$\left\| \sum_i (\varphi_{x_i} - w_i)^2 \right\|_{L^1(K)} = \min_{\psi \in P^{k+1}(K)} \left\| \sum_i (\psi_{x_i} - w_i)^2 \right\|_{L^1(K)} \qquad (38)$$

  after each time stage. This is the method proposed by Hu and Shu in [18].
- *Formulation II*: $\mathbf{V}^k = \mathbf{V}_2^k$.

We have proven in [27] that the two formulations are mathematically equivalent. Clearly, the second formulation has several advantages over the first formulation:

(1) Formulation II allows the method of lines version of the scheme, while Formulation I does not have a method of lines version due to the least square procedure which is applied after each time step or stage. The method of lines version allows more natural and direct analysis for stability and accuracy of discontinuous Galerkin methods, e.g. the results in [24].

(2) The implementation of the algorithm is significantly simplified by using Formulation II since a smaller solution space is used and the least square procedure is completely avoided. If we characterize the computational cost of (32) per time step per element simply by the dimension of $\mathbf{V}^k|_K$, we can get

$$n_1 = dim(\mathbf{V}_1^k|_K) = n \sum_{r=0}^{k} C_{r+n-1}^{n-1}, \quad n_2 = dim(\mathbf{V}_2^k|_K) = \sum_{r=1}^{k+1} C_{r+n-1}^{n-1}.$$

For example, for the two dimensional case $n = 2$, $n_1 = (k+2)(k+1)$, $n_2 = \frac{(k+4)(k+1)}{2}$, hence $\frac{n_2}{n_1} \to \frac{1}{2}$ as $k \to \infty$; i.e. the cost is reduced to about half for higher order schemes. For the three dimensional case $n = 3$, $n_1 = \frac{k^3+6k^2+11k+6}{2}$, $n_2 = \frac{(k+1)(k^2+8k+18)}{6}$, hence $\frac{n_2}{n_1} \to \frac{1}{3}$ as $k \to \infty$; i.e. the cost is reduced to about one third for higher order schemes.

Representative numerical examples using the discontinuous Galerkin methods for solving the two dimensional Hamilton-Jacobi equations (4) will be given in section 7. More numerical examples can be found in [18,24,27].

6. High Order Strong Stability Preserving Runge-Kutta Time Discretizations

For all of the spatial discretizations discussed in the previous sections, the time variable t is left undiscretized. A popular time discretization method is the class of strong stability preserving (SSP), also referred to as total variation diminishing (TVD), high order Runge-Kutta time discretizations, see [37,35,15,16].

We start with the following ordinary differential equation (ODE)

$$\frac{d}{dt}u(t) = L(u(t), t) \tag{39}$$

resulting from a method of lines spatial discretization of a time dependent partial differential equation, such as (17), (21), (22) or (32) in the previous sections. Here $u = u(t)$ is a (usually very long) vector and $L(u, t)$ depends on u either linearly or non-linearly. In many applications $L(u, t) = L(u)$ which does not explicitly depend on t. The starting point for the SSP method is an *assumption* that the first order Euler forward time discretization to (39):

$$u^{n+1} = u^n + \Delta t L(u^n, t^n), \tag{40}$$

where u^n is an approximation to $u(t^n)$, are stable under a certain (semi) norm

$$||u^{n+1}|| \leq ||u^n|| \tag{41}$$

with a suitable time step restriction

$$\Delta t \leq \Delta t_0, \tag{42}$$

which typically depends on the spatial discretization mesh size. With this assumption, we would like to find SSP time discretization methods to (39), that are higher order accurate in time, yet still maintain the same stability condition (41). This might require a different restriction on the time step Δt than that in (42) of the form

$$\Delta t \leq c\Delta t_0, \tag{43}$$

where c is called the *CFL coefficient* of the SSP method. The objective is to find such methods with simple format, low computational cost and least restriction on the time step Δt, i.e. larger CFL coefficient c.

We remark that the strong stability assumption for the forward Euler step in (41) can be relaxed to the more general stability assumption

$$||u^{n+1}|| \leq (1 + O(\Delta t))||u^n||.$$

This general stability property is also preserved by the high order SSP time discretizations.

Runge-Kutta methods are time discretizations which can be written in several different ways. In [37], a general m stage Runge-Kutta method for (39) is written in the form:

$$\begin{aligned} u^{(0)} &= u^n, \\ u^{(i)} &= \sum_{k=0}^{i-1} \left(\alpha_{i,k} u^{(k)} + \Delta t \beta_{i,k} L(u^{(k)}, t^n + d_k \Delta t) \right), \quad i = 1, ..., m \\ u^{n+1} &= u^{(m)} \end{aligned} \tag{44}$$

where d_k are related to $\alpha_{i,k}$ and $\beta_{i,k}$ by

$$d_0 = 0, \qquad d_i = \sum_{k=0}^{i-1} (\alpha_{i,k} d_k + \beta_{i,k}), \qquad i = 1, ..., m-1.$$

Thus, we do not need to discuss the choice of d_k separately. In most ODE literatures, e.g. [7], a Runge-Kutta method is written in the form of a Butcher array. Every Runge-Kutta method in the form of (44) can be easily converted in a unique way into a Butcher array, see [37]. A Runge-Kutta method written in a Butcher array can also be rewritten into the form (44), however this conversion is in general *not* unique. This non-uniqueness in the representation (44) is exploited in the literature to seek the largest provable time steps (43) for SSP.

We always need and require that $\alpha_{i,k} \geq 0$ in (44). If this is violated no SSP methods are possible. Basically, we rely heavily on convexity arguments which would require that all $\alpha_{i,k}$'s to be non-negative.

If all the $\beta_{i,k}$'s in (44) are also nonnegative, $\beta_{i,k} \geq 0$, we have the following simple lemma, which is the backbone of SSP Runge-Kutta methods:

Lemma 4: [37] *If the forward Euler method* (40) *is stable in the sense of* (41) *under the time step restriction* (42)*, then the Runge-Kutta method* (44) *with* $\alpha_{i,k} \geq 0$ *and* $\beta_{i,k} \geq 0$ *is SSP, i.e. its solution also satisfies the same stability* (41) *under the time step restriction* (43) *with the CFL coefficient*

$$c = \min_{i,k} \frac{\alpha_{i,k}}{\beta_{i,k}}. \tag{45}$$

The most popular and successful SSP methods are those covered by Lemma 4. We will only give examples of SSP methods covered by Lemma 4 in this section. If some of the $\beta_{i,k}$'s must be negative because of accuracy constraints, there is also a way to obtain SSP methods, see [37,15,16] for details.

We list below a few popular SSP Runge-Kutta methods:

(1) A second order SSP Runge-Kutta method [37]:

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n, t^n) \\ u^{n+1} &= \frac{1}{2}u^n + \frac{1}{2}u^{(1)} + \frac{1}{2}\Delta t L(u^{(1)}, t^n + \Delta t) \end{aligned} \tag{46}$$

with a CFL coefficient $c = 1$ in (43). This is just the classical Heun or modified Euler method.

(2) A third order SSP Runge-Kutta method [37]:

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n, t^n) \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}, t^n + \Delta t) \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}, t^n + \frac{1}{2}\Delta t), \end{aligned} \tag{47}$$

with a CFL coefficient $c = 1$ in (43).

(3) A third order low storage SSP Runge-Kutta method [15]:

$$\begin{aligned} u^{(0)} &= u^n, \qquad du^{(0)} = 0, \\ du^{(i)} &= A_i du^{(i-1)} + \Delta t L(u^{(i-1)}, t^n + d_{i-1}\Delta t), \qquad i = 1, \ldots, 3, \\ u^{(i)} &= u^{(i-1)} + B_i du^{(i)}, \qquad\qquad i = 1, \ldots, 3, \\ u^{n+1} &= u^{(3)}. \end{aligned} \tag{48}$$

with

$$
\begin{aligned}
z_1 &= \sqrt{36b^4 + 36b^3 - 135b^2 + 84b - 12} \\
z_2 &= 2b^2 + b - 2 \\
z_3 &= 12b^4 - 18b^3 + 18b^2 - 11b + 2 \\
z_4 &= 36b^4 - 36b^3 + 13b^2 - 8b + 4 \\
z_5 &= 69b^3 - 62b^2 + 28b - 8 \\
z_6 &= 34b^4 - 46b^3 + 34b^2 - 13b + 2 \\
d_0 &= 0 \\
A_1 &= 0 \\
B_1 &= b \\
d_1 &= B_1 \\
A_2 &= \frac{-z_1(6b - 4b + 1) + 3z_3}{(2b+1)z_1 - 3(b+2)(2b-1)^2} \\
B_2 &= \frac{12b(b-1)(3z_2 - z_1) - (3z_2 - z_1)^2}{144b(3b-2)(b-1)^2} \\
d_2 &= B_1 + B_2 + B_2 A_2 \\
A_3 &= \frac{-z_1 z_4 + 108(2b-1)b^5 - 3(2b-1)z_5}{24z_1 b(b-1)^4 + 72bz_6 + 72b^6(2b-13)} \\
B_3 &= \frac{-24(3b-2)(b-1)^2}{(3z_2 - z_1)^2 - 12b(b-1)(3z_2 - z_1)}
\end{aligned}
$$

where $b = 0.924574$, with a CFL coefficient $c = 0.32$ in (43). Only u and du must be stored, resulting in two storage units for each variable. This method can be used when storage is a paramount consideration, such as in large scale three dimensional calculations.

(4) A fourth order, five stage SSP Runge-Kutta method. It can be proven [15] that all four stage, fourth order SSP Runge-Kutta scheme (44) with a nonzero CFL coefficient c in (43) must have at least one negative $\beta_{i,k}$. To obtain fourth order SSP Runge-Kutta methods with nonnegative $\beta_{i,k}$ covered by Lemma 4, we would need at least five stages. The following is a five stage, fourth order SSP Runge-Kutta method [40] with

a CFL coefficient $c = 1.508$ in (43):

$$\begin{aligned}
u^{(1)} &= u^n + 0.39175222700392\,\Delta t L(u^n, t^n)\\
u^{(2)} &= 0.44437049406734\,u^n + 0.55562950593266\,u^{(1)}\\
&\quad +0.36841059262959\,\Delta t L(u^{(1)}, t^n + 0.39175222700392\,\Delta t)\\
u^{(3)} &= 0.62010185138540\,u^n + 0.37989814861460\,u^{(2)}\\
&\quad +0.25189177424738\,\Delta t L(u^{(2)}, t^n + 0.58607968896780\,\Delta t)\\
u^{(4)} &= 0.17807995410773\,u^n + 0.82192004589227\,u^{(3)} \qquad (49)\\
&\quad +0.54497475021237\,\Delta t L(u^{(3)}, t^n + 0.47454236302687\,\Delta t)\\
u^{n+1} &= 0.00683325884039\,u^n + 0.51723167208978\,u^{(2)}\\
&\quad +0.12759831133288\,u^{(3)}\\
&\quad +0.08460416338212\,\Delta t L(u^{(3)}, t^n + 0.47454236302687\,\Delta t)\\
&\quad +0.34833675773694\,u^{(4)}\\
&\quad +0.22600748319395\,\Delta t L(u^{(4)}, t^n + 0.93501063100924\,\Delta t).
\end{aligned}$$

7. A Few Numerical Examples

We will show a few numerical examples simulated by the discontinuous Galerkin method in section 5 [18] as representatives. Other examples can be found in the references listed in each sections for different numerical methods discussed in these notes.

Example 5: Two dimensional Burgers' equation:

$$\begin{cases} \varphi_t + \frac{(\varphi_x+\varphi_y+1)^2}{2} = 0, & -2 < x < 2,\ -2 < y < 2 \\ \varphi(x,y,0) = -\cos\left(\frac{\pi(x+y)}{2}\right) \end{cases} \qquad (50)$$

with periodic boundary conditions.

At $t = 0.5/\pi^2$, the solution is still smooth. We use non-uniform rectangular meshes obtained from the tensor product of one dimensional nonuniform meshes via randomly shifting the cell boundaries in a uniform mesh in the range $[-0.1h, 0.1h]$ (the meshes in two directions are independent). The L^2-errors computed by a 6×6 point Gaussian quadrature in each cell are shown in Table 1.

At $t = 1.5/\pi^2$, the solution has discontinuous derivatives. Figure 3 is the graph of the numerical solution with 40×40 elements (uniform mesh).

Finally we use triangle based triangulation, the mesh with $h = \frac{1}{4}$ is shown in Figure 4. The accuracy at $t = 0.5/\pi^2$ is shown in Table 2. Similar

Table 1. Accuracy for 2D Burgers equation, non-uniform rectangular mesh, $t = 0.5/\pi^2$.

	P^1		P^2		P^3	
$N \times N$	L^2 error	order	L^2 error	order	L^2 error	order
10×10	4.47E-01	—	6.28E-02	—	1.61E-02	—
20×20	1.83E-01	1.288	1.50E-02	2.066	2.06E-03	2.966
40×40	8.01E-02	1.192	3.63E-03	2.047	3.48E-04	2.565
80×80	3.82E-02	1.068	9.17E-04	1.985	6.03E-05	2.529
160×160	1.87E-02	1.031	2.34E-04	1.970	8.58E-06	2.813

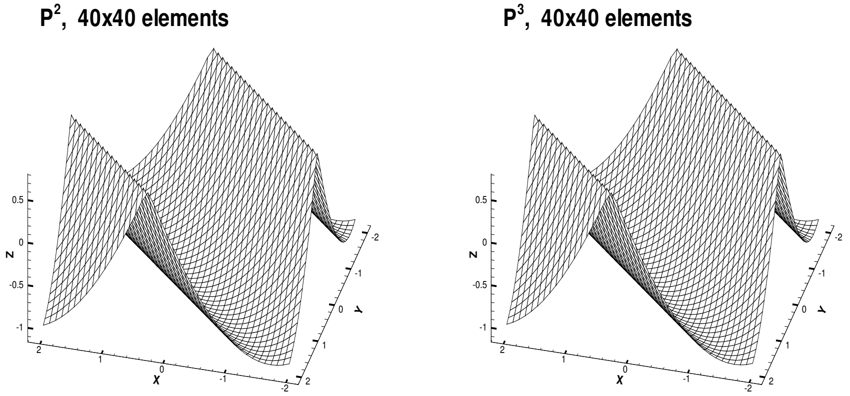


Fig. 3. Two dimension Burgers' equation, rectangular mesh, t=1.5/π^2.

accuracy pattern is observed as in the rectangular case. The result at $t = 1.5/\pi^2$, when the derivative is discontinuous, is shown in Figure 5.

Table 2. Accuracy for 2D Burgers equation, triangular mesh as those in Figure 4, $t = 0.5/\pi^2$.

	P^2		P^3	
h	L^1 error	order	L^1 error	order
1	5.48E-02	—	1.17E-02	—
1/2	1.35E-02	2.02	1.35E-03	3.12
1/4	2.94E-03	2.20	1.45E-04	3.22
1/8	6.68E-04	2.14	1.71E-05	3.08

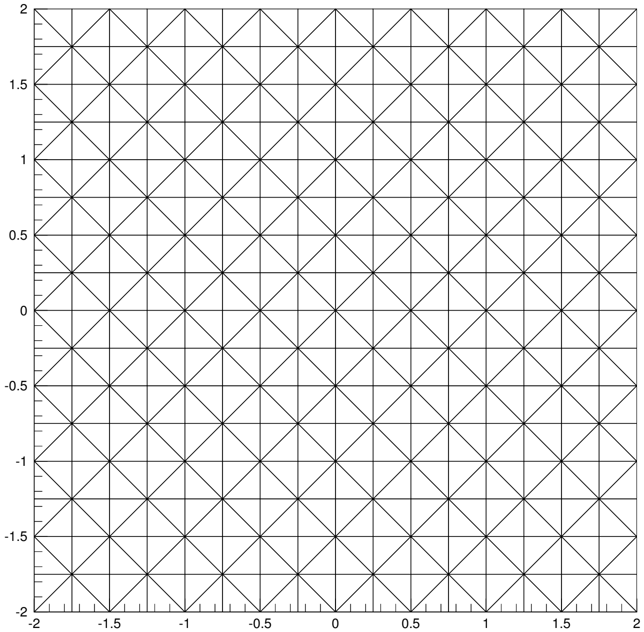


Fig. 4. Triangulation for two dimensional Burgers equation, $h = \frac{1}{4}$.

Example 6: The level set equation in a domain with a hole:

$$\begin{cases} \varphi_t + sign(\varphi_0)(\sqrt{\varphi_x^2 + \varphi_y^2} - 1) = 0, & \frac{1}{2} < \sqrt{x^2 + y^2} < 1 \\ \varphi(x, y, 0) = \varphi_0(x, y) \end{cases} \tag{51}$$

This problem is introduced in [41]. The solution φ to (51) has the same zero level set as φ_0, and the steady state solution is the distance function to that zero level curve. We use this problem to test the effects using various integration paths (35) when there is a hole in the region. Notice that the exact steady state solution is the distance function to the inner boundary of domain when boundary condition is adequately prescribed. We compute the time dependent problem to reach a steady state solution, using the exact solution for the boundary conditions of φ_x and φ_y. Four symmetric elements near the outer boundary are updated by (34), all other elements are recovered from (35) by the shortest path to the nearest one of above four elements. The results are shown in Table 3. Also shown in Table 3 is the error (difference) between the numerical solution φ thus recovered, and

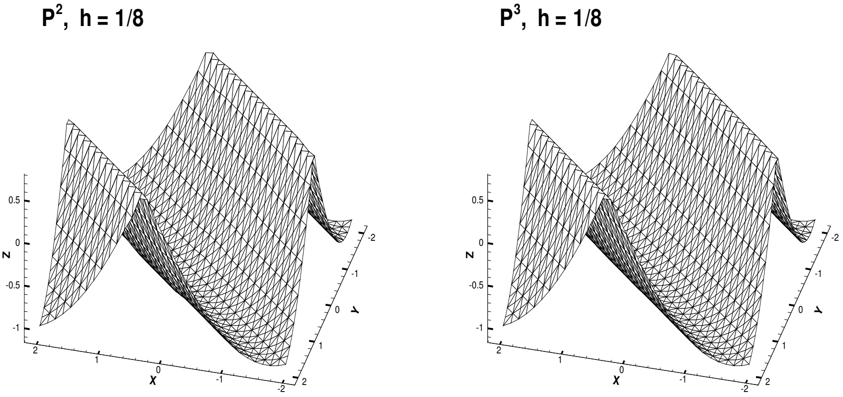


Fig. 5. Two dimension Burgers' equation, triangular mesh, t=1.5/π^2.

the value of φ after another integration along a circular path (starting and ending at the same point in (35)). We can see that the difference is small with the correct order of accuracy, further indicating that the dependency of the recovered solution φ on the integration path is on the order of the truncation errors even for such problems with holes. Finally, the mesh with 1432 triangles and the solution with 5608 triangles are shown in Figure 6.

Table 3. Errors for the level set equation, triangular mesh with P^2.

	Errors for the Solution		Errors by Integration Path	
N	L^1 error	order	L^1 error	order
403	1.02E-03	—	1.61E-04	—
1432	1.23E-04	3.05	5.84E-05	1.46
5608	1.71E-05	2.85	9.32E-06	2.65
22238	2.09E-06	3.03	1.43E-06	2.70

Example 7: The problem of a propagating surface:

$$\begin{cases} \varphi_t - (1-\varepsilon K)\sqrt{1+\varphi_x^2+\varphi_y^2} = 0, & 0 < x < 1,\, 0 < y < 1 \\ \varphi(x,y,0) = 1 - \frac{1}{4}(\cos(2\pi x - 1))\,(\cos(2\pi y - 1)) & \end{cases} \tag{52}$$

where K is the mean curvature defined by

$$K = -\frac{\varphi_{xx}(1+\varphi_y^2) - 2\varphi_{xy}\varphi_x\varphi_y + \varphi_{yy}(1+\varphi_x^2)}{(1+\varphi_x^2+\varphi_y^2)^{\frac{3}{2}}}, \tag{53}$$

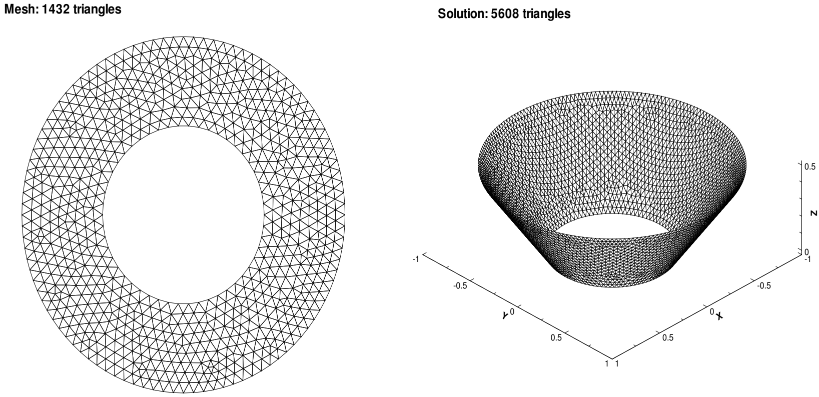


Fig. 6. The level set equation, P^2.

and ε is a small constant. Periodic boundary condition is used.

We apply the discontinuous Galerkin method, with the second derivative terms handled by the local discontinuous Galerkin techniques presented and analyzed in [11], which amounts to solving the following system

$$\begin{cases} u_t - \left(\sqrt{1+u^2+v^2} + \varepsilon \frac{p(1+v^2)-2quv+r(1+u^2)}{1+u^2+v^2}\right)_x = 0 \\ v_t - \left(\sqrt{1+u^2+v^2} + \varepsilon \frac{p(1+v^2)-2quv+r(1+u^2)}{1+u^2+v^2}\right)_y = 0 \\ p - u_x = 0 \\ q - u_y = 0 \\ r - v_y = 0 \end{cases} \tag{54}$$

using the discontinuous Galerkin method. The details of the method, especially the choices of fluxes, which are important for stability, can be found in [11].

We use a triangulation shown in Figure 7. We refine the mesh around the center of domain where the solution develops discontinuous derivatives (for the $\varepsilon = 0$ case). There are 2146 triangles and 1108 nodes in this triangulation. The solutions are displayed in Figure 8 and Figure 9, respectively, for $\varepsilon = 0$ (pure convection) and $\varepsilon = 0.1$. Notice that we shift the solution at $t = 0.0$ downward by 0.35 to show the detail of the solutions at later time.

Example 8: The problem of a propagating surface on a unit disk. The

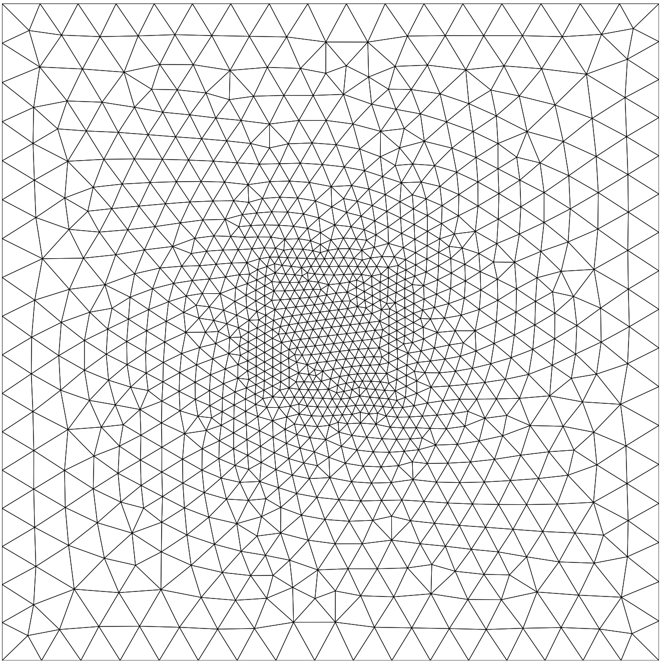


Fig. 7. Triangulation used for the propagating surfaces.

equation is the same as (52) in the previous example, but it is solved on a unit disk $x^2 + y^2 < 1$ with an initial condition

$$\varphi(x, y, 0) = \sin\left(\frac{\pi(x^2 + y^2)}{2}\right)$$

and a Neumann type boundary condition $\nabla\varphi = 0$.

It is difficult to use rectangular meshes for this problem. Instead we use the triangulation shown in Figure 10. Notice that we have again refined the mesh near the center of the domain where the solution develops discontinuous derivatives. There are 1792 triangles and 922 nodes in this triangulation. The solutions with $\varepsilon = 0$ are displayed in Figure 11. Notice that the solution at $t = 0$ is shifted downward by 0.2 to show the detail of the solution at later time.

The solution with $\varepsilon = 0.1$ are displayed in Figure 12. Notice that the solution at $t = 0$ is again shifted downward by 0.2 to show the detail of the solution at later time.

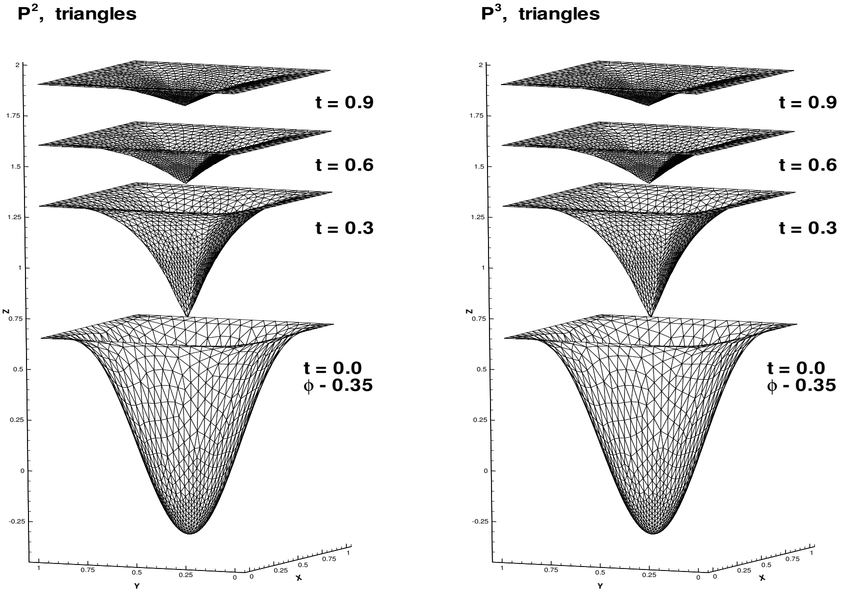


Fig. 8. Propagating surfaces, triangular mesh, $\varepsilon = 0$.

Example 9: A problem from optimal control [32]:

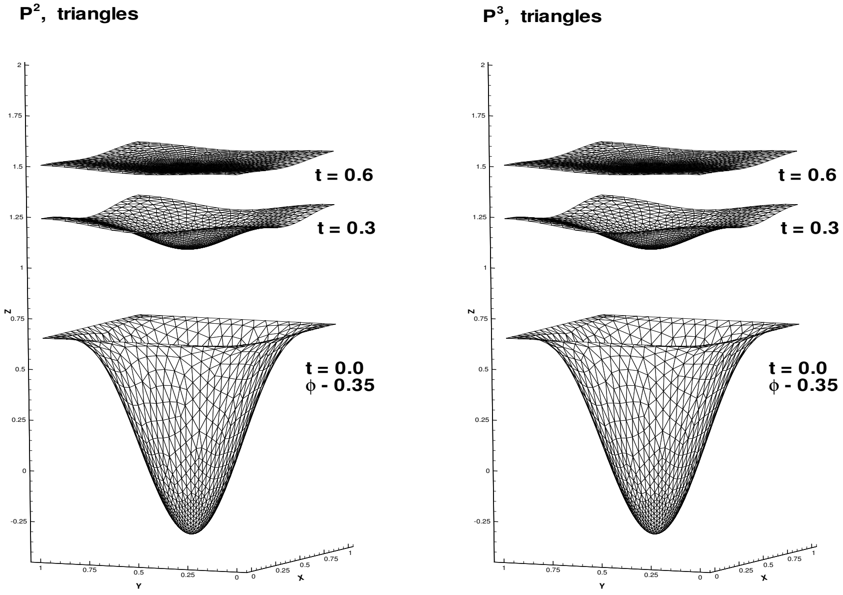
$$\begin{cases} \varphi_t + (\sin y)\varphi_x + (\sin x + \text{sign}(\varphi_y))\varphi_y - \frac{1}{2}\sin^2 y - (1 - \cos x) = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad -\pi < x < \pi,\ -\pi < y < \pi \\ \varphi(x, y, 0) = 0 \end{cases} \tag{55}$$

with periodic boundary conditions. We use a uniform rectangular mesh of 40×40 elements. The solution at $t = 1$ is shown in Figure 13, while the optimal control $w = \text{sign}(\varphi_y)$ is shown in Figure 14.

Notice that the discontinuous Galerkin method computes $\nabla\varphi$ as an independent variable. It is very desirable for those problems in which the most interesting features are contained in the first derivatives of φ, as in this optimal control problem.

Example 10: A problem from computer vision [33]:

$$\begin{cases} \varphi_t + I(x, y)\sqrt{1 + \varphi_x^2 + \varphi_y^2} - 1 = 0, \qquad -1 < x < 1, -1 < y < 1 \\ \varphi(x, y, 0) = 0 \end{cases} \tag{56}$$


P^2, triangles

P^3, triangles

Fig. 9. Propagating surfaces, triangular mesh, $\varepsilon = 0.1$.

with $\varphi = 0$ as the boundary condition. The steady state solution of this problem is the shape lighted by a source located at infinity with vertical direction. The solution is not unique if there are points at which $I(x, y) = 1$. Conditions must be prescribed at those points where $I(x, y) = 1$. Since our method is a finite element method, we need to prescribe suitable conditions at the correspondent elements. We take

$$I(x, y) = 1/\sqrt{1 + (1 - |x|)^2 + (1 - |y|)^2} \tag{57}$$

The exact steady solution is $\varphi(x, y, \infty) = (1 - |x|)(1 - |y|)$. We use a uniform rectangular mesh of 40×40 elements. We impose the exact boundary conditions for $u = \varphi_x, v = \varphi_y$ from the above exact steady solution, and take the exact value at one point (the lower left corner) to recover φ. The results for P^2 and P^3 are presented in Figure 15, while Figure 16 contains the history of iterations to the steady state.

Next we take

$$I(x, y) = 1/\sqrt{1 + 4y^2(1 - x^2)^2 + 4x^2(1 - y^2)^2} \tag{58}$$

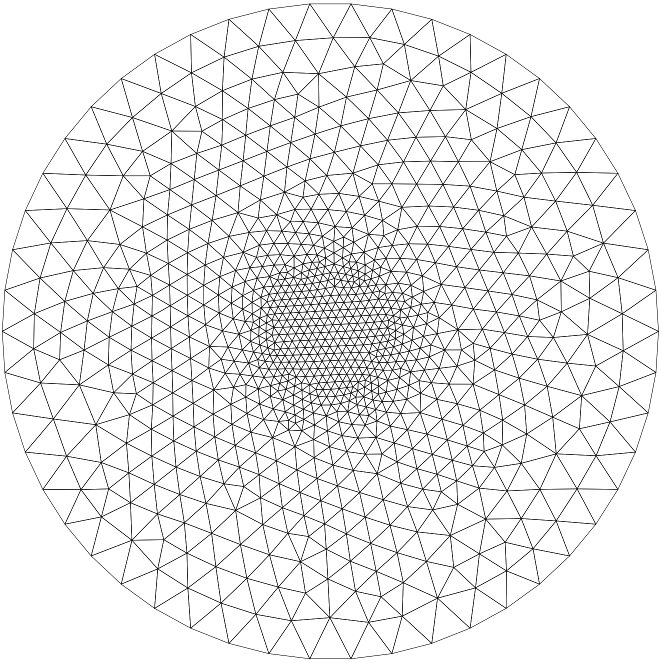


Fig. 10. Triangulation for the propagating surfaces on a disk.

The exact steady solution is $\varphi(x, y, \infty) = (1 - x^2)(1 - y^2)$. We again use a uniform rectangular mesh of 40×40 elements and impose the exact boundary conditions for $u = \varphi_x, v = \varphi_y$ from the above exact steady solution, and take the exact value at one point (the lower left corner) to recover φ. A continuation method is used, with the steady solution using

$$I_\varepsilon(x, y) = 1/\sqrt{1 + 4y^2(1 - x^2)^2 + 4x^2(1 - y^2)^2 + \varepsilon} \tag{59}$$

for bigger ε as the initial condition for smaller ε. The sequence of ε used are $\varepsilon = 0.2, 0.05, 0$. The results for P^2 and P^3 are presented in Figure 17.

8. Concluding Remarks

We have briefly surveyed the properties of Hamilton-Jacobi equations and a few numerical schemes for solving these equations. Because of space limitations, there are many related topics that we have not discussed, for example the class of central non-oscillatory schemes (e.g. [28]), techniques for efficiently solving steady state Hamilton-Jacobi equations, etc.

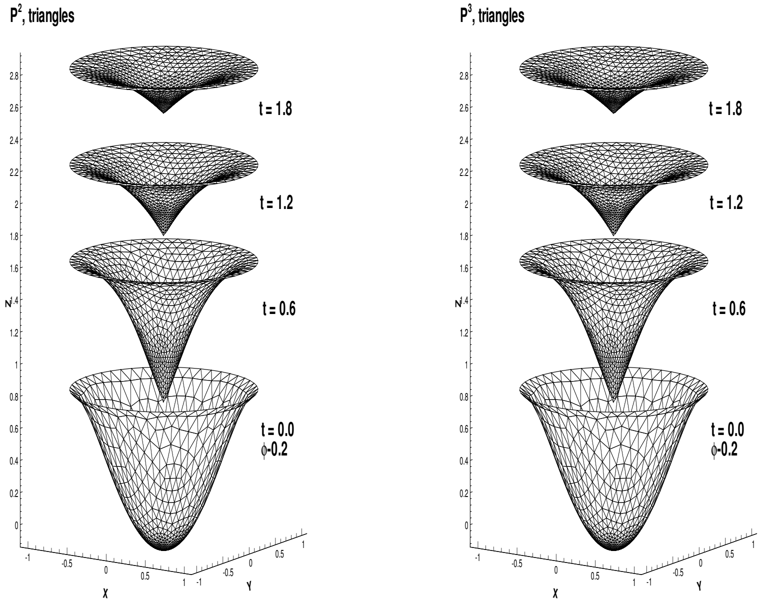


Fig. 11. Propagating surfaces on a disk, triangular mesh, $\varepsilon = 0$.

References

1. R. Abgrall, *On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation*, Journal of Computational Physics, 114 (1994), 45-54.
2. R. Abgrall, *Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes*, Communications on Pure and Applied Mathematics, 49 (1996), 1339-1373.
3. R. Abgrall and Th. Sonar, *On the use of Muehlbach expansions in the recovery step of ENO methods*, Numerische Mathematik, 76 (1997), 1-25.
4. S. Augoula and R. Abgrall, *High order numerical discretization for Hamilton-Jacobi equations on triangular meshes*, Journal of Scientific Computing, 15 (2000), 197-229.
5. M. Bardi and S. Osher, *The non-convex multi-dimensional Riemann problem for Hamilton-Jacobi equations*, SIAM Journal on Mathematical Analysis, 22 (1991), 344-351.
6. T. Barth and J. Sethian, *Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains*, Journal of Computational Physics, 145 (1998), 1-40.
7. J. C. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*, John Wiley, New York, 1987.

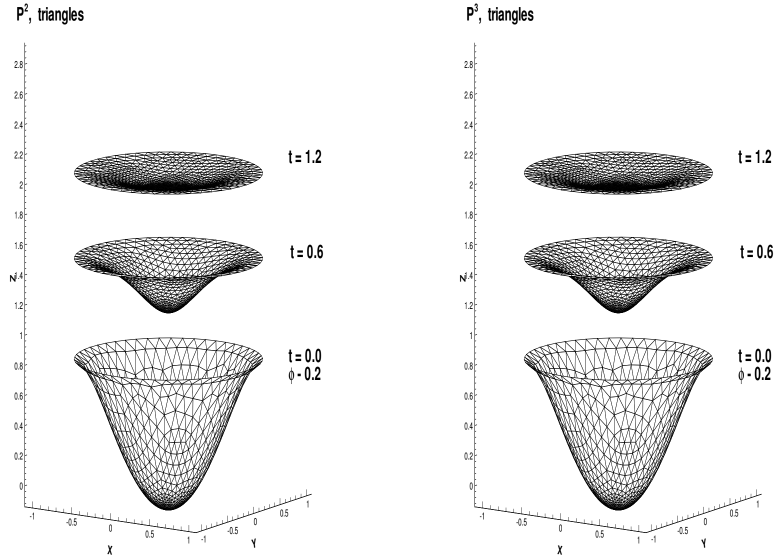


Fig. 12. Propagating surfaces on a disk, triangular mesh, $\varepsilon = 0.1$.

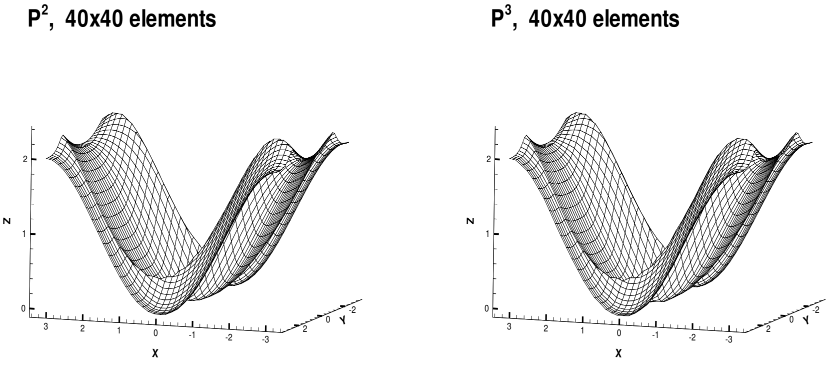
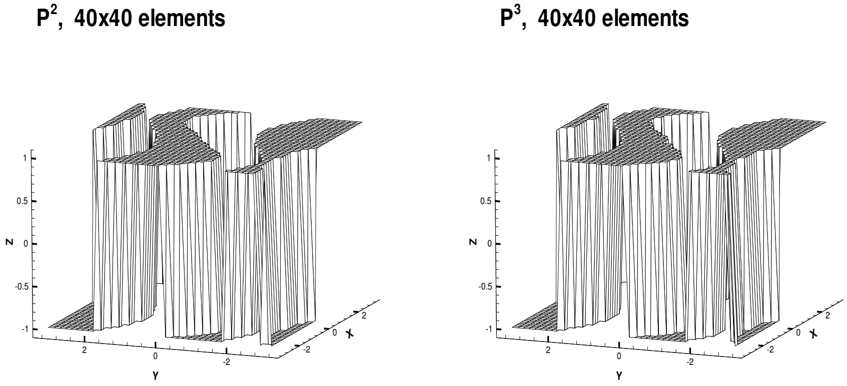


Fig. 13. Control problem, $t = 1$.

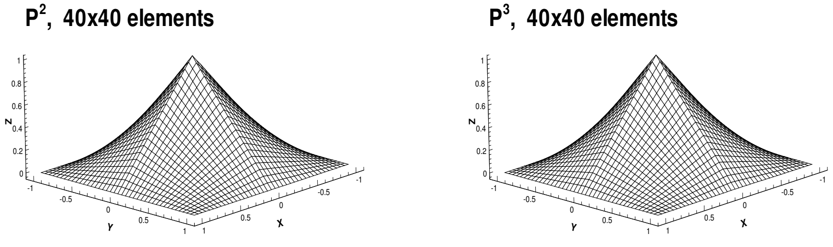
8. B. Cockburn, S. Hou and C.-W. Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case*, Mathematics of Computation, 54 (1990), 545-581.



P^2, 40x40 elements

P^3, 40x40 elements

Fig. 14. Control problem, $t = 1, w = \text{sign}(\varphi_y)$.



P^2, 40x40 elements

P^3, 40x40 elements

Fig. 15. Computer vision problem, $\varphi(x, y, \infty) = (1 - |x|)(1 - |y|)$.

9. B. Cockburn, F. Li and C.-W. Shu, *Locally divergence-free discontinuous Galerkin methods for the Maxwell equations*, Journal of Computational Physics, 194 (2004), 588-610.

10. B. Cockburn and C.-W. Shu, *The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*, Journal of Computational Physics, 141 (1998), 199-224.

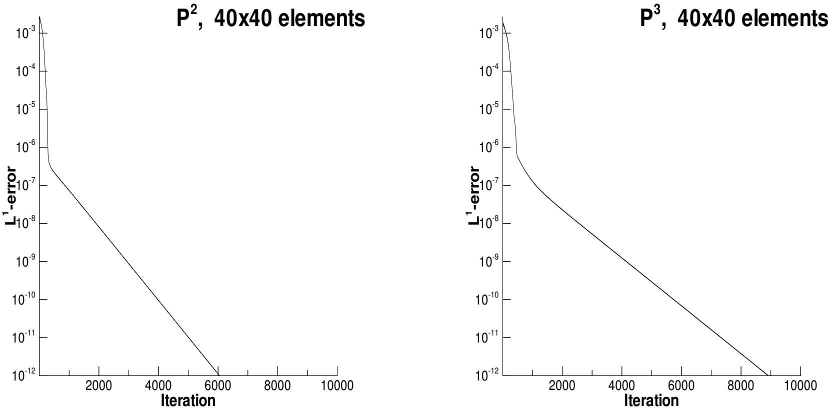


Fig. 16. Computer vision problem, history of iterations.

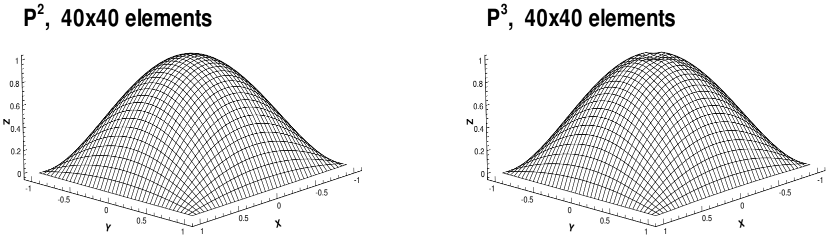


Fig. 17. Computer vision problem, $\varphi(x, y, \infty) = (1 - x^2)(1 - y^2)$.

11. B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM Journal on Numerical Analysis, 35 (1998), 2440-2463.

12. B. Cockburn and C.-W. Shu, *Runge-Kutta discontinuous Galerkin methods for convection-dominated problems*, Journal of Scientific Computing, 16 (2001), 173-261.

13. M. Crandall and P. L. Lions, *Viscosity solutions of Hamilton-Jacobi equations*, Transactions of American Mathematical Society, 277 (1983), 1-42.
14. M. Crandall and P. L. Lions, *Monotone difference approximations for scalar conservation laws*, Mathematics of Computation, 34 (1984), 1-19.
15. S. Gottlieb and C.-W. Shu, *Total variation diminishing Runge-Kutta schemes*, Mathematics of Computation, 67 (1998), 73-85.
16. S. Gottlieb, C.-W. Shu and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Review, 43 (2001), 89-112.
17. A. Harten, B. Engquist, S. Osher and S. Chakravathy, *Uniformly high order accurate essentially non-oscillatory schemes, III*, Journal of Computational Physics, 71 (1987), 231-303.
18. C. Hu and C.-W. Shu, *A discontinuous Galerkin finite element method for Hamilton-Jacobi equations*, SIAM Journal on Scientific Computing, 21 (1999), 666-690.
19. C. Hu and C.-W. Shu, *Weighted Essentially Non-Oscillatory Schemes on Triangular Meshes*, Journal of Computational Physics, 150 (1999), 97-127.
20. G. Jiang and D.-P. Peng, *Weighted ENO schemes for Hamilton-Jacobi equations*, SIAM Journal on Scientific Computing, 21 (2000), 2126-2143.
21. G. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics, 126 (1996), 202-228.
22. S. Jin and Z. Xin, *Numerical passage from systems of conservation laws to Hamilton-Jacobi equations and relaxation schemes*, SIAM Journal on Numerical Analysis, 35 (1998), 2385-2404.
23. P. D. Lax, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, SIAM Regional Conference series in Applied Mathematics, SIAM, Philadelphia, 1973.
24. O. Lepsky, C. Hu and C.-W. Shu, *Analysis of the discontinuous Galerkin method for Hamilton-Jacobi equations*, Applied Numerical Mathematics, 33 (2000), 423-434.
25. R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhauser Verlag, Basel, 1992.
26. F. Li and C.-W. Shu, *Locally divergence-free discontinuous Galerkin methods for MHD equations*, Journal of Scientific Computing, 22-23 (2005), 413-442.
27. F. Li and C.-W. Shu, *Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations*, Applied Mathematics Letters, 18 (2005), 1204-1209.
28. C.-T. Lin and E. Tadmor, *High-resolution non-oscillatory central schemes for approximate Hamilton-Jacobi equations*, SIAM Journal on Scientific Computing, 21 (2000), 2163-2186.
29. P. L. Lions, *Generalized Solutions of Hamilton-Jacobi Equations*, Pitman, Boston, 1982.
30. X.-D. Liu, S. Osher and T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics, 115 (1994), 200-212.
31. S. Osher and J. Sethian, *Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations*, Journal of Computational Physics, 79 (1988), 12-49.

32. S. Osher and C.-W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SIAM Journal on Numerical Analysis, 28 (1991), 907-922.
33. E. Rouy and A. Tourin, *A viscosity solutions approach to shape-from-shading*, SIAM Journal on Numerical Analysis, 29 (1992), 867-884.
34. J. Shi, C. Hu and C.-W. Shu, *A technique of treating negative weights in WENO schemes*, Journal of Computational Physics, 175 (2002), 108-127.
35. C.-W. Shu, *Total-Variation-Diminishing time discretizations*, SIAM Journal on Scientific and Statistical Computing, 9 (1988), 1073-1084.
36. C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), Lecture Notes in Mathematics, volume 1697, Springer, Berlin, 1998, 325-432.
37. C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes*, Journal of Computational Physics, 77 (1988), 439-471.
38. C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes II*, Journal of Computational Physics, 83 (1989), 32-78.
39. J. Smoller, *Shock Waves and Reaction-Diffusion Equations*, Springer-Verlag, New York, 1983.
40. R. Spiteri and S. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM Journal on Numerical Analysis, 40 (2002), 469-491.
41. M. Sussman, P. Smereka and S. Osher, *A level set approach for computing solution to incompressible two-phase flow*, Journal of Computational Physics, 114 (1994), 146-159.
42. Y.-T. Zhang and C.-W. Shu, *High order WENO schemes for Hamilton-Jacobi equations on triangular meshes*, SIAM Journal on Scientific Computing, 24 (2003), 1005-1030.

THEORY AND COMPUTATION OF VARIATIONAL IMAGE DEBLURRING

Tony F. Chan

Department of Mathematics, UCLA
Los Angeles, CA 90095, USA
E-mail: TonyC@college.ucla.edu

Jianhong Shen

School of Mathematics
University of Minnesota
Minneapolis, MN 55455, USA
E-mail: jhshen@math.umn.edu

To recover a sharp image from its blurry observation is the problem known as *image deblurring*. It frequently arises in imaging sciences and technologies, including optical, medical, and astronomical applications, and is crucial for allowing to detect important features and patterns such as those of a distant planet or some microscopic tissue.

Mathematically, image deblurring is intimately connected to backward diffusion processes (e.g., inverting the heat equation), which are notoriously unstable. As inverse problem solvers, deblurring models therefore crucially depend upon proper regularizers or conditioners that help secure stability, often at the necessary cost of losing certain high-frequency details in the original images. Such regularization techniques can ensure the existence, uniqueness, or stability of deblurred images.

The present work follows closely the general framework described in our recent monograph [18], but also contains more updated views and approaches to image deblurring, including, e.g., more discussion on stochastic signals, the Bayesian/Tikhonov approach to Wiener filtering, and the iterated-shrinkage algorithm of Daubechies et al. [30,31] for wavelet-based deblurring. The work thus contributes to the development of generic, systematic, and unified frameworks in contemporary image processing.

1. Mathematical Models of Blurs

Throughout the current work, an image u is identified with a Lebesgue measurable real function on an open two-dimensional (2D) regular domain Ω. A general point $\boldsymbol{x} = (x_1, x_2) \in \Omega$ shall also be called a *pixel* as in digital image processing. The framework herein applies readily to color images for which u could be considered an RGB-vectorial function.

1.1. *Linear blurs*

Deblurring is to undo the blurring process applied to a sharp and clear image earlier, and is thus an inverse problem. We hence start with the description of the forward problem - mathematical models of blurring.

In most applications, blurs are introduced by three different types of physical factors: optical, mechanical, or medium-induced, which could lead to familiar out-of-focus blurs, motion blurs, or atmospheric blurs respectively. We refer the reader to [18] for a more detailed account on the associated physical processes. Figures 1 and 2 show two real blur examples directly taken by a digital camera under different circumstances.

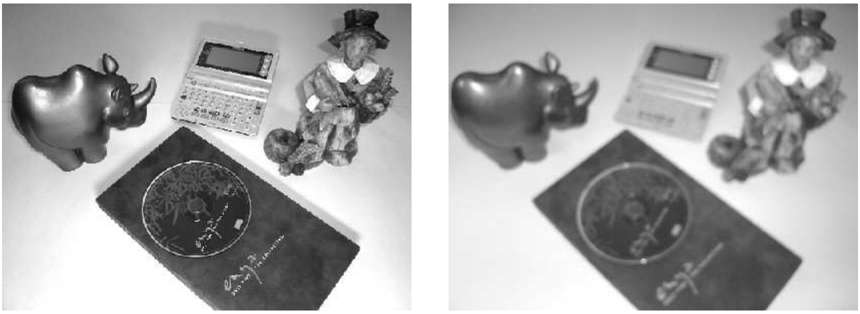


Fig. 1. A real example of an out-of-focus blur. Left: the clear image; Right: the out-of-focus image taken by a digital camera that focuses on a point closer than the scene.

Mathematically, blurring can be either *linear* or *nonlinear*. The latter is more challenging to invert due to the scarcity of proper nonlinear models. The current work shall mainly focus on linear deblurring problems.

A general linear blur $u_0 = K[u]$ is defined by a linear operator K. In most applications noise is unavoidable and a real observation is thus often modelled by

$$u_0 = K[u] + n,$$

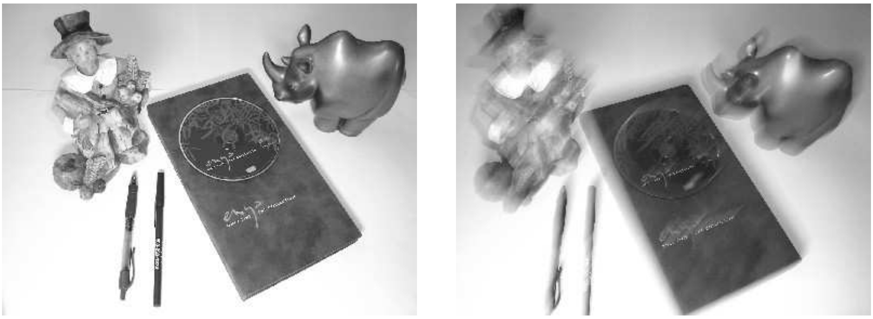


Fig. 2. A real example of a motion blur. Left: the clear image; Right: the motion-blurred image taken by a camera that experiences a rapid jitter during the exposure.

provided that the noise n is additive. (Multiplicative noises can be handled similarly.)

Among all linear blurs, the most frequently encountered type is *shift-invariant.* A linear blur K is said to be shift-invariant if for any shift $\boldsymbol{a} \in \mathbb{R}^2$,

$$u_0(\boldsymbol{x}) = K[u(\boldsymbol{x})] \quad \text{implies that} \quad u_0(\boldsymbol{x} - \boldsymbol{a}) = K[u(\boldsymbol{x} - \boldsymbol{a})].$$

It is well known in signal processing as well as system theory [56] that a shift-invariant linear operator must be in the form of convolution:

$$K[u] = k * u(\boldsymbol{x}) = \int_{\mathbb{R}^2} k(\boldsymbol{x} - \boldsymbol{y}) u(\boldsymbol{y}) d\boldsymbol{y}, \tag{1}$$

for some suitable kernel function $k(\boldsymbol{x})$, or the *point spread function* (PSF).

At any fixed pixel $\boldsymbol{x} \in \Omega$, a general linear blur K induces a linear functional on u, or a generalized function $L_{\boldsymbol{x}} : u \to K[u](\boldsymbol{x})$. Denote it symbolically by $k(\boldsymbol{x}, \cdot)$ so that as in distribution theory [68], one has

$$L_{\boldsymbol{x}}[u] = \langle k(\boldsymbol{x}, \cdot), u(\cdot) \rangle.$$

Suppose that the distribution $k(\boldsymbol{x}, \cdot)$ is actually an ordinary measurable function in $L^1(\Omega)$. Then the linear blur becomes ordinary integrals:

$$u_0(\boldsymbol{x}) = \int_{\Omega} k(\boldsymbol{x}, \boldsymbol{y}) u(\boldsymbol{y}) d\boldsymbol{y}, \qquad \boldsymbol{x} \in \Omega.$$

Herein we shall assume that the image u belongs to $L^p(\Omega)$ with $p \in [1, +\infty]$, and that K is a bounded linear operator from $L^p(\Omega)$ to $L^q(\Omega)$ with some $q \in [1, +\infty]$. As a result, the adjoint K^* is defined from $(L^q)^*$ to $(L^p)^*$, the dual spaces. (One must be aware, however, that $(L^\infty)^* \neq L^1$ [48].)

1.2. *The DC-condition*

The most outstanding characteristic of a blur operator is the DC-condition:

$$K[1] = 1, \qquad \text{treating} \quad 1 \in L^\infty(\Omega). \tag{2}$$

In classical signal processing [56], DC stands for *direct current* since the Fourier transform of a constant contains no oscillatory frequencies. By duality, $\langle K[u], v\rangle = \langle u, K^*[v]\rangle$, and the DC-condition on K amounts to the *mean-preserving condition* on K^*:

$$\langle K^*[v]\rangle = \langle v\rangle, \quad \text{by setting } u = 1; \quad \text{or} \quad \int_\Omega K^*[v](\boldsymbol{x})d\boldsymbol{x} = \int_\Omega v(\boldsymbol{x})d\boldsymbol{x}, \tag{3}$$

if both v and $K^*[v]$ belong to $L^1(\Omega)$.

In terms of information theory [27], the DC condition implies that constant signals are *invariant* under blurring. In particular, blurs cannot *generate* ripples from flat signals, and thus can never *create* information.

When the blur is shift-invariant with a PSF k, the DC-condition requires

$$\int_{\mathbb{R}^2} k(\boldsymbol{x})d\boldsymbol{x} = 1, \quad \text{or in terms of its Fourier transform,} \quad K(\boldsymbol{\omega} = 0) = 1,$$

since the adjoint is also shift-invariant with PSF $k(-x)$. Moreover, a more convincing blur operator has to be *lowpass* [56,67], i.e., $K(\boldsymbol{\omega})$ must decay rapidly at high frequencies.

1.3. *Nonlinear blurs*

Blurs could be nonlinear, though linear models prevail in the literature. Consider for example the following nonlinear diffusion model:

$$v_t = \nabla \cdot \left[\frac{1}{\sqrt{1 + |\nabla v|^2}} \nabla v \right], \qquad v\big|_{t=0} = u(\boldsymbol{x}). \tag{4}$$

Let the solution be denoted by $v(\boldsymbol{x}, t)$. For any fixed finite time $T > 0$, define a nonlinear operator $K = K_T$ by: $u_0 = K[u] = v(\boldsymbol{x}, T)$. Nonlinearity is evident since for example $K[\lambda u] \neq \lambda K[u]$ for general u and $\lambda \neq 0$. But the operator K apparently satisfies the DC-condition. Furthermore, (4) is the gradient descent equation of the minimum surface energy

$$E[v] = \int_{\mathbb{R}^2} \sqrt{1 + |\nabla v|^2} d\boldsymbol{x}.$$

As a result, the above nonlinear diffusion model always smoothens out any rough initial surfaces. In particular, small scale features and oscillations of u

must be wiped out in $u_0 = K[u]$, making u_0 a visually blurred and mollified version of the original image u. Notice remarkably that the nonlinear blur is in fact shift-invariant.

2. Illposedness of Deblurring

The illposedness of deblurring could be readily understood in four intriguing aspects. Understanding the root and nature of illposedness helps one design good deblurring models. The following four viewpoints are in some sense the four different facets of a same phenomenon, and hence must not be taken individually.

A. Deblurring is Inverting Lowpass Filtering. In the Fourier domain, a blur operator is often *lowpass* so that high frequency details are compressed by vanishing multipliers. As a result, to deblur a blurry image, one has to multiply approximately the reciprocals of the vanishing multipliers, which is conceivably unstable to noises or other high-frequency perturbations in the image data.

B. Deblurring is Backward Diffusion. By the canonical PDE theory, to blur an image with a Gaussian kernel amounts to running the heat diffusion equation for some finite duration with the given image as the initial data. Therefore, to deblur is naturally equivalent to inverting the diffusion process, which is notoriously unstable.

Stochastically, diffusion corresponds to the Brownian motions of an initial ensemble of particles. Thus to deblur or to de-diffuse amounts to reversing an irreversible random spreading process, which is physically illposed.

C. Deblurring is Entropy Decreasing. The goal of deblurring is to reconstruct the detailed image features from a mollified blurry image. Thus from the standpoint of statistical mechanics, deblurring is a process to increase (Shannon) information, or equivalently, to *decrease entropy*. According to the second law of statistical mechanics [41], deblurring thus could never occur naturally and extra work has to be done to the system.

D. Deblurring is Inverting Compact Operators. In terms of abstract functional analysis, a blurring process is typically a *compact* operator. A compact operator is one that maps any bounded set (according to the associated Hilbert or Banach norms) to a much better behaved set which is precompact. To achieve this goal, intuitively speaking, a compact operator has to mix spatial information or introduce many coherent structures, which is often realized essentially by dimensionality reduction based on vanishing eigenvalues or singular values. Therefore to invert a compact operator is again equivalent to de-correlating spatial coherence or reconstructing the formerly suppressed dimensions (during the blurring process) of features and information, which is unstable.

This illustration can be further vivified via finite-dimensional linear algebra [65,66]. Looking for an unknown vector $\boldsymbol{u}$ of dimension much *higher* than its observation $\boldsymbol{b}$ for the matrix-vector equation $A\boldsymbol{u} = \boldsymbol{b}$ often has either no solution or infinitely many. Any *unique* meaningful solution has to be defined in some proper way.

3. Tikhonov and Bayesian Regularization

From the above discussion, proper regularization techniques have to be sought after in order to alleviate the illposedness of the deblurring process.

Two universal regularization approaches, which are essentially reciprocal in the two dual worlds of deterministic and stochastic methodologies, are Tikhonov regularization [69] and the Bayesian inference theory [45]. Their intimate connection has been explained in, for example, Mumford [53], and Chan, Shen, and Vese [20].

In essence, both approaches introduce some *prior* knowledge about the target images u to be reconstructed. In the Bayesian framework, it is to introduce some proper probability distribution over all possible image candidates, and necessary bias (i.e., regularization) is encouraged to favor more likely ones. In the Tikhonov setting, the prior knowledge is often reflected through some properly designed "energy" formulations, e.g., a quadratic energy like $a\|u\|^2$ under some proper functional norm.

We now introduce the most general framework of Bayesian-Tikhonov regularization for deblurring. Consider the blur model

$$u_0(x) = K[u](x) + n(x), \quad x = (x_1, x_2) \in \mathbb{R}^2,$$

with a general blur operator K and additive white noise n.

First, assume that blur process K is either known explicitly or estimated in advance [18]. As an estimation problem, deblurring can be carried out by the Bayesian principle or MAP (maximum a posteriori probability):

$$\hat{u} = \operatorname{argmax} \operatorname{Prob}(u \mid u_0, K),$$

or equivalently, in terms of the logarithmic likelihood or Gibbs' ensemble formula $E[\cdot] = -\log p(\cdot) +$ *a constant or fixed free energy* [24],

$$\hat{u} = \operatorname{argmin} E[u \mid u_0, K].$$

The Bayesian formula with a *known* blur K is given by

$$\mathrm{Prob}(u \mid u_0, K) = \mathrm{Prob}(u \mid K)\mathrm{Prob}(u_0 \mid u, K)/\mathrm{Prob}(u_0 \mid K).$$

Given an image observation u_0, the denominator is simply a fixed probability normalization constant. Thus effectively one seeks an estimator $\hat{u}$ to minimize the product of the *prior* model $\mathrm{Prob}(u \mid K)$ and the data (or fidelity) model $\mathrm{Prob}(u_0 \mid u, K)$. Since ideal images and blurs are often independent, one has $\mathrm{Prob}(u \mid K) = \mathrm{Prob}(u)$. Therefore in terms of the energy formulation, one attempts to minimize the *posterior energy*

$$E[u \mid u_0, K] = E[u] + E[u_0 \mid u, K]. \tag{5}$$

In the setting of Tikhonov regularization, the prior energy $E[u]$ is virtually a regularizer for the data fitting model $E[u_0 \mid u, K]$. Functionally, $E[u]$ can be specified by a suitable norm or semi-norm in some proper function space such as the BV or Besov spaces, which will be discussed later.

For *blind* deblurring when the kernel K is unknown, the Bayesian formula becomes

$$\max_{u,K} \mathrm{Prob}(u, K \mid u_0) = \mathrm{Prob}(u_0 \mid u, K)\mathrm{Prob}(u, K)/p(u_0).$$

In most applications, the blur mechanism K is uncorrelated to the image content u (e.g., in astronomical imaging, atmospheric turbulence activities K are not influenced by the ideal image observation u of the stars and galaxies many lightyears away). Then one has

$$\mathrm{Prob}(u, K) = \mathrm{Prob}(u)\mathrm{Prob}(K),$$

and the posterior energy takes the form of

$$E[u, K \mid u_0] = E[u_0 \mid u, K] + E[u] + E[K], \tag{6}$$

up to a fixed additive constant (corresponding to the free energy under the given parameters in the models).

In both models (5) and (6) for non-blind and blind deblurring, the data generation model $E[u_0 \mid u, K]$ is often readily expressible via squared fitting error for Gaussian white noise. Thus the key to effective deblurring relies upon the proper proposals on the prior knowledge for the target image u, as well as the blur process K in the blind scenario.

4. Optimal Wiener Filtering for Non-Blind Deblurring

From now on, instead of the black-faced symbols $\boldsymbol{x}$ and $\boldsymbol{\omega}$, a general pixel will be denoted by $x = (x_1, x_2)$ and its frequency dual variable by $\omega = (\omega_1, \omega_2)$. Due to the stochastic nature of Wiener filtering, we shall begin with a brief introduction to 2-D stochastic signals.

4.1. *2-D stochastic spatial signals*

Consider only *real* stochastic images defined on the domains of either $\mathbb{R}^2$ for analog images or the lattice $\mathbb{Z}^2$ for digital ones.

A stochastic image $\boldsymbol{u}(x)$ is said to be *homogeneous* if any of its finite marginal distributions carries no spatial memory, or equivalently, is translation invariant:

$$P_{x+z,\cdots,y+z}(u,\cdots,v) \equiv P_{x,\cdots,y}(u,\cdots,v), \quad \forall z = (z_1, z_2),$$

where the marginal probability is defined by

$$P_{x,\cdots,y}(u,\cdots,v)du\cdots dv = \mathrm{Prob}(\boldsymbol{u}(x) \in [u, u+du], \cdots, \boldsymbol{u}(y) \in [v, v+dv]).$$

Familiar sources for homogenous images include Gibbs' random fields with translation invariant potentials, or Markov random fields with translation invariant graph structures and local conditionals [6,18,40]. Homogeneity is appropriate for modelling certain ideal single-species textures such as sandy beaches or grasslands, which are more or less uniform.

More generally, a stochastic signal $\boldsymbol{u}$ is said to be wide-sense homogeneous (WSH), if its two-point auto-correlation function

$$R_{\boldsymbol{uu}}(x,y) = \mathrm{E}[\boldsymbol{u}(x)\boldsymbol{u}(y)],$$

is translation invariant: for any relocation z,

$$R_{\boldsymbol{uu}}(x+z, y+z) = R_{\boldsymbol{uu}}(x,y).$$

Thus if $\boldsymbol{u}$ is WSH, its auto-correlation function is essentially a single-pixel function: $R_{\boldsymbol{uu}}(x-y) = R_{\boldsymbol{uu}}(x,y)$. Let $\omega = (\omega_1, \omega_2)$ denote the spatial frequency variable. Then the power spectral density $S_{\boldsymbol{uu}}(\omega)$ is defined to be the Fourier transform of $R_{\boldsymbol{uu}}(x)$:

$$S_{\boldsymbol{uu}}(\omega) = \int_{\mathbb{R}^2} R_{\boldsymbol{uu}}(x) e^{-ix\omega} dx.$$

A WSH image $\boldsymbol{n}(x)$ is said to be *white noise*, if $S_{\boldsymbol{nn}}(\omega) \equiv \sigma^2$, or equivalently, its auto-correlation function $R_{\boldsymbol{nn}}(x)$ is a constant multiple of Dirac's delta signal $\delta(x)$.

Two WSH images $\boldsymbol{u}$ and $\boldsymbol{v}$ are said to be *cross*-WSH if their cross-correlation function is translation invariant as well.

$$R_{\boldsymbol{uv}}(x,y) = \mathrm{E}[\boldsymbol{u}(x)\boldsymbol{v}(y)] = R_{\boldsymbol{uv}}(x-y). \tag{7}$$

Define the cross-WSH set of a given WSH image $\boldsymbol{u}$ to be

$$\Lambda_{\boldsymbol{u}} = \{\boldsymbol{v} \mid \boldsymbol{v} \text{ is cross-WSH to } \boldsymbol{u}\ \}. \tag{8}$$

Then $\boldsymbol{u} \in \Lambda_{\boldsymbol{u}}$. Furthermore, we have the following list of straightforward but useful properties.

Theorem 1: Suppose $\boldsymbol{u}$ is WSH and $\Lambda_{\boldsymbol{u}}$ its cross-WSH set. Then $\Lambda_{\boldsymbol{u}}$ is a linear space which is closed under *spatial relocation*:

$$\boldsymbol{v}(\cdot) \in \Lambda_{\boldsymbol{u}} \Rightarrow \boldsymbol{v}(\cdot + z) \in \Lambda_{\boldsymbol{u}},$$

for any relocation $z \in \mathbb{R}^2$, as well as under *linear filtering*:

$$\boldsymbol{v} \in \Lambda_{\boldsymbol{u}} \Rightarrow h * \boldsymbol{v} \in \Lambda_{\boldsymbol{u}},$$

for any filter $h = h(x)$. Let $H(\omega)$ denote the impulse response of h. Then,

$$R_{h*\boldsymbol{v},\boldsymbol{u}}(x) = h * R_{\boldsymbol{vu}}(x), \text{ and } S_{h*\boldsymbol{v},\boldsymbol{u}}(\omega) = H(\omega)S_{\boldsymbol{vu}}(\omega).$$

4.2. *Stochastic signals as random generalized functions*

Another intriguing approach to stochastic signals is to treat a stochastic signal as a *random* generalized function.

Recall that a generalized function F, or a Schwartz distribution, is a linear functional on the test function space $\mathcal{D} = C_0^\infty(\mathbb{R}^2)$, so that for any test function $\phi \in \mathcal{D}$,

$$\text{the determinisitc values } \langle F, \phi\rangle \text{ are linear in } \phi.$$

A 2D stochastic field $\boldsymbol{u}$ on $\mathbb{R}^2$ can be treated as a *random* generalized function so that for any test function $\phi \in \mathcal{D}$, the value

$$U_\phi = \langle \boldsymbol{u}, \phi\rangle \text{ is a random variable, and } U_{a\phi+b\psi} = aU_\phi + bU_\psi.$$

The mean field of $\boldsymbol{u}$ is an *ordinary* generalized function $m_{\boldsymbol{u}}$ such that

$$\mathrm{E}(U_\phi) = \langle m_{\boldsymbol{u}}, \phi\rangle, \qquad \text{for any } \phi \in \mathcal{D}.$$

If $m_{\boldsymbol{u}} = 0$, $\boldsymbol{u}$ is said to have zero means. Two random fields $\boldsymbol{u}$ and $\boldsymbol{v}$ are said to be *equivalent* if U_ϕ and V_ϕ share the same probability distribution for any test function $\phi \in \mathcal{D}$.

For any shifting operator $S_z : \phi(x) \to \phi(x-z)$ with $z \in \mathbb{R}^2$, the shifted random field $S_z\boldsymbol{u}$ is defined by the dual formula

$$\langle S_z\boldsymbol{u}, \phi\rangle = \langle \boldsymbol{u}, S_{-z}\phi\rangle, \qquad \phi \in \mathcal{D}.$$

$\boldsymbol{u}$ is said to be *shift invariant* if $\boldsymbol{u}$ is equivalent to $S_z\boldsymbol{u}$ for any shift z. If a test function ϕ is interpreted as a measurement sensor, then a random field is shift invariant if and only if *no* statistical difference can be detected from the measurements when a sensor is moved from one location to another.

A random field $\boldsymbol{u}$ of zero means is said to be wide-sense homogeneous (WSH) if there exists some locally integrable function $R_{\boldsymbol{uu}}(x)$ such that for any two test functions ϕ and ψ, one has

$$\mathrm{E}(U_\phi U_\psi) = \langle \phi, R_{\boldsymbol{uu}} * \psi\rangle.$$

Similarly, two random fields $\boldsymbol{u}$ and $\boldsymbol{v}$ of zero means are said to be cross-WSH if there exists some locally integrable function $R_{\boldsymbol{uv}}(x)$ such that for any test functions ϕ and ψ,

$$\mathrm{E}(U_\phi V_\psi) = \langle \phi, R_{\boldsymbol{uv}} * \psi\rangle.$$

$R_{\boldsymbol{uu}}$ and $R_{\boldsymbol{uv}}$ are called the auto-correlation and cross-correlation functions, and are apparently unique if in existence. If one *formerly* takes Dirac's delta functions as test functions, it is easy to verify the consistency between the current functional definitions and the pointwise definitions in the preceding subsection.

The reader can familiarize the above theory with the help of the following example of *random* harmonic waves in 1D:

$$\boldsymbol{u}(x) = A\cos(x+B), \qquad x \in \mathbb{R},$$

where A and B are independent random variables with B *uniformly* distributed over $[0, 2\pi)$, and A *exponentially* distributed on $(0, \infty)$. Then it is easy to show, for example, that $\boldsymbol{u}$ must be homogenous (or shift invariant).

4.3. *Filtering-based deblurring*

Assume that the blur is shift invariant: $u_0 = k * u + n$. Filtering-based deblurring is to produce an estimator $\hat{u}$ of the ideal image u via a linear filtering scheme:

$$\hat{u} = \hat{u}_w = w * u_0, \quad \text{with a suitable filter } w.$$

Without noise, the ideal filter would be directly given by $W(\omega) = \frac{1}{K(\omega)}$, in the Fourier domain, so that $\hat{u} = w * u_0 \equiv u$ for any clear image u and *perfect reconstruction* is reached! However, it is a rather unnerving formula since a typical blur k is often *lowpass* and $K(\omega)$ decays rapidly at high frequencies. Such a naive filter therefore exaggerates any high-frequency errors or perturbations.

To alleviate such unwanted instability, in the noise-free case one rewrites the naive filter $W = 1/K$ to

$$W(\omega) = \frac{K^*(\omega)}{K(\omega)K^*(\omega)} = \frac{K^*}{|K|^2}, \quad \text{where } ^* \text{ denotes complex conjugacy.}$$

The vanishing denominator at high frequencies can be guarded away from zero by incorporating a positive factor $r = r(\omega)$:

$$W \to W_r = \frac{K^*}{|K|^2 + r}. \tag{9}$$

The resultant deblurred image $\hat{u}_r$ is then given by

$$\hat{u}_r = w_r * k * u,$$

or in the Fourier domain, the composite effect of the blurring and deblurring processes is achieved by the multiplier

$$W_r(\omega)K(\omega) = \frac{|K(\omega)|^2}{|K(\omega)|^2 + r(\omega)}. \tag{10}$$

The restoration indeed well approximates the identity operator on low frequencies where $r \ll |K|^2$ since K is lowpass. High frequencies are however suppressed since K almost vanishes and $|K|^2 \ll r$. Thus the regularizer r plays a soft cutoff role.

The reader should pay constant attention to the frequent emergency of such an r-factor henceforth. It embodies a universal quantity that is critical for any deblurring problem.

The question is how to choose wisely an optimal regularizer r. A uniform constant is a reasonable guess but lacks clear theoretical backup. What Wiener discovered was that r should be related to the signal-to-noise ratio in the observation u_0, which will be explained in the next subsection.

It is also the right spot to reiterate the earlier analogy drawn from the finite linear algebra of solving $A\boldsymbol{u} = \boldsymbol{b}$. Recall that the least square solution [65,66] is given by the *normal equation*:

$$A^*A\hat{\boldsymbol{u}} = A^*\boldsymbol{b}, \;\text{ or }\; \hat{\boldsymbol{u}} = (A^*A)^{-1}A^*A\boldsymbol{u}.$$

Thus the "filter" in the least-square solution is given by

$$W = (A^*A)^{-1}A^*.$$

When the linear operator A "mixes" information too intensely so that

$$\operatorname{rank}(A) = \dim(\operatorname{Range}(A)) \ll \#\text{ of rows of } A,$$

A^*A becomes singular (or almost singular if the approximate rank only counts the nonnegligible singular values of A). Then the inversion of $(A^*A)^{-1}$ still remains illposed or unstable. In linear algebra, the filter is then regularized to

$$W_r = (A^*A + rI)^{-1}A^* \tag{11}$$

for some positive small parameter $r > 0$, where I is the identity matrix. The resultant estimator corresponds to the regularized least-square problem:

$$\hat{\boldsymbol{u}}_r = \operatorname{argmin}_{\boldsymbol{u}} \|A\boldsymbol{u} - \boldsymbol{b}\|^2 + r\|\boldsymbol{u}\|^2.$$

Notice the characteristic similarity between (9) and (11).

4.4. *Optimal Wiener filtering*

Wiener's filter w is to minimize the mean squared estimation error e_w defined by $e_w(x) = \hat{u}_w(x) - u(x)$. That is,

$$w = \operatorname{argmin}_h \mathrm{E}(e_h^2) = \operatorname{argmin}_h \mathrm{E}(h * u_0(x) - u(x))^2. \tag{12}$$

Notice that Wiener's filter is independent of the particular pixel x used in the above definition since e_h is easily seen to be WSH for any fixed real filter $h = h(x)$, provided that u the ideal image and n the noise are independent and WSH.

Variation on the optimal Wiener filter: $w \to w + \delta h$ gives the "equilibrium" equation

$$\mathrm{E}[(w * u_0(x) - u(x))\ (\delta h * u_0(x))] = 0.$$

Taking localized small variation $\delta h(x) = \varepsilon\delta(x - a)$ for some $\varepsilon \ll 1$ at any site a, one can rewrite the equation to $\mathrm{E}[(w * u_0(x) - u(x))u_0(x - a)] = 0$. Since a is arbitrary, it is equivalent to

$$\mathrm{E}[(w * u_0(x) - u(x))u_0(y)] = 0, \quad \forall x, y \in \Omega, \tag{13}$$

the one known as the *orthogonal condition* for Wiener's filter.

By Theorem 1, in terms of the correlation functions, one has

$$w * R_{u_0u_0}(z) = R_{uu_0}(z), \qquad z \in \mathbb{R}^2.$$

The optimal Wiener filter is thus given by $W(\omega) = S_{uu_0}(\omega)/S_{u_0u_0}(\omega)$, expressed in terms of the power spectral densities. For the blur model: $u_0 = k * u + n$, one has, according to Theorem 1,

$$S_{uu_0} = K^*(\omega)S_{uu}(\omega), \text{ and } S_{u_0u_0} = |K(\omega)|^2 S_{uu}(\omega) + S_{nn}(\omega).$$

Therefore, we have established the following theorem.

Theorem 2: (Wiener Filter for Deblurring) The optimal Wiener filter is given by, in the Fourier domain,

$$W(\omega) = \frac{K^* S_{uu}}{|K|^2 S_{uu} + S_{nn}} = \frac{K^*}{|K|^2 + r_w}, \tag{14}$$

where the regularizer $r_w = S_{nn}/S_{uu}$ is the squared noise-to-signal ratio.

For a Gaussian white noise with variance σ^2, one has $S_{nn}(\omega) \equiv \sigma^2$. Since S_{uu} is often bounded, the Wiener regularizer r_w is therefore well bounded above zero.

We refer the reader to, e.g., [43,47] for further improvement of the above classical Wiener filters, especially on relaxing the stochastic assumptions on the signals and the conditions on the blur model.

4.5. *Connection to the Bayesian/Tikhonov method*

We now show that Wiener filtering is intimately connected to the general framework of Bayesian or Tikhonov regularization laid out in the preceding section.

Take the quadratic data-fitting model

$$E[u_0 \mid u, k] = \lambda \|k * u - u_0\|^2 = \lambda \int_{\mathbb{R}^2} (k * u - u_0)^2 dx$$

for additive Gaussian white noise, where λ is inversely proportional to the noise variance σ^2.

For the prior model $E[u]$, assume the ideal image u belongs to the fractional-Sobolev space $H^\gamma(\mathbb{R}^2)$. Formally, this means that $u \in L^2(\mathbb{R}^2)$ and its fractional gradient $\nabla^\gamma u \in L^2(\mathbb{R}^2)$. More rigorously, the norm is properly defined in the Fourier domain by:

$$\|u\|_\gamma^2 = \int_{\mathbb{R}^2} (1 + |\omega|^2)^\gamma |U(\omega)|^2 d\omega,$$

where $U(\omega)$ denotes the Fourier transform of $u(x)$. Define

$$r(\omega) = \frac{(1 + |\omega|^2)^\gamma}{\lambda} = \frac{\lambda^{-1}}{(1 + |\omega|^2)^{-\gamma}}. \tag{15}$$

Notice that $r(\omega)$ can indeed be considered as the squared noise-to-signal ratio as in the Wiener filter since λ^{-1} is proportional to the noise variance and the denominator is proportional to the squared signal strength. (More precisely, the noise n has been assumed in L^2 and its power spectral density is the ideal variance σ^2 modulated by some decay factor $|\omega|^{-2\alpha}$, which is however shared by the signal and cancelled out. This makes the above r a more authentic squared noise-to-signal ratio.)

Notice that the power-law decay in (15) is very common in stochastic signal analysis and processing.

One is thus led to the following posterior energy for deblurring:

$$E[u \mid u_0, k] = \lambda\|k * u - u_0\|^2 + \|u\|_\gamma^2.$$

In the Fourier domain, it is equivalent to the energy

$$E[U \mid U_0, K] = \int_\Omega |K(\omega)U(\omega) - U_0(\omega)|^2 \, d\omega + \int_\Omega r(\omega)|U(\omega)|^2 d\omega. \quad (16)$$

Performing variation on U, one has the equilibrium equation for the optimal estimator:

$$K^*(KU - U_0) + rU = 0, \quad \text{or} \quad U(\omega) = \frac{K^*(\omega)}{|K|^2(\omega) + r(\omega)} U_0(\omega),$$

with r being the squared noise-to-signal ratio. This could be considered as the deterministic version of Wiener filtering. To our best knowledge, such an explicit connection has never been made before in the literature.

More generally, with $r(\omega)$ already computed for Wiener filtering (14), one can substitute it into the Bayesian/Tikhonov formulation (16), and arrive at the precise posterior energy form for *deterministic* Wiener filtering.

An interesting case occurs if $r(\omega) = (1+|\omega|^2)^{-\mu}$ for some notable $\mu > 0$, which corresponds to the scenario when the target image signal u is highly oscillatory, or is functionally a generalized function instead of L^2 [64].

5. Deblurring Blurred BV Images

One of the most powerful deterministic image prior model is the space of functions of bounded variations $\mathrm{BV}(\Omega)$, first introduced into image processing by Rudin, Osher, and Fatemi [61]. In this section, we discuss the theory and computation of deblurring BV images.

5.1. *TV deblurring by Rudin, Osher, and Fatemi*

The total variation (TV) of a BV image u is conventionally denoted by $\int_\Omega |Du|$ or $|Du|(\Omega)$ [38,42]. When the image u is smooth so that its gradient

∇u belongs to L^1, the TV is simply the ordinary L^1 integral

$$\int_\Omega |Du| = \int_\Omega |\nabla u| dx,$$

in the sense of Sobolev norm. For a more generic BV image u that has discontinuous jumps, the TV $|Du|$ is in fact a Radon measure so that for any open set $Q \subset \Omega$,

$$|Du|(Q) = \sup_{\boldsymbol{g} \in C_0^1(Q, B^2)} \int_Q u(\nabla \cdot \boldsymbol{g}) dx,$$

where $B^2 \subset \mathbb{R}^2$ denotes the unit open disk centered at the origin and $\boldsymbol{g} = (g_1, g_2)$ is vectorial. For more introduction to BV images and their applications in image analysis and processing, we refer the reader to our new monograph [18], the more mathematically oriented monographs [38,42], as well as numerous existent works, e.g., [1,7,9,12,16,17,50,57,58,62,63,70,71,72].

In one adopts the TV measure for image regularization: $E[u] = \alpha \int_\Omega |Du|$, the posterior energy for Bayesian/Tikhonov deblurring then takes the form of

$$\begin{aligned} E[u \mid u_0, k] &= E[u] + E[u_0 \mid u, k] \\ &= \alpha \int_\Omega |Du| + \frac{\lambda}{2} \int_\Omega (k * u - u_0)^2 dx, \end{aligned} \tag{17}$$

with $x = (x_1, x_2) \in \Omega = \mathbb{R}^2$ and two suitable positive weights α and λ. This was the restoration model originally proposed and computed by Rudin-Osher-Fatemi [60,61], and later further studied by many others [9,72,73].

Notice that as far as energy minimization is concerned, only the ratio $r = \alpha/\lambda$ contributes to the solution process. As for parametric estimation in statistics, one could also treat r as an unknown as well, and expand the energy to $E[u, r \mid u_0, k]$ by absorbing some prior knowledge $E[r]$ on r.

The previous discussion on the optimal Wiener filtering (14) seems to suggest that the ratio $r = \alpha/\lambda$ is in the same dimension of the noise-to-signal ratio r_w. In particular, r should be proportional to the variance σ^2 of the noise, which is natural since by the Bayesian rationale for least square fidelities, one indeed has $\lambda = O(1/\sigma^2)$.

5.2. *Dealing with bounded image domains*

In model (17), it has been conveniently assumed that the image domain Ω is the entire plane $\mathbb{R}^2$ to facilitate shift invariance. In real applications, however, Ω is often a bounded disk or square for which the blur

$$K[u] = k * u(x) = \int_{\mathbb{R}^2} k(x-y)u(y)dy, \qquad x \in \Omega$$

needs to be properly redefined.

First, one can remodify the blur to a *shift-variant* PSF given by

$$k(x,y) = \frac{k(x-y)}{\int_\Omega k(x-z)dz}, \quad \forall x, y \in \Omega. \tag{18}$$

We assume that the original PSF $k(x)$ is nonnegative, and $x = (0,0)$ belongs to the support of the measure $d\mu(x) = k(x)dx$. That is, the integral of $k(x)$ on any neighborhood of $(0,0)$ is positive. Then the denominator in (18) is always positive. It is also easy to see that the DC-condition $K[1] = 1$ still holds after the modification.

An alternative way is to first extrapolate u beyond Ω. Let

$$Q : u\big|_\Omega \to \tilde{u} = Q[u]\big|_{\mathbb{R}^2},$$

be a suitable linear extrapolation operator which extends u on Ω onto the entire plane. (Functionally, Q could be some linear operator from, e.g., $W^{1,\infty}(\Omega)$ to $W^{1,\infty}(\mathbb{R}^2)$.) Then the blur is modified to

$$K[u](x) = k * \tilde{u}(x) = k * Q[u](x), \quad \forall x \in \Omega, \tag{19}$$

or equivalently, $K = 1_\Omega \cdot (k * Q)$ with a multiplier $1_\Omega(x)$.

The DC-condition is satisfied if and only if $k * Q[1] \equiv 1$ when restricted in Ω. In particular, the natural condition $Q[1] \equiv 1$ would suffice since k satisfies the DC-condition on $\mathbb{R}^2$.

If Q is represented by some kernel $g(x,y)$ with $y \in \Omega, x \in \mathbb{R}^2$. Then the modified K is represented by

$$k(x,y) = \int_{\mathbb{R}^2} k(x-z)g(z,y)dz, \quad x, y \in \Omega.$$

Therefore the DC-condition is satisfied when g and k meet the following compatibility condition

$$\int_\Omega \int_{\mathbb{R}^2} k(x-z)g(z,y)dzdy \equiv 1, \quad \forall x \in \Omega. \tag{20}$$

Finally, another less traditional approach to handling bounded domains can be based on the inpainting technique [4,5,11,13,14,15,16,17,19,36]. Suppose that $k(x)$ is compactly supported on a disk $B_\rho(0) = \{x \in \mathbb{R}^2 : |x| < \rho\}$, and the ρ-neighborhood of Ω is defined by

$$\Omega_\rho = \{x \in \mathbb{R}^2 \mid \operatorname{dist}(x, \Omega) < \rho\}.$$

Assume also the ideal image $u \in \mathrm{BV}(\Omega)$. Then, instead of the original model (17), one can attempt to minimize the modified version –

$$E[u \mid u_0, k, \rho] = \alpha \int_{\Omega_\rho} |Du| + \frac{\lambda}{2} \int_{\Omega} (k * u - u_0)^2 dx. \tag{21}$$

The convolution inside the fidelity term no longer stirs up any problem.

In summary, both the restricted-kernel method (18) and the image-extrapolation method (19) lead to a *shift-variant* blur K with kernel $k(x, y)$, and the deblurring model for BV images becomes

$$\min_u E_{\mathrm{TV}}[u \mid u_0, K] = \alpha \int_{\Omega} |Du| + \frac{\lambda}{2} \int_{\Omega} (K[u] - u_0)^2 dx. \tag{22}$$

Next we briefly discuss the solutions to this model. More details can be found, for example, in [1,9,18].

5.3. *Existence and uniqueness*

Following the preceding preparation, the image domain Ω can be assumed bounded and Lipschitz in $\mathbb{R}^2$. In addition, we assume that (i) the ideal image $u \in \mathrm{BV}(\Omega)$, (ii) the blurry and noisy observation $u_0 \in L^2(\Omega)$, and (iii) the linear blur $K : L^1(\Omega) \to L^2(\Omega)$ is bounded, injective, and satisfies the DC-condition: $K[1] \equiv 1$.

Condition (i) and (ii) are necessary for (22) to be well defined. Injectivity in (iii) is also necessary for the uniqueness of optimal deblurring.

The proof for the following theorem can be found in, e.g., [9,18].

Theorem 3: (Existence and Uniqueness of BV Deblurring) Under the preceding three conditions, the optimal deblurring $u_* = \mathrm{argmin} E[u \mid u_0, K]$ for model (22) exists and is unique.

Furthermore, the unique minimizer must satisfy the mean constraint.

Corollary 4: *(The Mean Constraint) The unique minimizer u_* must automatically satisfy the mean constraint $\langle K[u_*] \rangle = \langle u_0 \rangle$.*

Stochastically, this is a natural inference from the blur model

$$u_0 = K[u] + n,$$

since the noise has zero means. Deterministically, this fact has to be proven from the deblurring model.

Proof: For the unique minimizer u_*, define for any $c \in \mathbb{R}$

$$e(c) = E[u_* - c \mid u_0, K].$$

Then $c_* = \operatorname{argmin} e(c)$ has to minimize

$$\int_\Omega (K[u_*] - u_0 - c)^2 dx, \qquad \text{since } K[c] = c.$$

As a result, the unique minimizer $c_* = \langle K[u_*] - u_0 \rangle$. On the other hand c_* has to be zero since $u_* - c_* = u_*$ due to uniqueness. Therefore,

$$\langle K[u_*] \rangle = \langle u_0 \rangle,$$

which establishes the assertion. □

Figures 3, 4, and 5 are three generic examples from [18] that demonstrate the performance of the deblurring model (22).

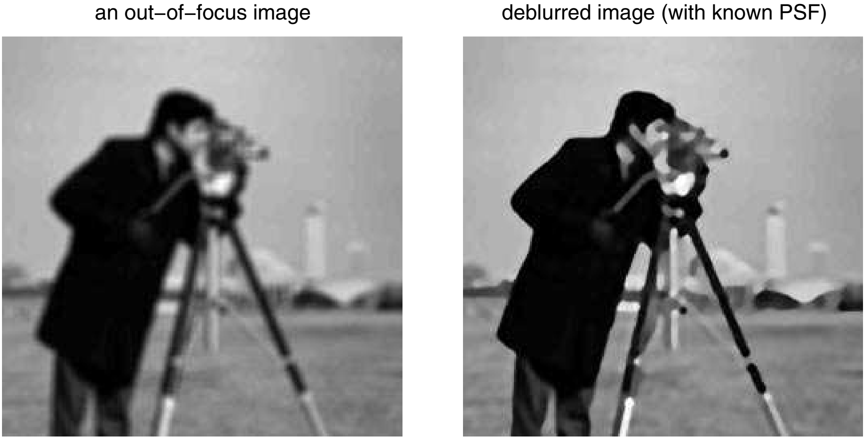


Fig. 3. Deblurring an out-of-focus image.

5.4. *Computation and examples*

The variational deblurring model (22) has been computed more or less based on the formal Euler-Lagrange equation [1,9,60,61]:

$$\alpha \nabla \cdot \left[\frac{\nabla u}{|\nabla u|} \right] - \lambda K^*[K[u] - u_0] = 0, \tag{23}$$

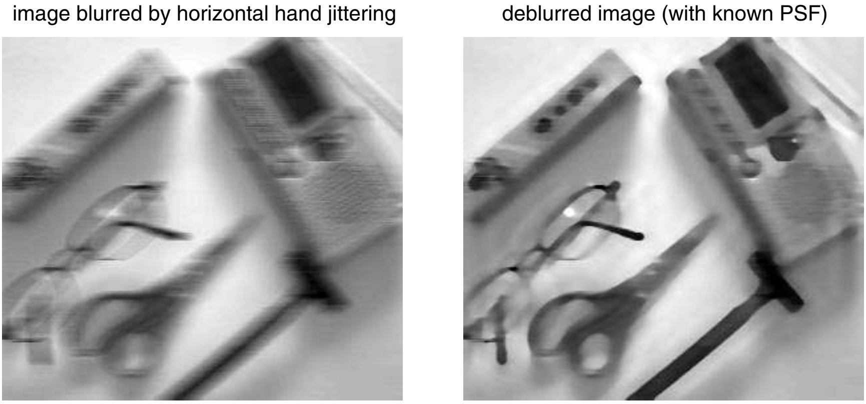


image blurred by horizontal hand jittering
deblurred image (with known PSF)

Fig. 4. Deblurring a motion-blurred image.

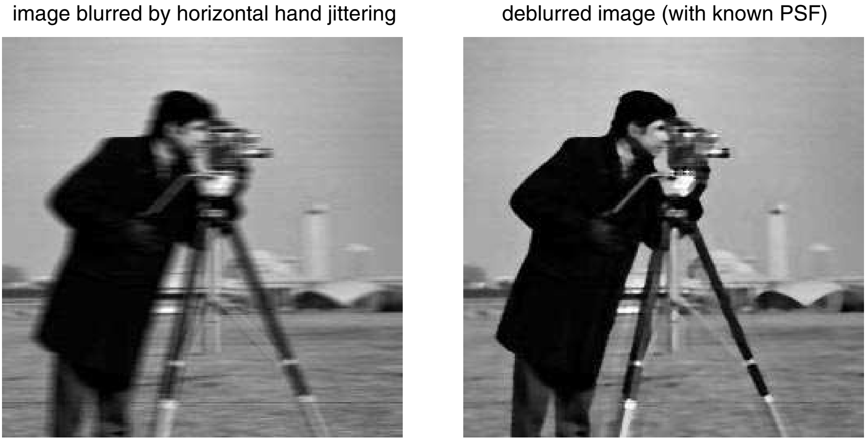


image blurred by horizontal hand jittering
deblurred image (with known PSF)

Fig. 5. Restoring another motion-blurred image.

with Neumann adiabatic condition $\partial u/\partial n = 0$ along the boundary $\partial\Omega$. Equation (23) holds in the distributional sense, i.e., for any compactly supported smooth test function ϕ, the solution satisfies

$$\alpha\langle\nabla\phi, \frac{\nabla u}{|\nabla u|}\rangle + \lambda\langle K[\phi], K[u] - u_0\rangle = 0.$$

The nonlinear degenerate elliptic equation (23) is often regularized to

$$\alpha\nabla\cdot\left[\frac{\nabla u}{|\nabla u|_a}\right] - \lambda K^*[K[u] - u_0] = 0, \tag{24}$$

where the notation $|x|_a$ denotes $\sqrt{x^2+a^2}$ for some fixed positive parameter a. It amounts to minimizing

$$E[u \mid u_0, K, a] = \alpha \int_\Omega \sqrt{|Du|^2 + a^2} + \frac{\lambda}{2} \int_\Omega (K[u] - u_0)^2 dx, \tag{25}$$

which is closely connected to the minimal surface problem [42].

Computationally, the most common algorithm has been based on the so-called *lagged-diffusivity* technique [9,18,72], which is an iterative procedure. Based on the current best estimation $u^{(n)}$, one solves for $u^{(n+1)}$ the following linearized equation:

$$\alpha \nabla \cdot \left[\frac{\nabla u^{(n+1)}}{|\nabla u^{(n)}|_a} \right] - \lambda K^*[K[u^{(n+1)}] - u_0] = 0, \tag{26}$$

with the Neumann condition. Notice that given $u^{(n)}$, the linear operator

$$L_n = -\alpha \nabla \cdot \frac{1}{|\nabla u^{(n)}|_a} \nabla + \lambda K^* K$$

is positive definite or strictly elliptic.

This algorithm guarantees convergence since it is equivalent to the alternating-minimization (AM) algorithm for the *augmented* energy

$$E_a[u, z \mid u_0, K] = \frac{\alpha}{2} \int_\Omega \left(z|\nabla u|^2 + z^{-1} \right) dx + \frac{\lambda}{2} \int_\Omega (K[u] - u_0)^2 dx,$$

where $z = z(x)$ is an auxiliary field, which corresponds to the edge signature function in image processing [17]. Then it can be easily shown that

$$\min_{u,z} E_a[u, z \mid u_0, K] = \min_u E_{\mathrm{TV}}[u \mid u_0, K].$$

Furthermore, the above lagged-diffusivity algorithm corresponds to exactly the AM algorithm for the augmented energy:

$$\cdots \to u^{(n)} \to z^{(n)} \to u^{(n+1)} \to \cdots .$$

6. Parametric Blind Deblurring

In all the above models, the blur K has been assumed known. We now develop variational deblurring models when the blur is unknown, a scenario often nicknamed "blind deblurring" [39,46,75]. Inspired by the theory of statistical estimation, we shall classify such models into ones that are *parametric* or *nonparametric*, or figuratively, *partially blind* or *completely blind*.

6.1. *Parametric modeling*

Suppose that the unknown linear blur belongs to a known parametric family

$$\mathcal{K} = \{K_\theta \mid \theta \in I \subset \mathbb{R}^d\},$$

where $\theta = (\theta_1, \cdots, \theta_d)$ denotes a d-dimensional parametric vector and varies on a subset or domain I in $\mathbb{R}^d$. One is therefore not completely "blind" to the blur operator, and the uncertainty only arises from θ. A familiar example is the Gaussian family of shift-invariant blurs $K_\theta = g*$ given by

$$g(x \mid \theta) = \frac{1}{2\pi\theta} \exp\left(-\frac{x_1^2 + x_2^2}{2\theta}\right), \qquad \theta \in I = (0, \infty), \tag{27}$$

where in statistics θ precisely corresponds to the variance σ^2.

By the Bayesian rationale stated previously [18,53], parametric blind deblurring becomes the minimization of

$$E[u, \theta \mid u_0] = E[u_0 \mid u, \theta] + E[u] + E[\theta]. \tag{28}$$

The first two terms can be safely copied from the non-blind deblurring model discussed previously. Thus it suffices to incorporate some appropriate model for the parameter distribution $p(\theta)$ or $E[\theta]$.

Suppose $u \in \mathrm{BV}(\Omega)$, $\theta \in I \subset R^d$, and $E[\theta] = \phi(\theta)$ for some suitable function ϕ. Then the deblurring model is explicitly given by

$$E[u, \theta \mid u_0] = \alpha \int_\Omega |Du| + \frac{\lambda}{2} \int_\Omega (K_\theta[u] - u_0)^2 dx + \phi(\theta). \tag{29}$$

Assume that $\phi(\theta)$ is bounded below: $\phi(\theta) \geq M > -\infty$ for all $\theta \in I$. Otherwise it can attenuate the role of the first two terms in (29) and distort the real intention of the model. As an example, consider the Gaussian family in (27). Suppose the variance θ is subject to the exponential distribution with density function:

$$p(\theta) = a \exp(-a\theta), \qquad \theta \in I = (0, \infty), \quad \text{for some } a > 0. \tag{30}$$

Then $\phi(\theta) = E[\theta] = -\ln p(\theta) = a\theta - \ln a \geq -\ln a > -\infty$.

Following Theorem 3, K_θ is assumed to be injective and satisfy the DC-condition $K_\theta[1] = 1$. Then for any given θ, the conditional minimizer

$$\hat{u}_\theta = \operatorname{argmin} E[u \mid u_0, K_\theta] = \operatorname{argmin} \alpha \int_\Omega |Du| + \frac{\lambda}{2} \int_\Omega (K_\theta[u] - u_0)^2 dx \tag{31}$$

always exists and is unique by Theorem 3. The original model (29) is then reduced to an optimization problem on the parameter domain $I \subset \mathbb{R}^d$:

$$\min_{\theta \in I} e(\theta), \ \text{ with } \ e(\theta) = E[\hat{u}_\theta, \theta \mid u_0].$$

$e(\theta)$ is, however, generally non-convex and consequently the global minimizer $(\hat{u}_{\theta_*}, \theta_*)$ could be non-unique.

6.2. *The AM algorithm*

Such a multivariable optimization problem can usually be solved by the alternating-minimization (AM) algorithm [2,3,18,22,23,36,54]. One starts with some initial guess $\theta^{(0)}$, which could be drawn from $\operatorname{argmin} \phi(\theta)$ for instance. Then, one successively obtains the alternating conditional minimizers

$$\theta^{(0)} \to u^{(0)} \to \theta^{(1)} \to u^{(1)} \to \cdots \tag{32}$$

by optimizing the conditional energies:

$$\begin{aligned} u^{(n)} &= \operatorname{argmin} E[u \mid u_0, \theta^{(n)}], \quad \text{followed by} \\ \theta^{(n+1)} &= \operatorname{argmin} E[\theta \mid u_0, u^{(n)}], \quad \text{where} \\ E[\theta \mid u_0, u] &= \frac{\lambda}{2} \int_\Omega (K_\theta[u] - u_0)^2 dx + \phi(\theta). \end{aligned} \tag{33}$$

Notice that in the language of conditional probabilities, the Markov property holds for the zigzag sequence (32):

$$\begin{aligned} \operatorname{Prob}(\theta^{(n+1)} \mid u^{(n)}, \theta^{(n)}, u^{(n-1)}, \cdots) &= \operatorname{Prob}(\theta^{(n+1)} \mid u^{(n)}), \\ \operatorname{Prob}(u^{(n)} \mid \theta^{(n)}, u^{(n-1)}, \theta^{(n-1)}, \cdots) &= \operatorname{Prob}(u^{(n)} \mid \theta^{(n)}). \end{aligned}$$

By Theorem 3, the conditional update $\theta^{(n)} \to u^{(n)}$ must be unique, while the conditional parameter estimation $u^{(n)} \to \theta^{(n+1)}$ could be nonunique. Uniqueness can, however, still be enforced by some extra sorting scheme, e.g.,

$$\theta^{(n+1)} = \operatorname{argmin} \{\phi(\theta) \mid \theta \in \operatorname{argmin} E[\theta \mid u_0, u^{(n)}]\},$$

provided that $\phi(\theta)$ is strictly convex. The following is evident for AM.

Proposition 5: *(Alternating Minimization is Monotone) For each $n \geq 0$,*

$$E[u^{(n+1)}, \theta^{(n+1)} \mid u_0] \leq E[u^{(n)}, \theta^{(n)} \mid u_0].$$

Let $B(L^1, L^2)$ denote the Banach space of all bounded linear operators from $L^1(\Omega)$ to $L^2(\Omega)$. Then the following convergence result holds, whose proof can be found in our monograph [18].

Theorem 6: (Convergence of Alternating Minimization) Assume that

(a) the blur parametrization

$$K : I \subset \mathbb{R}^d \to B(L^1, L^2), \quad \theta \to K_\theta$$

is a continuous mapping; and

(b) $\phi(\theta)$ is lower semi-continuous in $\theta \in I$.

Then, if as $n \to \infty$, $u^{(n)} \to u_*$ in $L^1(\Omega)$ and $\theta^{(n)} \to \theta_* \in I$, the limit pair (u_*, θ_*) satisfies

$$u_* = \operatorname{argmin} E[u \mid u_0, \theta_*], \quad \theta_* = \operatorname{argmin} E[\theta \mid u_0, u_*]. \tag{34}$$

We must point out that the continuity on blur parametrization is strong but not baseless. Consider, for example, the shift-invariant Gaussian family (27) on $\Omega = \mathbb{R}^2$. By Young's inequality [48], one has

$$\|(K_\theta - K_{\theta'})[u]\|_2 = \|(g(x \mid \theta) - g(x \mid \theta')) * u\|_2 \leq \|g(x \mid \theta) - g(x \mid \theta')\|_2 \|u\|_1.$$

Therefore, $\|K_\theta - K_{\theta'}\| \leq \|g(x \mid \theta) - g(x \mid \theta')\|_2$, which indeed converges to zero for any $\theta' > 0$, and $\theta \to \theta'$.

In terms of the first formal variations,

$$\frac{\partial}{\partial u} E[u_* \mid u_0, \theta_*] = \frac{\partial}{\partial u} E[u_*, \theta_* \mid u_0]$$
$$\frac{\partial}{\partial \theta} E[\theta_* \mid u_0, u_*] = \frac{\partial}{\partial \theta} E[u_*, \theta_* \mid u_0].$$

Thus the limit (u_*, θ_*) does satisfy the equilibrium equations of the deblurring model $E[u, \theta \mid u_0]$, and consequently offers a good candidate for optimal deblurring. In particular, if $E[u, \theta \mid u_0]$ is strictly convex on $(u, \theta) \in \mathrm{BV}(\Omega) \times I$, (u_*, θ_*) must be the unique global minimizer.

7. Non-Parametric Blind Deblurring: Double-BV Model

7.1. *General formulation of blind deblurring*

If the blur operator K is completely unknown, deblurring is conceivably much more challenging than the previous cases. Instead of estimating a few parameters, now one has to reconstruct the *entire* blur process K.

Herein we study only the shift-invariant case when the image observation u_0 is defined on $\Omega = \mathbb{R}^2$ with a PSF $k(x)$. The blur operator is thus reduced to a function, which is simpler than the general situation and can be managed with proper regularizations.

By the general Bayesian/Tikhonov framework, one attempts to minimize the posterior energy

$$E[u, k \mid u_0] = E[u] + E[u_0 \mid u, k] + E[k],$$

provided that the blur is independent of the image, as discussed earlier. In the case of BV images and Gaussian white noise, one has

$$E[u] = \alpha \int_{\mathbb{R}^2} |Du|, \qquad E[u_0 \mid u, k] = \frac{\lambda}{2} \int_{\mathbb{R}^2} (k * u - u_0)^2 dx.$$

Thus the key to successful deblurring lies in a proper proposal for the blur prior $E[k]$.

When the blur k is smooth, e.g., a Gaussian kernel, one may naturally enforce the Sobolev regularity [75]: $E[k] = \beta \int_{\mathbb{R}^2} |\nabla k|^2 dx$. Generally, such prior knowledge must be formulated based on the physical mechanism that drives the blur process, e.g., the atmospheric turbulence.

7.2. *Double-BV blind deblurring model of Chan and Wong*

In motion blurs due to sudden jitters or out-of-focus blurs arising from ideal diffraction-free lenses (see, e.g., Chan and Shen [18]), the PSF's are typically compactly supported with sharp cutoff boundaries. In such scenarios, as for images with sharp edges, the total variation regularity seems more appealing for the blur k as well. This leads to the double-BV blind deblurring model of Chan and Wong [23]:

$$E[u, k \mid u_0 = \alpha \int_{\mathbb{R}^2} |Du| + \beta \int_{\mathbb{R}^2} |Dk| + \frac{\lambda}{2} \int_{\mathbb{R}^2} (k * u - u_0)^2 dx. \qquad (35)$$

The detailed analysis for the double-BV model first appeared in our recent monograph [18]. Herein we only briefly introduce the most essential ingredients and refer the reader to [18] for more involved proofs and explanations.

For $\Omega = \mathbb{R}^2$, the BV norm is conventionally defined as [37,42]

$$\|u\|_{\mathrm{BV}} = \|u\|_{L^1(\mathbb{R}^2)} + |Du|(\mathbb{R}^2). \qquad (36)$$

While most results on BV functions in the literature are for bounded domains, it is worthwhile to pay extra attention to the complexity arising from unboundedness. We refer the reader to the more detailed discussion in Chan and Shen [18].

We now first extend a Poincaré inequality from bounded domains to $\mathbb{R}^2$, whose proof can be found in [18].

Theorem 7: (Poincaré Inequality for $\mathrm{BV}(\mathbb{R}^2)$) Suppose u belongs to $\mathrm{BV}(\mathbb{R}^2)$ with finite BV-norm defined as in (36). Then $u \in L^2(\mathbb{R}^2)$, and more specifically,

$$\|u\|_{L^2(\mathbb{R}^2)} \le C|Du|(\mathbb{R}^2), \qquad \text{for some constant } C \text{ independent of } u.$$

For the double-BV blind deblurring model of Chan and Wong [23], we impose three conditions:

Condition A. The observation $u_0 \in L^2(\mathbb{R}^2) \cap L^\infty(\mathbb{R}^2)$.
Condition B. The ideal image u belongs to $\mathrm{BV}(\mathbb{R}^2)$.
Condition C. The blur PSF k belongs to $\mathrm{BV}(\mathbb{R}^2)$, and satisfies the DC-condition: $\int_{\mathbb{R}^2} k(x)dx = 1$.

The L^2 constraint in Condition A naturally comes from the data model in (35), while the L^∞ constraint is satisfied by most real imaging devices and is convenient for mathematical analysis. Furthermore, according to the Poincaré inequality in Theorem 7, Condition B implies that $u \in L^2(\mathbb{R}^2)$. Then by Young's inequality [48], one has

$$\|k * u\|_{L^2(\mathbb{R}^2)} \leq \|k\|_{L^1(\mathbb{R}^2)} \|u\|_{L^2(\mathbb{R}^2)},$$

which makes the data fitting term in (35) finite and well defined.

7.3. *On the uniqueness: Hidden symmetries*

In what follows we reveal some special symmetries hidden in the double-BV deblurring model (35). Such symmetries could lead to nonuniqueness of solutions.

Theorem 8: (Image-PSF Uncertainty) Suppose (u_*, k_*) minimizes the double-BV deblurring model (35) with (α, β, λ). Assume in addition that

$$m = \int_{\mathbb{R}^2} u_*(x)dx = \beta/\alpha.$$

Then $(u_+, k_+) = (mk_*, u_*/m)$ must be a minimizer as well.

The proof is a straightforward verification. Now for any given $a = (a_1, a_2) \in \mathbb{R}^2$, define the shifting operator

$$S_a : g(x) \to S_a[g] = g(x-a), \quad \text{for any measurable function } g.$$

Then it is well known that shifting commutes with convolution:

$$S_a[k * u] = S_a[k] * u = k * S_a[u].$$

Furthermore, for any $u, k \in \mathrm{BV}(\mathbb{R}^2)$, the invariance holds:

$$\int_{\mathbb{R}^2} |DS_a[u]| = \int_{\mathbb{R}^2} |Du| \ \text{ and } \ \int_{\mathbb{R}^2} S_a[k]dx = \int_{\mathbb{R}^2} kdx,$$

which induces the following symmetry.

Theorem 9: (Dual-Translation Uncertainty) Suppose (u_*, k_*) minimizes the double-BV deblurring model (35). Then for any $a \in \mathbb{R}^2$, $(S_a[u], S_a[k])$ is also a minimizer.

In order to better understand the double-BV deblurring model, consider now an easier but intimately related model - the double-Sobolev blind deblurring model $E_2[u, k \mid u_0]$ given by

$$\frac{\alpha}{2}\int_{\mathbb{R}^2} |\nabla u|^2 dx + \frac{\beta}{2}\int_{\mathbb{R}^2} |\nabla k|^2 dx + \frac{\lambda}{2}\int_{\mathbb{R}^2} (k * u - u_0)^2 dx, \tag{37}$$

for which both u and k belong to the Sobolev space $H^1(\mathbb{R}^2)$.

The unitary Fourier transform of a function $g(x)$ on $\mathbb{R}^2$ is defined by

$$G(\omega) = G(\omega_1, \omega_2) = \int_{\mathbb{R}^2} g(x) e^{-i2\pi\omega\cdot x} dx.$$

Then the unitary property of Fourier transform gives:

$$\int_{\mathbb{R}^2} |G(\omega)|^2 d\omega = \int_{\mathbb{R}^2} |g(x)|^2 dx, \text{ and } \int_{\mathbb{R}^2} |\nabla g(x)|^2 dx = 4\pi^2 \int_{\mathbb{R}^2} \omega^2 |G(\omega)|^2 d\omega,$$

with $\omega^2 = |\omega|^2 = \omega_1^2 + \omega_2^2$. Notice that the Fourier transform of $k * u$ is given by a direct product $K(\omega)U(\omega)$. Therefore, in the Fourier domain, the double-Sobolev blind deblurring energy $E_2[u, k \mid u_0]$ becomes $E_2[U, K \mid U_0]$, which is simply given by

$$2\pi^2\alpha \int_{\mathbb{R}^2} \omega^2 |U(\omega)|^2 d\omega + 2\pi^2\beta \int_{\mathbb{R}^2} \omega^2 |K(\omega)|^2 d\omega + \frac{\lambda}{2}\int_{\mathbb{R}^2} |K(\omega)U(\omega) - U_0(\omega)|^2 d\omega. \tag{38}$$

The DC-condition now requires $K(0) = 1$. Furthermore, since u, k, and u_0 are all real, one requires that both U and K satisfy the conjugate condition

$$\bar{U}(\omega) = U(-\omega) \text{ and } \bar{K}(\omega) = K(-\omega), \qquad \omega \in \mathbb{R}^2. \tag{39}$$

This leads to a nonuniqueness theorem more general than Theorem 9.

Theorem 10: (Dual-Phase Uncertainty) Let $(u_*, k_*) \in H^1(\mathbb{R}^2) \times H^1(\mathbb{R}^2)$ be a minimizer to the double-Sobolev blind deblurring model (37). And let

$$\phi(\omega) : \mathbb{R}^2 \to \mathbb{R}, \quad \omega \to \phi(\omega)$$

be any real smooth phase factor that is odd: $\phi(-\omega) = -\phi(\omega)$. Then

$$(u_+, k_+) = \text{Inverse Fourier Transforms of } (U_*(\omega)e^{i\phi(\omega)}, K_*(\omega)e^{-i\phi(\omega)})$$

must be also a minimizer.

Proof: It is straightforward to verify on the Fourier domain that

$$E[u_+, k_+ \mid u_0] = E[u_*, k_* \mid u_0],$$

and that both u_+ and k_+ are indeed real. Furthermore, k_+ does satisfy the DC-condition since

$$\int_{\mathbb{R}^2} k_+ dx = K_+(0) = K_*(0)e^{-i\phi(0)} = K_*(0) = 1. \qquad \square$$

In particular, by taking $\phi(\omega) = a \cdot \omega = a_1\omega_1 + a_2\omega_2$, one recovers the dual-translation uncertainty stated in Theorem 9. For uniqueness, it is therefore desirable to impose further conditions to break up the potential symmetries.

7.4. *The existence theorem*

The Poincaré's inequality in Theorem 7 can be further improved by dropping off the L^1 condition [18].

Theorem 11: (Poincaré's Inequality) For any $u \in L^2(\mathbb{R}^2)$, the Poincaré inequality holds:

$$\|u\|_{L^2(\mathbb{R}^2)} \le C|Du|(\mathbb{R}^2), \quad \text{for some constant } C \text{ independent of } u.$$

The finiteness of the L^2-norm appears necessary due to counterexamples like $u \equiv 1$. The proof can be found in Chan and Shen [18].

Define the space BV_2 by

$$\mathrm{BV}_2(\mathbb{R}^2) = \{u \in L^2(\mathbb{R}^2) \mid |Du|(\mathbb{R}^2) < \infty\}.$$

Then by Theorem 7, $\mathrm{BV}(\mathbb{R}^2) \subset \mathrm{BV}_2(\mathbb{R}^2)$. The larger space BV_2 shall play a natural role for the blind deblurring model to be discussed below. We now study the existence of the double-BV blind deblurring model

$$E[u, k \mid u_0] = \alpha \int_{\mathbb{R}^2} |Du| + \beta \int_{\mathbb{R}^2} |Dk| + \frac{\lambda}{2} \int_{\mathbb{R}^2} (k * u - u_0)^2 dx. \tag{40}$$

The following conditions will be assumed for the study of existence.

Condition (a). Observation $u_0 \in L^2(\mathbb{R}^2) \cap L^\infty(\mathbb{R}^2)$.
Condition (b). Image $u \in \mathrm{BV}_2(\mathbb{R}^2)$, and $\|u\|_{L^\infty} \le \|u_0\|_{L^\infty}$.
Condition (c). PSF $k \in \mathrm{BV}(\mathbb{R}^2)$, nonnegative, and satisfies the DC-condition: $\langle k, 1\rangle = 1$.

Notice that the constraints put differently on u and k help break their symmetric role in the model. However, even under Condition (b) and (c), the dual-translation uncertainty addressed by Theorem 9 is still not got rid of, since both conditions are still translation invariant.

For this purpose, Chan and Wong adopted the following centrosymmetry condition [23] to break the symmetry:

Condition(d'). The PSF is centrosymmetric: $k(-x) = k(x)$.

It amounts to requiring that the blur operator K is Hermitian. Numerical evidences in [23] seem to suggest that this condition can stably lead to unique deblurring solutions, though the theory has not been explored.

Herein to restrict the PSF to be highly concentrated near the origin, we impose the condition on the "tail" behavior of k.

Condition(d). There exists some nonnegative function $F(x) \in L^1(\mathbb{R}^2)$, and some positive radius $R > 0$, so that

$$0 \leq k(x) \leq F(x), \quad \forall x \in \mathbb{R}^2 : \ |x| \geq R. \tag{41}$$

For example, if $F(x) \equiv 0$ for all $|x| \geq R$, Condition (d) amounts to requiring k to be compactly supported on the disk $B_R = \{x \in \mathbb{R}^2 : |x| < R\}$.

Theorem 12: (Existence of Double-BV Blind Deblurring) Under Conditions (a), (b), (c), and (d), the minimizers to the double-BV blind deblurring model (40) exist.

The more technical proof could be found in our recent monograph [18]. Computationally, the double-BV blind deblurring model (40) can be also implemented via the AM algorithm, similar to that described in Eqn. (33) for parametric blind deblurring. More computational details can be found in the work of Chan and Wong [23].

8. Deblurring Besov Images via Iterated Shrinkage

In this section, we introduce the iterated-shrinkage algorithm of Daubechies et al. [30,31] for wavelet-based image deblurring.

Shrinkage has been one of the most efficient algorithms for image denoising and compression due to its low complexity and simple implementation, as studied in the stochastic framework by Donoho and Johnstone [34,35], and also in the variational framework by DeVore, et al. [8,32,33]. For deblurring, a direct shrinkage scheme becomes infeasible due to the global spatial correlation induced by the blur (integral) operator. Consequently, Daubechies et al. in the aforementioned works developed the iterated-shrinkage algorithm to still take advantage of the efficiency of the shrinkage scheme.

8.1. *Wavelets and Besov images*

We first briefly introduce wavelets and Besov images on $\mathbb{R}^2$. The reader is referred to, e.g., [18,29,51,52,67,74] for more details on these topics.

Let $\phi(x) = \phi(x_1, x_2)$ denote the scaling function, assumed to be compactly supported and sufficiently smooth for simplicity. For example, one can take the tensor product $\phi(x_1) \otimes \phi(x_2)$ of a 1D scaling function by Daubechies' design [28,29]. Assume that the three canonical wavelets associated to the multiresolution analysis of ϕ are given by

$$\psi^{(0,1)}(x), \quad \psi^{(1,0)}(x), \ \text{ and } \ \psi^{(1,1)}(x).$$

In the tensor-product framework, these can similarly be constructed from a 1D scaling function and its associated canonical wavelet. The wavelets are similarly assumed to be compactly supported and sufficiently smooth. Let

$$t = (t_1, t_2) \in T = \{(0,1), (1,0), (1,1)\}$$

denote one of the three wavelet types. Assume that each $\psi^t(x)$ has been normalized to have a unit L^2 norm, and the associated multiresoltuion analysis is orthogonal (*biorthogonality* imposes no extra challenge).

For any triple index

$$\lambda = (j, n, t) = (j, (n_1, n_2), t) \in \mathbb{Z} \times \mathbb{Z}^2 \times T,$$

define the L^2 normalized copy

$$\psi_\lambda(x) = \psi^t_{j,n}(x) = 2^j \psi^t(2^j x - n).$$

Similarly, define $\phi_n(x) = \phi(x - n)$ for any $n \in \mathbb{Z}^2$. Then $L^2(\mathbb{R}^2)$ has the *homogenous* orthonormal wavelet basis:

$$\left\{\psi_\lambda(x) : \lambda = (j, n, t) \in \mathbb{Z} \times \mathbb{Z}^2 \times T\right\}. \tag{42}$$

For $\lambda = (j, n, t)$, one defines $|\lambda| = j$ to be the associated resolution index. For any *locally integral* image u, its wavelet coefficients are defined by

$$u_\lambda = u_{(j,n,t)} = \langle u, \psi_\lambda \rangle = \int_{\mathbb{R}^2} u(x)\psi_\lambda(x)dx, \qquad \lambda \in \mathbb{Z} \times \mathbb{Z}^2 \times T.$$

In 2D, the space of Besov images, $B_q^\alpha(L^p)$ with $\alpha > 0, q, p \geq 1$, can be equivalently defined by:

$$\|u\|_{B_q^\alpha(L^p)} = \|\langle u, \phi_\bullet \rangle\|_{l^p} + \left[\sum_{j \geq 0} 2^{jq(\alpha+1-2/p)} \|u_{(j,\bullet,\bullet)}\|_{l^p}^q \right]^{1/q} .$$

The *homogeneous* Besov semi-norm $|u|_{\dot{B}_q^\alpha(L^p)}$, on the other hand, can be characterized by merely the wavelet coefficients:

$$|u|_{\dot{B}_q^\alpha(L^p)} = \left[\sum_{j \in \mathbb{Z}} 2^{jq(\alpha+1-2/p)} \|u_{(j,\bullet,\bullet)}\|_{l^p}^q \right]^{1/q} .$$

One Besov space of particular interest to image processing is when $\alpha = 1$ and $p = q = 1$, for which the semi-norm takes the simple form of:

$$|u|_{\dot{B}_1^1(L^1)} = \sum_{j \in \mathbb{Z}} \|u_{(j,\bullet,\bullet)}\|_{l^1} = \sum_\lambda |u_\lambda|. \tag{43}$$

The BV image prior, which has been extensively employed in the previous sections, is closely related to the Besov class $B_1^1(L^1)$. Roughly speaking, BV is somewhere between $B_1^1(L^1)$ and a weaker version of $B_1^1(L^1)$, which is the remarkable result established by Cohen et al. [25,26]. Thus in the wavelet literature, the BV image prior has often been approximated by the $B_1^1(L^1)$ [8,21], which shall also be adopted herein.

8.2. *Besov image deblurring via iterated shrinkage*

Consider the linear blur model with a known blur K and additive Gaussian noises:

$$u^0(x) = K[u](x) + n(x), \qquad x = (x_1, x_2) \in \mathbb{R}^2.$$

Assume that the blur $K : L^2 \to L^2$ is bounded and with operator norm $\|K\| \leq 1$. This is always true for any shift-invariant blur with a PSF $k(x)$ that is nonnegative everywhere. Then by Young's inequality, one has

$$\|k * u\|_{L^2} \leq \|k\|_{L^1} \, \|u\|_{L^2}, \tag{44}$$

and consequently the operator norm $\|K\| \leq \|k\|_{L^1} = 1$ since k satisfies the lowpass condition $\int_{\mathbb{R}^2} k(x)dx = 1$ and is nonnegative.

By the general Bayesian/Tikhonov framework discussed earlier, if the prior image model is taken to be the Besov space $B_1^1(L^1)$, the deblurring model is then given by

$$\hat{u} = \text{argmin}_u \; E[u \mid u^0, K] = 2\alpha|u|_{\dot{B}_1^1(L^1)} + \beta \int_{\mathbb{R}^2} (K[u] - u^0)^2 dx, \tag{45}$$

where $\alpha, \beta > 0$ with β inversely proportional to the variance σ^2 of the noise, and α characterizing the sensitivity to signal roughness.

Under the wavelet basis, the deblurring energy takes the simple form of

$$E[u \mid u^0, K] = \sum_\lambda \left(2\alpha|u_\lambda| + \beta(K[u]_\lambda - u^0_\lambda)^2\right). \tag{46}$$

Unlike image denoising or compression for which $K = Id$ is the identity operator and the energy is completely decoupled, the present model is coupled across all the resolutions due to the blur operator K. This makes the classical wavelet shrinkage algorithms [34,35] not directly applicable.

Let $\delta u = \sum_\lambda \delta u_\lambda \psi_\lambda$ denote a perturbation. Then the first variation of the model energy $E[u \mid u^0, K]$ in (45) is given by

$$\begin{aligned} \delta E/2 &= \alpha \sum_\lambda \operatorname{sign}(u_\lambda)\delta u_\lambda + \beta \int_{\mathbb{R}^2} K^*[K[u] - u^0]\delta u dx \\ &= \sum_\lambda \left(\alpha \operatorname{sign}(u_\lambda) + \beta(M[u]_\lambda - g_\lambda)\right)\delta u_\lambda, \end{aligned} \tag{47}$$

where $M = K^*K$ and $g(x) = K^*[u^0]$. As a result, we have established the following theorem.

Theorem 13: The optimal deblurring must satisfy the system of equations:

$$0 = r \operatorname{sign}(u_\lambda) + (M[u]_\lambda - g_\lambda), \qquad \lambda \in \mathbb{Z} \times \mathbb{Z}^2 \times T, \tag{48}$$

where $r = \alpha/\beta$ could be considered as the noise-to-signal ratio as inspired by Wiener filtering and BV deblurring discussed earlier.

As in classical wavelet analysis [18,34], define the soft-shrinkage operator $S_r(t)$ by

$$S_r(t) = \operatorname{sign}(t)(|t| - r)^+ = \operatorname{sign}(t)\max(|t| - r, 0), \quad \text{for } t \in \mathbb{R}.$$

Then if there is no blur so that both K and M are the identity operator, the system of equilibrium equations (48) are then completely decoupled, and the optimal solution is directly given by

$$u_\lambda = S_r(g_\lambda), \quad \text{for } \lambda \in \mathbb{Z} \times \mathbb{Z}^2 \times T.$$

For deblurring, generally $M = K^*K$ is a mixing operator which could be sparse but is often not the identity matrix. Then Daubechies et al. [30,31] proposed the following iterated-shrinkage algorithm. Similar ideas also appeared in the variational-PDE literature for deblurring-related applications (see, e.g., Chan and Shen [17]).

To proceed, one first modifies that equilibrium system (48) to

$$0 = r\ \mathrm{sign}(u_\lambda) + (u_\lambda - g_\lambda) - (u_\lambda - M[u]_\lambda).$$

The the iterated-shrinkage algorithm of Daubechies et al. [30,31] is based on the following iteration scheme $u^k \to u^{k+1}$ at each time step k:

$$0 = r\ \mathrm{sign}(u_\lambda^{k+1}) + u_\lambda^{k+1} - \left(u_\lambda^k + g_\lambda - M[u^k]_\lambda\right). \tag{49}$$

Notice that due to the one-step time delay, the new system for u^{k+1} is *decoupled.* Furthermore, it takes the form of a blur-free denoising problem! Therefore, we have the following [30,31].

Theorem 14: At each time step k, the iteration is efficiently carried out by the shrinkage operator applied to each wavelet channel:

$$u_\lambda^{k+1} = S_r\left(u_\lambda^k + g_\lambda - M[u^k]_\lambda\right), \ \text{ for }\ \lambda \in \mathbb{Z} \times \mathbb{Z}^2 \times T.$$

8.3. *Understanding the iterated-shrinkage algorithm*

We now present two ways to better understand the above iterated-shrinkage algorithm of Daubechies et al. [30,31], from both the differential-equation and variational points of view.

8.3.1. *As semi-implicit time marching*

Suppose more generally that the equilibrium system is augmented to

$$0 = r\ \mathrm{sign}(u_\lambda) + (Au_\lambda - g_\lambda) - (Au_\lambda - M[u]_\lambda), \tag{50}$$

for some constant $A \gg 1$. Then the iteration algorithm is given by

$$0 = r\ \mathrm{sign}(u_\lambda^{k+1}) + Au_\lambda^{k+1} - (Au_\lambda^k + g_\lambda - M[u^k]_\lambda), \tag{51}$$

which again allows an explicit shrinkage solution. Suppose $\Delta t = A^{-1} \ll 1$. Then the last equation can be rewritten to

$$\frac{u_\lambda^{k+1} - u_\lambda^k}{\Delta t} = -\left(r\ \mathrm{sign}(u_\lambda^{k+1}) + (M[u^k]_\lambda - g_\lambda)\right). \tag{52}$$

If one introduces the continuous time variable t so that

$$u_\lambda^k = u_\lambda(t = k\Delta t), \qquad k = 0, 1, \cdots.$$

Then the iteration (52) is precisely a semi-implicit scheme for the infinite system of coupled ordinary differential equations:

$$\frac{d}{dt}u_\lambda(t) = -\left(r\ \mathrm{sign}(u_\lambda(t)) + (M[u(t)]_\lambda - g_\lambda)\right), \qquad \lambda \in \mathbb{Z} \times \mathbb{Z}^2 \times T,$$

where the right hand side is precisely the negative gradient $-\frac{1}{2}\frac{\partial E}{\partial u_\lambda}$ by (47).

Therefore, in the limit of $A \gg 1$, or equivalently, $\Delta t = A^{-1} \ll 1$, the iterated shrinkage algorithm of Daubechies et al. [30,31] can be considered as a semi-implicit scheme for gradient-descent time marching for the deblurring energy $E[u \mid u^0, K]$.

8.3.2. *Via augmentation and auxiliary variables*

The second way to understand the iterated-shrinkage algorithm is via the variational method on auxiliary variables and augmented functionals [30,31].

If one introduces an auxiliary variable z, which is considered as a delayed version of the target image u during the iteration, then the system (50) can be rewritten to

$$0 = r \operatorname{sign}(u_\lambda) + A(u_\lambda - z_\lambda) + M[z]_\lambda - g_\lambda. \tag{53}$$

Consequently, the iterated-shrinkage algorithm (51) can be considered as the solution to the above equation given $z = u^k$.

This motivates one to introduce an augmented energy $E[u, z \mid u^0, K]$ whose conditional energy $E[u \mid z, u^0, K]$ is in the integral form of (53):

$$E[u \mid z, u^0, K] = 2r|u|_{\dot{B}^1_1(L^1)} + A\|u - z\|^2 + 2\langle M[z] - g, u\rangle,$$

where both the norm and inner product are in the L^2 sense. Then given z, the system (53) yields the optimal u for $E[u, \mid z, u^0, K]$.

As a result, the full augmented energy must be given in the form of

$$E[u, z \mid u^0, K] = E[u \mid z, u^0, K] + \Phi[z \mid u^0, K],$$

where the functional Φ is independent of u.

We look for the specific form of Φ, such that (i) the iterated-shrinkage algorithm (51) corresponds to the AM (alternating-minimization) algorithm for the augmented energy $E[u, z \mid u^0, K]$; and (ii)

$$E[u, z \mid u^0, K] \geq \beta^{-1} E[u \mid u^0, K], \tag{54}$$

and the equality (or minimum) holds when $z = u$. The equality condition leads to

$$2\langle M[u] - g, u\rangle + \Phi[u \mid u^0, K] = \|K[u] - u^0\|^2.$$

Since $\langle M[u] - g, u\rangle = \langle K[u] - u^0, Ku\rangle$, this gives explicitly

$$\Phi[z \mid u^0, K] = \|K[z] - u^0\|^2 - 2\langle K[z] - u^0, Kz\rangle = -\langle K[z] - u^0, K[z] + u^0\rangle.$$

Notice that

$$\begin{aligned} & 2\langle M[z]-g,u\rangle - \langle K[z]-u^0, K[z]+u^0\rangle \\ =& 2\langle K[z]-u^0, Ku\rangle - \langle K[z]-u^0, K[z]+u^0\rangle \\ =& -\langle K[z],K[z]\rangle + 2\langle K[z],K[u]\rangle - 2\langle K[u],u^0\rangle + \langle u^0,u^0\rangle \\ =& -\|K[z]-K[u]\|^2 + \|K[u]-u^0\|^2 \\ =& -\|K[u-z]\|^2 + \|K[u]-u^0\|^2. \end{aligned}$$

Therefore, the augmented energy is ultimately given by

$$E[uz \mid u^0, K] = 2r|u|_{\dot{B}^1_1(L^1)} + A\|u-z\|^2 - \|K[u-z]\|^2 + \|K[u]-u^0\|^2. \quad (55)$$

Since $A \gg 1$ (and in particular $A \geq 1$) and the operator norm $\|K\| \leq 1$ as explained in (44), the augmented energy is indeed bounded below by $\beta^{-1}E[u \mid u^0, K]$ as required in (54). To conclude, we have the following theorem.

Theorem 15: The iterated-shrinkage algorithm for $E[u \mid u^0, K]$ of Daubechies et al. [30,31] is exactly the AM algorithm for the augmented energy $E[u, z \mid u^0, K]$. In particular, the algorithm must be stable and satisfy the monotone condition $E[u^{k+1} \mid u^0, K] \leq E[u^k \mid u^0, K]$.

9. Further Reading

For the several more involved proofs that have been left out, we refer the reader to our recent monograph [18]. For readers who are interested in this area, we also recommend to explore and read about other methodologies or related works, for example, the recursive inverse filtering (RIF) technique of Richardson [59] and Lucy [49] arising from astronomy imaging, as well as numerous works by other active researchers such as James Nagy et al. [55], Chan, Chan, Shen, and Shen [10] on wavelet deblurring via spatially varying filters, and Kindermann, Osher, and Jones [44] on nonlocal deblurring.

Acknowledgements

The authors would like to acknowledge the Institute for Mathematical Sciences, National University of Singapore, for her generous support to the visiting of the two authors, both financially and academically. The current work has also been partially supported by the National Science Foundation (NSF) of USA under grant numbers DMS-9973341 (Chan) and DMS-0202565 (Shen), the Office of Naval Research (ONR) of USA under grant number N00014-03-1-0888 (Chan), as well as the National Institute of Health (NIH) of USA under grant number P20MH65166 (Chan). We are also very thankful to Yoon-Mo Jung for his careful proofreading of the manuscript and numerous suggestions and feedbacks.

References

1. R. Acar and C. R. Vogel. Analysis of total variation penalty methods for ill-posed problems. *Inverse Prob.*, 10:1217–1229, 1994.
2. L. Ambrosio and V. M. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. *Comm. Pure Appl. Math.*, 43:999–1036, 1990.
3. L. Ambrosio and V. M. Tortorelli. On the approximation of free discontinuity problems. *Boll. Un. Mat. Ital.*, 6-B:105–123, 1992.
4. M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-Stokes, fluid dynamics, and image and video inpainting. *IMA Preprint 1772 at: www.ima.umn.edu/preprints/jun01*, June, 2001.
5. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. *Computer Graphics, SIGGRAPH 2000*, July, 2000.
6. P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues.* Springer-Verlag New York, Inc., 1998.
7. A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imag. Vision*, 20:89–97, 2004.
8. A. Chambolle, R. A. DeVore, N.-Y. Lee, and B. J. Lucier. Nonlinear wavelet image processing: variational problems, compression and noise removal through wavelet shrinkage. *IEEE Trans. Image Processing*, 7(3):319–335, 1998.
9. A. Chambolle and P. L. Lions. Image recovery via Total Variational minimization and related problems. *Numer. Math.*, 76:167–188, 1997.
10. R. Chan, T. Chan, L. Shen, and Z. Shen. Wavelet deblurring algorithms for sparially varying blur from high-resolution image reconstruction. *Linear. Alg. Appl.*, 366:139–155, 2003.
11. T. F. Chan, S.-H. Kang, and J. Shen. Euler's elastica and curvature based inpainting. *SIAM J. Appl. Math.*, 63(2):564–592, 2002.
12. T. F. Chan, S. Osher, and J. Shen. The digital TV filter and nonlinear denoising. *IEEE Trans. Image Process.*, 10(2):231–241, 2001.
13. T. F. Chan and J. Shen. Nontexture inpainting by curvature driven diffusions (CDD). *J. Visual Comm. Image Rep.*, 12(4):436–449, 2001.
14. T. F. Chan and J. Shen. Bayesian inpainting based on geometric image models. *in Recent Progress in Comput. Applied PDEs, Kluwer Academic, New York*, pages 73–99, 2002.
15. T. F. Chan and J. Shen. Inpainting based on nonlinear transport and diffusion. *AMS Contemp. Math.*, 313:53–65, 2002.
16. T. F. Chan and J. Shen. Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.*, 62(3):1019–1043, 2002.
17. T. F. Chan and J. Shen. On the role of the BV image model in image restoration. *AMS Contemp. Math.*, 330:25–41, 2003.

18. T. F. Chan and J. Shen. *Image Processing and Analysis: variational, PDE, wavelet, and stochastic methods.* SIAM Publisher, Philadelphia, 2005.
19. T. F. Chan and J. Shen. Variational image inpainting. *Comm. Pure Applied Math.*, 58:579–619, 2005.
20. T. F. Chan, J. Shen, and L. Vese. Variational PDE models in image processing. *Notices Amer. Math. Soc.*, 50:14–26, 2003.
21. T. F. Chan, J. Shen, and H.-M. Zhou. Total variation wavelet inpainting. *J. Math. Imag. Vision, to appear*, 2005.
22. T. F. Chan and L. A. Vese. A level set algorithm for minimizing the Mumford-Shah functional in image processing. *IEEE/Computer Society Proceedings of the 1st IEEE Workshop on "Variational and Level Set Methods in Computer Vision"*, pages 161–168, 2001.
23. T. F. Chan and C. K. Wong. Total variation blind deconvolution. *IEEE Trans. Image Process.*, 7(3):370–375, 1998.
24. D. Chandler. *Introduction to Modern Statistical Mechanics.* Oxford University Press, New York and Oxford, 1987.
25. A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore. Harmonic analysis of the space BV. *Revista Matematica Iberoamericana*, 19:235–263, 2003.
26. A. Cohen, R. DeVore, P. Petrushev, and H. Xu. Nonlinear approximation and the space BV(R^2). *Amer. J. Math.*, 121:587–628, 1999.
27. T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., New York, 1991.
28. I. Daubechies. Orthogonal bases of compactly supported wavelets. *Comm. Pure. Appl. Math.*, 41:909–996, 1988.
29. I. Daubechies. *Ten lectures on wavelets.* SIAM, Philadelphia, 1992.
30. I. Daubechies, M. Defrise, and C. DeMol. An iterative thresholding algorithm for lienar inverse problems with a sparsity constraint. *to appear in Comm. Pure Applied Math.*, 2005.
31. I. Daubechies and G. Teschke. Variational image restoration by means of wavelets: Simultaneous decomposition, deblurring, and denoising. *Appl. Comput. Harmon. Anal.*, 19(1):1–16, 2005.
32. R. A. DeVore, B. Jawerth, and B. J. Lucier. Image compression through wavelet transform coding. *IEEE Trans. Information Theory*, 38(2):719–746, 1992.
33. R. A. DeVore, B. Jawerth, and V. Popov. Compression of wavelet coefficients. *Amer. J. Math.*, 114:737–785, 1992.
34. D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Information Theory*, 41(3):613–627, 1995.
35. D. L. Donoho and I. M. Johnstone. Ideal spacial adaption by wavelet shrinkage. *Biometrika*, 81:425–455, 1994.
36. S. Esedoglu and J. Shen. Digital inpainting based on the Mumford-Shah-Euler image model. *European J. Appl. Math.*, 13:353–370, 2002.
37. L. C. Evans. *Partial Differential Equations.* Amer. Math. Soc., 1998.
38. L. C. Evans and R. F. Gariepy. *Measure Theory and Fine Properties of Functions.* CRC Press, Inc., 1992.
39. D. Fish, A. Brinicombe, and E. Pike. Blind deconvolution by means of the

richardsonlucy algorithm. *J. Opt. Soc. Am. A*, 12:58–65, 1996.

40. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721–741, 1984.
41. W. Gibbs. *Elementary Principles of Statistical Mechanics.* Yale University Press, 1902.
42. E. Giusti. *Minimal Surfaces and Functions of Bounded Variation.* Birkhäuser, Boston, 1984.
43. A. D. Hillery and R. T. Chin. Iterative Wiener filters for image restoration. *IEEE Trans. Signal Processing*, 39:1892–1899, 1991.
44. S. Kindermann, S. Osher, and P. W. Jones. Deblurring and denoising of images by nonlocal functionals. *UCLA CAM Tech. Report*, 04-75, 2004.
45. D. C. Knill and W. Richards. *Perception as Bayesian Inference.* Cambridge Univ. Press, 1996.
46. R. L. Lagendijk, A. M. Tekalp, and J. Biemond. Maximum likelihood image and blur identification: A unifying approach. *Opt. Eng.*, 29:422–435, 1990.
47. J. S. Lee. Digital image enhancement and noise filtering by use of local statistics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2:165–168, 1980.
48. E. H. Lieb and M. Loss. *Analysis.* Amer. Math. Soc., second edition, 2001.
49. L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astron. J.*, 79:745–754, 1974.
50. F. Malgouyres. Mathematical analysis of a model which combines total variation and wavelet for image restoration. *Journal of information processes*, 2(1):1–10, 2002.
51. S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1998.
52. Y. Meyer. *Wavelets and Operators.* Cambridge University Press, 1992.
53. D. Mumford. *Geometry Driven Diffusion in Computer Vision*, chapter "The Bayesian rationale for energy functionals", pages 141–153. Kluwer Academic, 1994.
54. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Applied. Math.*, 42:577–685, 1989.
55. J. G. Nagy. A detailed publication list is available at the URL: http://www.mathcs.emory.edu/˜nagy/research/pubs.html. Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA.
56. A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice Hall Inc., New Jersey, 1989.
57. S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision and Graphics.* Springer Verlag, 2002.
58. S. Osher and J. Shen. Digitized PDE method for data restoration. In G. A. Anastassiou, editor, *Analytical-Computational Methods in Applied Mathematics.* Chapman & Hall/CRC, FL, 2000.
59. W. H. Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.*, 62:55–59, 1972.

60. L. Rudin and S. Osher. Total variation based image restoration with free local constraints. *Proc. 1st IEEE ICIP*, 1:31–35, 1994.
61. L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
62. J. Shen. On the foundations of vision modeling I. Weber's law and Weberized TV (total variation) restoration. *Physica D: Nonlinear Phenomena*, 175:241–251, 2003.
63. J. Shen. Bayesian video dejittering by BV image model. *SIAM J. Appl. Math.*, 64(5):1691–1708, 2004.
64. J. Shen. Piecewise $H^{-1} + H^0 + H^1$ images and the Mumford-Shah-Sobolev model for segmented image decomposition. *Appl. Math. Res. Exp.*, 4:143-167, 2005.
65. G. Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, MA, 1993.
66. G. Strang. *Introduction to Linear Algebra.* Wellesley-Cambridge Press, 3rd edition, 1998.
67. G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, Wellesley, MA, 1996.
68. R. Strichartz. *A Guide to Distribution Theory and Fourier Transforms.* CRC Press, Ann Arbor, MI, USA, 1994.
69. A. N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math. Dokl.*, 4:1624–1627, 1963.
70. L. A. Vese. A study in the BV space of a denoising-deblurring variational problem. *Appl. Math. Optim.*, 44(2):131–161, 2001.
71. L. A. Vese and S. J. Osher. Modeling textures with Total Variation minimization and oscillating patterns in image processing. *J. Sci. Comput.*, 19:553–572, 2003.
72. C. R. Vogel. *Computational Methods for Inverse Problems.* SIAM, Philadelphia, 2002.
73. C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comput.*, 17(1):227–238, 1996.
74. P. Wojtaszczyk. *A Mathematical Introduction to Wavelets.* London Mathematical Society Student Texts 37. Cambridge University Press, 1997.
75. Y. You and M. Kaveh. A regularization approach to joint blur identification and image restoration. *IEEE Transactions on Image Processing*, 5:416–428, 1996.

DATA HIDING – THEORY AND ALGORITHMS

Pierre Moulin[1] and Ralf Koetter[2]
University of Illinois
Beckman Institute, Coordinated Science Lab and ECE Department
Urbana, IL 61801, USA
E-mails: [1]moulin@ifp.uiuc.edu, [2]koetter@comm.csl.uiuc.edu

This paper reviews the fundamentals of the data hiding problem. Data hiding can be viewed as a game between two teams (embedder/decoder *vs* attacker), and optimal data-hiding and attack strategies may be developed in this context. This paper presents a framework for developing such strategies as well as practical codes. The theory is applied to image watermarking examples.

1. Introduction

Watermarking and data hiding are now major research areas in signal, image and video processing [13]. The goal is to conceal information (such as copyright information, annotations, movie subtitles, secret data, etc.) within a host data set. This hidden information should be decodable even if the watermarked data are modified (to some extent) by an adversary (attacker).

Beginning around 1990, a variety of watermarking and data hiding algorithms have been proposed in the literature, with mixed success. Many algorithms were based on heuristics and were unable to resist simple attacks. In the second part of the 1990's, it was realized that information theory plays a natural role in watermarking and data hiding, due to the need to reliably communicate information to a receiver. This theory also guides the development of good data hiding codes. The main challenge was to formulate a precise mathematical framework capturing the essential features of watermarking and data hiding problems:

(1) The watermarking process should introduce limited distortion in the

host signal. Likewise, the attack should introduce limited distortion.

(2) While the host signal is known to the information embedder, it may be unknown to the decoder (blind watermarking/data hiding). Additional side information (such as a cryptographic key) may be shared by the encoder and decoder.

(3) The communication channel is under control of the attacker.

The essential ingredients of the information-theoretic approach are as follows:

(1) Distortion metrics are used to define a broad class of admissible embedding functions and a class of attack channels.

(2) Statistical models for the host data, the message, and the secret key are used to meaningfully define probabilities of error.

(3) Reliable communication is sought under any attack channel in the prescribed class [24]. Game theory plays a natural role under this setup: one party (embedder/decoder team) tries to minimize probability of error, and the other party (attacker) tries to maximize it.

There is a cost in restricting the class of embedding functions, and a danger in restricting the class of attacks. In the first case, performance of the watermarking/data hiding system may be unnecessarily low – in particular, we shall see why spread-spectrum systems [12] are generally not competitive with quantization-based systems [4]. In the second case, performance of the watermarking/data hiding system may be catastrophic – for instance because the embedding algorithm was designed to resist white noise attacks, but not geometric attacks.

Application of information theory has revealed the following fundamental concept, which until early 1999 [4,14,39] had been overlooked in the watermarking literature: Even if the host signal S^N is not available at the decoder (*blind watermarking*), the fact that the encoder knows S^N signifies that achievable rates are higher than if S^N was some unknown interference. (Spread-spectrum systems do not exploit that property.) The watermarking problem falls in the category of communication problems where encoder and decoder have access to side information [11,21,10].

This paper begins with a brief overview of the data-hiding problem (Sec. 2). This is followed by a description of simple but instructive data-hiding coding techniques (Sec. 3) and more modern codes based on information-theoretic binning concepts (Secs. 4 and 5). Next we present a statistical analysis of these schemes (Sec. 6) and capacity analyses (Secs. 7

and 8). Desynchronization attacks are considered in Sec. 9. Applications to images are presented in Sec. 10. The authentication problem is reviewed in Sec. 11, and the paper concludes with a discussion in Sec. 12.

Notation. We use capital letters to denote random variables, small letters to denote their individual values, calligraphic fonts to denote sets, and a superscript N to denote length-N vectors, e.g., $x^N = (x_1, x_2, \cdots, x_N)$. We denote by $p(x), x \in \mathcal{X}$, the probability mass function (p.m.f.) of a random variable X taking its values in the set $\mathcal{X}$. The notation $Pr[\mathcal{E}]$ denotes the probability of an event $\mathcal{E}$, and the symbol $\mathbb{E}$ denotes mathematical expectation. The Gaussian distribution is denoted by $\mathcal{N}(\mu, \sigma^2)$.

2. Model for Data Hiding

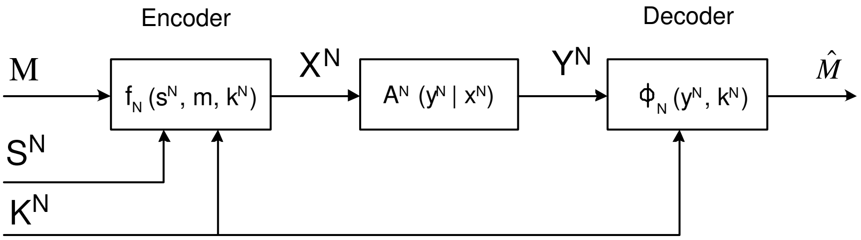


Fig. 1. Formulation of information hiding as a communication problem.

Referring to Fig. 1 [39], assume that a message M is to be embedded in a length-N host-data sequence $S^N = (S_1, \cdots, S_N)$: typically data from an host image, video, or audio signal. Side information $K^N = (K_1, \cdots, K_N)$ (possibly correlated with S^N) is shared by the encoder and decoder. The watermarked data $X^N = (X_1, \cdots, X_N)$ are subject to attacks that attempt to remove any trace of M from the modified data $Y^N = (Y_1, \cdots, Y_N)$. The mapping from X^N to Y^N is generally stochastic and is represented by a p.m.f. $A^N(y^N|x^N)$. The decoder has access to Y^N and K^N and produces an estimate $\hat{M}$ for the message that was originally transmitted. A decoding error occurs if $\hat{M} \neq M$. The variables S_i, K_i, X_i, Y_i take their values in sets $\mathcal{S}, \mathcal{K}, \mathcal{X}$ and $\mathcal{Y}$, respectively.

Data Embedding. Consider a bounded distortion function $d_1 : \mathcal{S} \times \mathcal{X} \to \mathbb{R}^+$. The distortion function is extended to a distortion on N–vectors by $d_1^N(s^N, x^N) = \frac{1}{N}\sum_{k=1}^N d_1(s_k, x_k)$. A *rate-R, length-N watermarking code subject to distortion* D_1 is defined as a triple $(\mathcal{M}, f_N, \phi_N)$, where:

- $\mathcal{M}$ is the message set of cardinality $|\mathcal{M}| = \lceil 2^{NR} \rceil$;
- $f_N : \mathcal{S}^N \times \mathcal{M} \times \mathcal{K}^N \to \mathcal{X}^N$ is the encoder which produces the sequence $x^N = f_N(s^N, m, k^N)$. The mapping f_N is subject to the average distortion constraint
$$\mathbb{E}[d_1^N(S^N, X^N)] \leq D_1; \tag{1}$$
- $\phi_N : \mathcal{Y}^N \times \mathcal{K}^N \to \mathcal{M}$ is the decoder which produces the decoded message $\hat{m} = \phi_N(y^N, k^N)$.

The definition of the distortion constraint (1) involves an averaging with respect to the distribution $p(s^N, k^N)$ and with respect to a uniform distribution on the messages. An alternative is to replace (1) with *almost-sure (a.s.) distortion constraints* [9,42,43]:

$$Pr[d_1^N(s^N, X^N) \leq D_1] = 1, \quad \forall s^N \in \mathcal{S}^N. \tag{2}$$

Attacker. Consider a distortion function $d_2 : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$. An *attack channel with memory, subject to distortion* D_2, is defined as a sequence of conditional p.m.f.'s $A^N(y^N|x^N)$ from $\mathcal{X}^N$ to $\mathcal{Y}^N$, such that

$$\mathbb{E}[d_2^N(X^N, Y^N)] \leq D_2. \tag{3}$$

In addition to distortion constraints, other restrictions may be imposed on the attack channels. For instance, for analysis purposes, the attack channel may be constrained to be memoryless, or blockwise memoryless. Denote by $\mathcal{A}^N$ the class of attack channels considered. Two alternatives to the average distortion constraint (3) are a.s. constraints:

$$Pr[d_2^N(x^N, Y^N) \leq D_2] = 1, \quad \forall x^N \in \mathcal{X}^N, \tag{4}$$

and an average distortion constraint with respect to the host data:

$$\mathbb{E}[d_2^N(S^N, Y^N)] \leq D_2. \tag{5}$$

(The attacker is assumed to know f_N and all probability distributions.)

Decoder. If the decoder knows the attack channel A^N, it can implement the Maximum a Posteriori (MAP) decoding rule, which minimizes the probability of error [40]:

$$\hat{m} = \underset{m \in \mathcal{M}}{\operatorname{argmax}}\, p(m|y^N, k^N). \tag{6}$$

If the decoder does not know A^N, one needs a *universal decoder* for the class $\mathcal{A}^N$ [24], with guaranteed performance level for all $A^N \in \mathcal{A}^N$, see Sec. 7. Heuristic decoding rules (such as the correlation rules and normalized correlation rules that are often used in the watermarking literature) might be severely suboptimal against an astute attacker.

3. Early Codes

The first papers on data hiding appeared in the early 1990's. The ideas proposed during that period include least significant bit (LSB) embedding techniques, which are elementary and nonrobust against noise. They are however closely related to more advanced binning techniques. The period 1995-1998 saw the development of spread-spectrum modulation (SSM) codes [12]. Both SSM and LSB methods are reviewed next.

3.1. *Spread-spectrum codes*

The watermarking problem presents similarities with the problem of communication in presence of a jammer. This has motivated many researchers to apply techniques from this branch of the communications literature, which was greatly developed during the 1980's. SSM techniques have been especially popular. We first briefly review these techniques and then show how they can be applied to watermarking and data hiding.

The jamming problem. In a standard radio or TV communication system, the transmitter sends a signal in a relatively narrow frequency band. This technique would be inappropriate in a communication problem with a jammer, because the jammer would allocate all his power to that particular band of frequencies. A SSM system therefore allocates secret sequences (with a broad frequency spectrum) to the transmitter, which sends data by modulating these sequences. The receiver demodulates the data using a filter matched to the secret sequences. Essentially the transmitter is communicating information over a secret low-dimensional subspace; only noise components in that subspace may affect communication performance. The jammer must spread his power over a broad frequency range, but only a small fraction of that power will have an effect on communication performance.

The application of SSM to data hiding is illustrated in Fig. 2. Associated with each message m and secret key k is a pattern $p^{(m,k)}$ which is "mixed" with the host s^N to form the marked signal x^N. Each pattern is typically a pseudo-random noise (PRN) sequence. The mixing could be as simple as a weighted addition:

$$x_n = s_n + \gamma p_n^{(m,k)}, \quad 1 \le n \le N \tag{7}$$

where γ is a strength parameter, which depends on the embedding distortion allowed. The mean-square embedding distortion is $ND_1 = \gamma^2 \|p^{(m,k)}\|^2$ and is usually the same for all m and k. The marked signal x^N is possibly corrupted by the attacker's noise, which produces a degraded signal

$$y^N = x^N + w^N. \tag{8}$$

The receiver knows the secret key k and can match y^N with the $|\mathcal{M}|$ waveforms $p^{(m,k)}$. If the host is not available to the receiver, the matching could be a simple correlation:

$$\hat{m} = \underset{m \in \mathcal{M}}{\operatorname{argmax}}\, t_m(y^N, k) \tag{9}$$

where

$$t_m(y^N, k) = \sum_{n=1}^{N} y_n\, p_n^{(m,k)}, \quad m \in \mathcal{M} \tag{10}$$

are the correlation statistics. If the host is available to the receiver, performance can be improved (see discussion at the end of this section) by subtracting the host from the data before correlating with the watermark patterns:

$$t_m(y^N, s^N, k) = \sum_{n=1}^{N} (y_n - s_n)\, p_n^{(m,k)}, \quad m \in \mathcal{M}.$$

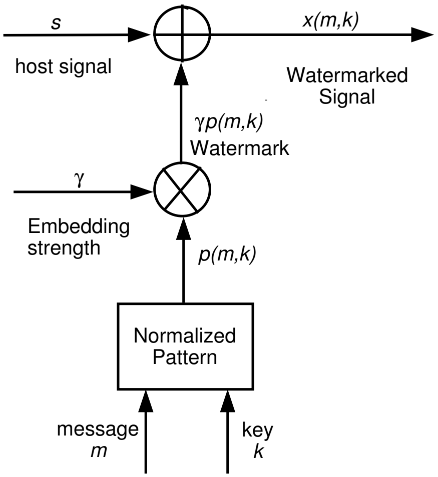


Fig. 2. Application of SSM to data hiding.

Several important refinements of the basic system of Fig. 2 have been developed over the years. Embedding distortion can be locally adapted to host signal characteristics, e.g., (7) can be replaced with

$$x_n = s_n + \gamma_n(s^N)\, p_n^{(m,k)} \tag{11}$$

where γ_n depends on the local characteristics of the host (e.g., frequency and temporal characteristics) [13,46].

Moreover, the basic correlator decoder (9) is often not well matched to noise statistics. An exception arises when the noise is white and Gaussian. Then the correlation statistic is a *sufficient statistic* [40], and the correlator decoder is ideal. For colored Gaussian noise however, a weighted correlation statistic is ideal. With non-Gaussian noise such as impulsive noise, the performance of a correlator decoder can be quite poor.

For the blind data hiding case, due to (7) and (8) we can write the received data as the sum of the watermark $\gamma p^{(m,k)}$ and total noise $s^N + w^N$:

$$y^N = \gamma p^{(m,k)} + (s^N + w^N).$$

Typically the host signal s^N has high energy relative to the embedding and attack distortions. As we shall see in Sec. 6, the performance of the decoder is limited by the high total noise level. For private data hiding, the decoder knows s, so the noise at the decoder is just w.

Authentication. It was assumed in (8) and (9) that the signal y^N submitted to the decoder is marked using one of the patterns $p^{(m,k)}$. The decoder can however be modified to account for the possible presence of an unmarked signal at the decoder. The expected value of the correlation statistics $t_m(y^N, k)$ is normally zero in that case. A positive threshold T is selected and the decoder returns

$$\hat{m} = \begin{cases} \text{argmax}_{m\in\mathcal{M}}\ t_m(y^N, k) & \text{if } \max_{m\in\mathcal{M}}\ t_m(y^N, k) \geq T \\ 0 & \text{else} \end{cases} \tag{12}$$

where $\hat{m} = 0$ indicates that no watermark was detected.

3.2. *LSB codes*

An early form of data hiding for grayscale images is based on LSB (Least Significant Bit) embedding techniques. These codes are rudimentary binning schemes.

The method is applicable to host signals of the form $s^N = \{s_1, s_2, \cdots, s_N\}$, where each sample s_i is encoded using b bits representing the natural binary decomposition of an integer between 0 and $2^b - 1$. For instance, s_i could represent one of the 256 intensity levels of a monochrome image, such as 77 = (01001101); the LSB is 1 in this case. The LSB plane is the length-N binary sequence made of all the LSB's. The LSB's can be changed without adversely affecting signal quality, and so LSB embedding methods simply replace the LSB plane with an information sequence; the information rate is 1 bit per sample of s^N. The payload could be increased by replacing the second LSB with an information sequence as well, but this would increase embedding distortion.

Note that the value of b (i.e., the range of host signal amplitudes) is immaterial here. The LSB embedding scheme is capable of rejecting host-signal interference. Unfortunately, LSB embedding does not survive modest amounts of noise. For instance, an attacker could simply randomize the LSB plane, effectively destroying the hidden information that was originally embedded there.

4. Binning Schemes: General Principles

Binning is an important information-theoretic technique. It is an ideal technique for encoding data with side information at the transmitter only, as well as for decoding data with side information at the decoder only [11,21]. Since blind data hiding is an instance of the former problem, we provide an overview of binning in this section. We begin with two examples.

Example 1. Let S be a length-3 binary sequence. There are eight such sequences: $000, 001, \cdots, 111$, all assumed equally likely. We want to embed information into S, producing a new sequence X. The embedding is subject to a distortion constraint: S and X may differ in at most one position. We transmit X to a receiver which should decode the embedded information without knowing the original S.
Question 1: How many bits of information can we embed in S?
Question 2: How can we design an appropriate encoding/decoding scheme?

Answer. Under the distortion constraint, the original S can be modified in at most four ways: $S \oplus X \in \{000, 001, 010, 110\}$, so *at most* two bits of information can be embedded. Straightforward spread-spectrum ideas don't work at all in this case: simply adding (modulo 2) one of the four patterns above to S conveys no information about the message to the receiver. Instead, we can communicate *exactly* two bits of information using the scheme depicted in Table 1. The eight possible sequences X are partitioned into four bins (column of the 2×4 array). Each bin corresponds to one of the 2-bit information sequences we want to communicate. Given an arbitrary sequence S and an arbitrary information sequence M, we look in bin M for the sequence closest to S in the Hamming sense, and declare that sequence to be X. For instance, if $S = 010$ and $M = 01$, we have to choose between the two sequences 001 and 110 in bin M. The latter is closer to S and is thus declared to be X. In Table 1, the four choices of X corresponding to the four possible messages (with $S = 010$) have been boxed. The decoder observes X and simply outputs the corresponding bin index m. Observe that:

(1) in any given bin, the two candidates X are maximally distant (Hamming distance = 3);
(2) in any given bin, there is always one sequence that satisfies the embedding distortion constraint;
(3) the receiver can decode the information bits *without error.*
(4) The codewords in column m of the array are obtained by adding m to the codewords in column 00.

Table 1. A simple binning scheme: embedding a length-2 binary message m into a length-3 binary sequence S, in a way that modifies at most one bit of S.

	$m = 00$	$m = 01$	$m = 10$	$m = 11$
$x =$	$\boxed{000}$	001	$\boxed{010}$	$\boxed{011}$
	111	$\boxed{110}$	101	100

Example 2. (Model for watermarking of grayscale images, where grayscale modifications are not allowed to exceed 1.) Let $\mathcal{S} = \{0, 1, \cdots, 2^b - 1\}$, and partition this set into the subset $\mathcal{S}_e = \{0, 2, \cdots, 2^b - 2\}$ of even integers and the subset $\mathcal{S}_o = \{1, 3, \cdots, 2^b - 1\}$ of odd integers. Let S^N be a length-N sequence in $\mathcal{S}^N$. Here the marked sequence X should satisfy $|X_i - S_i| \leq 1$ (addition is modulo 2^b) for $1 \leq i \leq N$. Denote by $m = \{m_1, \cdots, m_N\}$ a binary sequence to be embedded into s^N. Consider the LSB code of Sec. 3.2, which can be formulated as

$$x_i = m_i + 2 \left\lfloor \frac{s_i}{2} \right\rfloor, \quad 1 \leq i \leq N.$$

This code selects $x_i \in \mathcal{S}_e$ if $m_i = 0$, and $x_i \in \mathcal{S}_o$ if $m_i = 1$. Thus we may view $\mathcal{S}_e$ and $\mathcal{S}_o$ as two bins from which we select x_i depending on the value of m_i.

In both examples, if the decoder does not have access to the marked sequence X, but to a degraded sequence $Y = X + W$, there will be decoding errors. However one can construct binning schemes that offer protection against errors, as elaborated in Sec. 5.

5. Quantization-Based Codes

In 1999, Chen and Wornell introduced a class of data-hiding codes known as dither modulation codes, also referred to as quantization-index modulation (QIM) codes [3,4]. These methods embed signal-dependent watermarks using quantization techniques. They turn out to be binning schemes and are related to work from the early 1980's in information theory (see Sec. 7).

5.1. *Scalar-quantizer index modulation*

The basic idea of QIM can be explained by looking at the simple problem of embedding one bit in a real-valued sample. The use of scalar quantizers is of course a special case of QIM which is sometimes referred to as "scalar Costa scheme" [17]. Here we have $m \in \{0, 1\}$ (1-bit message), $s \in \mathbb{R}$ (1 sample), and no key k. A scalar, uniform quantizer $Q(s)$ with step size Δ is used to generate two dithered quantizers:[a]

$$Q_i(s) = Q(s - d_i) + d_i, \quad i = 0, 1 \tag{13}$$

where

$$d_0 = -\frac{\Delta}{4}, \qquad d_1 = \frac{\Delta}{4}. \tag{14}$$

The reproduction levels of quantizers Q_0 and Q_1 are shown as circles and crosses on the real line in Fig. 3. They form two lattices:

$$\Lambda_0 = -\frac{\Delta}{4} + \Delta\mathbb{Z}, \qquad \Lambda_1 = \frac{\Delta}{4} + \Delta\mathbb{Z}. \tag{15}$$

5.1.1. *Original QIM*

This is Chen and Wornell's original idea [3]. The marked sample is defined as

$$x = \begin{cases} Q_0(s) \ : \ m = 0 \\ Q_1(s) : \ m = 1. \end{cases} \tag{16}$$

[a] Dithering is classical technique used in signal compression for improving the perceptual aspect of quantized signals.

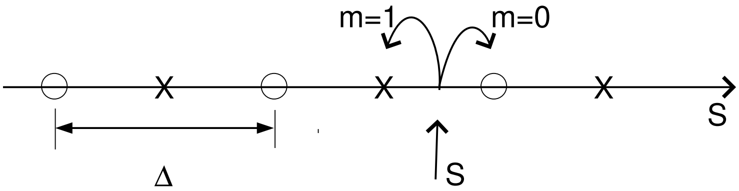


Fig. 3. Embedding one bit into one sample using original QIM. Here Λ_0 and Λ_1 are the sets of circles and crosses, respectively.

The maximum error due to embedding is $\frac{\Delta}{2}$. If the quantization errors are uniformly distributed over $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$ (in a sense made more precise in Sec. 6), the mean-squared distortion due to embedding is

$$D_1 = \frac{\Delta^2}{12}. \tag{17}$$

Assume the marked sample x is corrupted by the attacker, resulting in a noisy sample $y = x + w$. The QIM decoder is a *minimum-distance decoder.* It finds the lattice point closest to y and outputs the estimated message

$$\hat{m} = \operatorname*{argmin}_{m \in \{0,1\}} \operatorname{dist}(y, \Lambda_m) \tag{18}$$

where $\operatorname{dist}(y, \Lambda) \triangleq \min_{s \in \Lambda} |y - s|$. Clearly this scheme works perfectly (no decoding error) if $|w| < \Delta/4$. Observe that QIM may be thought of as a binning scheme with some error protection against noise (unlike the examples in Sec. 4). The two bins are the lattices Λ_0 and Λ_1, which have infinite size.

5.1.2. *Distortion-compensated scalar QIM*

The basic QIM embedding scheme (16) works poorly if the noise level exceeds $\Delta/4$. However, the scheme can be modified to increase resistance to noise, as described in [4,25]. The distortion-compensated scalar QIM embedding function is defined as

$$x = \begin{cases} Q_0(\alpha s) + (1-\alpha)s : & m = 0 \\ Q_1(\alpha s) + (1-\alpha)s : & m = 1 \end{cases} \tag{19}$$

(see Fig. 4), where $\alpha \in (0, 1]$ is a parameter to be optimized. Observe that (19) coincides with the original scheme (16) for $\alpha = 1$. The embedding formula (19) may also be written as the sum of s and a term due to quantization of αs:

$$x = \begin{cases} s + (Q_0(\alpha s) - \alpha s) : & m = 0 \\ s + (Q_1(\alpha s) - \alpha s) : & m = 1. \end{cases} \tag{20}$$

A third expression for the embedding function is

$$x = \begin{cases} \frac{d_0}{\alpha} + X_{sym}(s - \frac{d_0}{\alpha}) : & m = 0 \\ \frac{d_1}{\alpha} + X_{sym}(s - \frac{d_1}{\alpha}) : & m = 1 \end{cases} \tag{21}$$

where

$$X_{sym}(s) = Q(\alpha s) + (1 - \alpha)s \tag{22}$$

is the prototype sloped-staircase function shown in Fig. 4. This function is symmetric around $s = 0$, is made of linear segments with slope $1 - \alpha$, and takes its values in the union of intervals,

$$\mathcal{X}_{sym} := \frac{\Delta}{2\alpha} \cup_{n \in \mathbb{Z}} [n - (1 - \alpha), n + (1 - \alpha)].$$

The actual marked value x takes its values in the offset domain $\frac{d_m}{\alpha} + \mathcal{X}_{sym}$. The maximal quantization error is $|x - s| = \frac{\Delta}{2}$ and occurs when $x = \frac{\Delta}{\alpha}(\frac{1}{2} + n)$, $n \in \mathbb{Z}$. The decoder implements

$$\hat{m} = \operatorname*{argmin}_{m \in \{0,1\}} \operatorname{dist}(\alpha y, \Lambda_m), \tag{23}$$

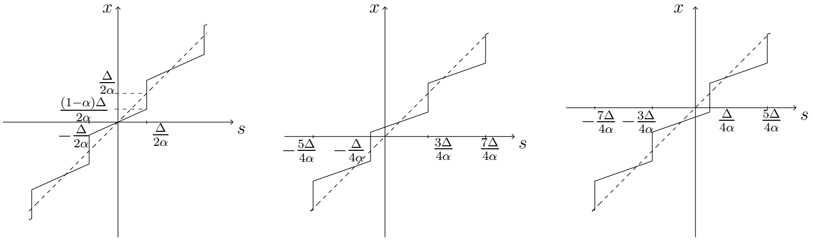
which differs from (18) due to the scaling of the received y by α.

The advantages of this generalized scheme are not obvious now but will become clear in Sec. 6 when a statistical model for the attack noise w is considered. So compelling are these advantages, in fact, that the distortion-compensated QIM scheme (19) has essentially replaced the original QIM scheme (16) in practice, and the qualifier "distortion-compensated" is often omitted for the sake of brevity.

5.2. *Sparse QIM*

Chen and Wornell showed how to extend the scalar QIM scheme above to embed one bit in a length-N host sequence [4]. They considered two basic methods.

The first method, which they called Spread Transform Dither Modulation (STDM), consists of quantizing the projection of the host vector in a



(a) Proto. $X_{sym}(s)$ (b) $m = 0$ (c) $m = 1$

Fig. 4. Selection of marked sample x given s and $m \in \{0, 1\}$, using distortion-compensated QIM.

given direction p. Specifically, given a host vector s and a unit vector p, they define the marked signal as

$$x = \begin{cases} s + (Q_0(s^T p) - s^T p)\, p : & m = 0 \\ s + (Q_1(s^T p) - s^T p)\, p : & m = 1 \end{cases} \tag{24}$$

where the superscript T denotes vector transpose. See Fig. 5. The decoder projects the received data onto direction p and decides whether quantizer Q_0 or Q_1 was used:

$$\hat{m} = \underset{m \in \{0,1\}}{\operatorname{argmin}} \operatorname{dist}(y^T p, \Lambda_m). \tag{25}$$

Observe that the distortion due to embedding takes place in direction p only; no other component of s is modified. Therefore the embedder can allocate the entire distortion budget in direction p, enabling the use of a large quantizer step size. Choosing $\Delta = \sqrt{12ND_1}$ results in an expected per-sample mean-square error equal to D_1. The large quantizer step size (relative to the case $N = 1$) offers an increased protection against noise. The distance between the lattices Λ_0 and Λ_1 is $d_{\min} = \frac{\Delta}{2} = \sqrt{3ND_1}$ and is thus proportional to $\sqrt{ND_1}$.

Various extensions and refinements of the basic STDM method are possible. In particular, one can use distortion-compensated STDM (as will be seen later, the optimal choice for α is close to 1 in that case, i.e., the scheme is very similar to basic STDM). Another idea is to quantize a few components of the host signal and not just one. All these codes are *sparse QIM codes*. The number of signal components used for embedding, divided by n, is the *sparsity factor* τ of the code; n/τ is sometimes called spreading factor.

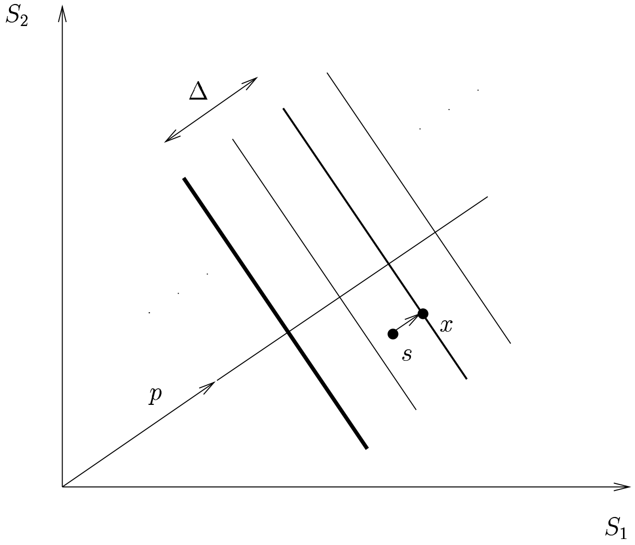


Fig. 5. STDM for embedding one bit in $N = 2$ samples.

5.3. *Lattice-quantizer index modulation*

Chen and Wornell [4] presented a second application of scalar QIM to the vector case. The idea is to replace the scalar quantizer of (19) with a L-dimensional vector quantizer. Fig. 6 illustrates this concept when $L = 2$ and the vector quantizer is obtained by independently quantizing each coordinate of αs^L with the scalar quantizer of (19). In effect αs^L is quantized using one of the two lattices

$$\Lambda_0 = \left(-\frac{\Delta}{4}, \cdots, -\frac{\Delta}{4}\right) + \Delta\mathbb{Z}^L, \qquad \Lambda_1 = \left(\frac{\Delta}{4}, \cdots, \frac{\Delta}{4}\right) + \Delta\mathbb{Z}^L. \tag{26}$$

Observe that the mean-squared distortion due to embedding is still $D_1 = \frac{\Delta^2}{12}$. The rate of the code (number of bits embedded per sample) is $R = 1/L$. The distance between the sets Λ_0 and Λ_1 is now

$$d_{\min} = \frac{1}{2}\Delta\sqrt{L} = \sqrt{3LD_1}.$$

The decoder's output is

$$\hat{m} = \operatorname*{argmin}_{m\in\{0,1\}} \operatorname{dist}(\alpha y^L, \Lambda_m), \tag{27}$$

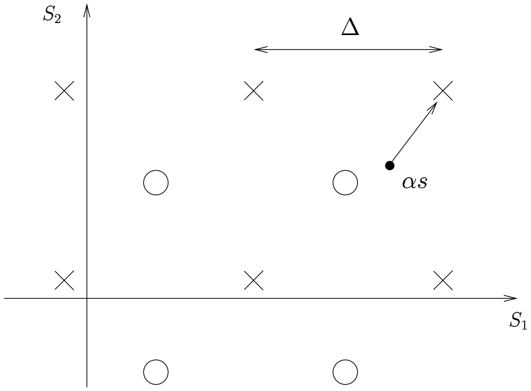


Fig. 6. QIM for embedding one bit in $L = 2$ samples using cubic lattice.

defining $\mathrm{dist}(y^L, \Lambda) := \min_{p^L \in \Lambda} \|y^L - p^L\|$. The quantity $\mathrm{dist}(\alpha y^L, \Lambda_m)$ is a coordinatewise sum of squared quantization errors.

5.3.1. *General construction*

The papers [23,48] presented a general approach for constructing *good* structured binning schemes. The approach is based on nested linear codes. A nested linear code is a N-dimensional *lattice partition* Λ/Λ' where Λ and Λ' are respectively referred to as the *fine* lattice and the *coarse* lattice. Define

Q = quantization function mapping each point $x \in \mathbb{R}^N$ to the nearest lattice point in Λ'

$\mathcal{V}$ = $\{x \in \mathbb{R}^N \ : \ Q(x) = 0\}$ = Voronoi cell of Λ'

$\mathcal{C}$ = quotient Λ/Λ'.

Example: let $\Lambda' = \Delta\mathbb{Z}^N$ and $\Lambda = D_N^+ = \Delta\mathbb{Z}^N \cup (\frac{\Delta}{2}, \cdots, \frac{\Delta}{2}) + \Delta\mathbb{Z}^N$, which is a lattice for all even N. We obtain $\mathcal{C} = \{(0,0), (\frac{\Delta}{2}, \frac{\Delta}{2})\}$. Then $\mathcal{V}$ is the N-dimensional cube $[-\frac{\Delta}{2}, \frac{\Delta}{2}]^N$; its normalized second-order moment is equal to $\frac{\Delta^2}{12}$. Fig. 6 illustrates this design (shifted by $(\frac{\Delta}{4}, \frac{\Delta}{4})$) when $N = 2$.

The lattice partition Λ/Λ' should have the following properties.

(P1) Q should be a *good* vector quantizer with mean-squared distortion D_1: loosely speaking, $\mathcal{V}$ should be nearly spherical.

(P2) $\mathcal{C}$ should be a good channel code with respect to Gaussian noise: loosely speaking, the codewords in $\mathcal{C}$ should be far away from each other.

To each $m \in \mathcal{M}$ corresponds a codeword $c_m \in \mathcal{C}$ and a translated coarse lattice $\Lambda_m = c_m + \Lambda$. The fine lattice is the union of all these translated lattices.

Given m and s^N, the encoder quantizes αs^N to the nearest point in Λ_m, obtaining

$$u^N(m) = Q_m(\alpha s^N) := Q(\alpha s^N - c_m) + c_m \quad \in \Lambda_m.$$

The difference $u^N(m) - \alpha s^N$ represents a quantization error. Finally, the marked sequence is given by

$$\begin{aligned} x^N &= (1-\alpha)s^N + u^N(m) && (28) \\ &= (1-\alpha)s^N + Q_m(\alpha s^N) && (29) \end{aligned}$$

which is a generalization of (19).

The decoder quantizes αy^N to the nearest point in the fine lattice $\Lambda' = \cup_{m \in \mathcal{M}} \Lambda_m$. It then outputs the corresponding index $\hat{m}$ according to (27).

5.3.2. *Practical codes*

To satisfy properties (P1) and (P2) above, we need Λ and Λ' to be high-dimensional. In practice, one cannot afford using arbitrary high-dimensional lattices, because quantization operations become prohibitively expensive. Instead one can would use lattices that have a special structure, e.g., products of low-dimensional lattices.[b] Another powerful idea is to use recursive quantization techniques such as trellis-coded quantization [7,22] to (implicitly) define the coarse lattice Λ. Similarly, one can use classical error-correction codes such as Hamming codes and turbo codes to (implicitly) define the fine lattice Λ'. The latter idea is illustrated in Fig. 7, where the actual message $m \in \mathcal{M}$ is first encoded into a longer (redundant) sequence $\tilde{m}$, which is used as an input to the nested lattice code. These two codes are termed outer code and inner code, respectively. Chou and Ramchandran [8] recently proposed the use of an outer erasure code; their scheme is intended to resist erasures, insertions and deletions, in addition to the Gaussian-type attacks that the inner code is designed to survive. Solanki *et al* [41] studied a closely related system, see Sec. 10 for more details.

[b] The cubic lattice is the simplest example of a product lattice.

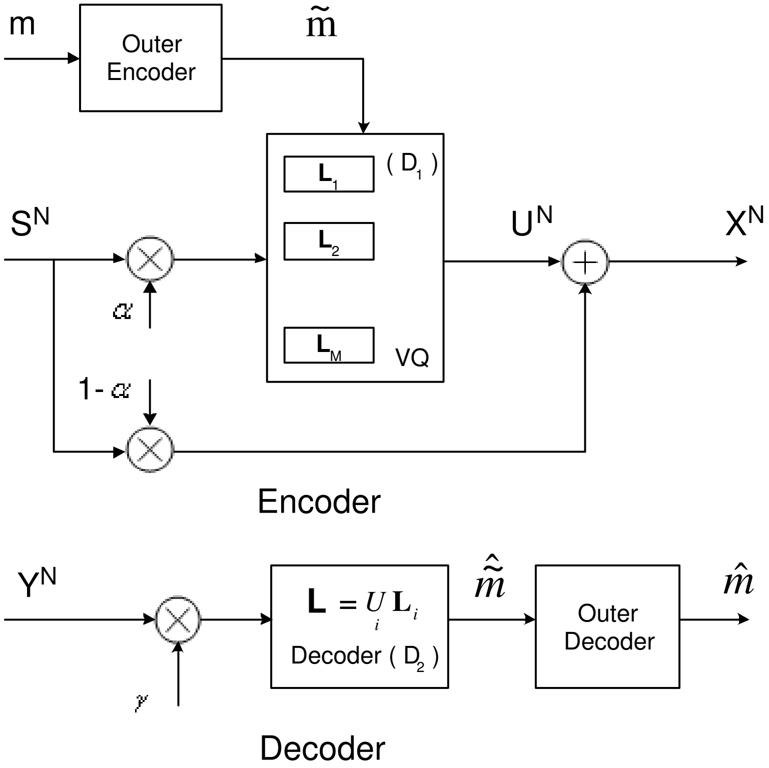


Fig. 7. Lattice-based encoder and decoder for data hiding, using the encoding function (28) and the decoding function (27).

6. Probability of Error

The natural metric for quantifying decoding performance is probability of decoding error. This type of analysis can be rather complicated but useful results can be obtained using appropriate asymptotic methods ($N \to \infty$).

For simplicity of the exposition assume that the only data available to the decoder is the degraded signal Y^N (i.e., no side information K^N). The decoding rule partitions the received data space into decoding regions $\mathcal{Y}_m, m \in \mathcal{M}$. The decoder outputs message m for all sequences that belong to $\mathcal{Y}_m$. The probability that message m is not decoded correctly is $P_{e|m} = Pr[Y^N \notin \mathcal{Y}_m \mid m \text{ sent}]$. It depends on $\{\mathcal{Y}_m\}$ and the statistics of the host signal and the randomized code. To analyze this problem, it is convenient to study the case of two codewords first.

6.1. *Binary detection*

The message set in the binary detection case is $\mathcal{M} = \{0, 1\}$. The decoding problem is a binary hypothesis testing problem:

$$\begin{cases} H_0 & : \; Y^N \sim p_0 \\ H_1 & : \; Y^N \sim p_1 \end{cases} \tag{30}$$

where the notation $Y^N \sim p$ means that Y^N is a random vector with probability distribution $p(y^N)$. Some detection rules are relatively simple. Such are the correlation and nearest-neighbor decoding rules encountered in SSM and QIM watermarking. A statistical model such as (30) is not required in this case.

Improved detection rules can often be derived by exploiting knowledge of the statistics of Y^N. For instance, if both messages are equally likely, the detector that minimizes probability of error is the maximum likelihood (ML) detector [40]:

$$L(y^N) = \frac{p_1(y^N)}{p_0(y^N)} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \tag{31}$$

where $L(y^N)$ is the likelihood ratio test statistic.

The probability of error for the test (31) is[c]

$$P_e = \frac{1}{2} \int \min(p_0(y^N), p_1(y^N)) \, dy^N. \tag{32}$$

Fig. 8 depicts the distribution of $L(y^N)$ under hypotheses H_0 and H_1. Two types of error are shown in the figure: deciding H_1 when H_0 is true (type I), and deciding H_0 when H_1 is true (type II). P_e is the average of these two error probabilities.

To achieve low P_e, we need to create a substantial disparity between the probability density functions (p.d.f.'s) p_0 and p_1. Let us see how some basic data-hiding codes perform in this respect. We use a simple model to illustrate the ideas: embed 1 bit into 1 sample.

Example: Consider real-valued s, x and y. The attack is $y = x + w$, where W is Gaussian noise, distributed as $\mathcal{N}(0, \sigma_w^2)$. The host signal sample S is distributed as $\mathcal{N}(0, \sigma_s^2)$. Define the *Watermark to Noise Ratio* $WNR := \frac{a^2}{\sigma_w^2}$ and the *Watermark to Host Ratio* $WHR := \frac{a^2}{\sigma_s^2}$. The performance of SSM and QIM systems is derived below.

[c] The integral is a sum if $\mathcal{Y}$ is a discrete set.

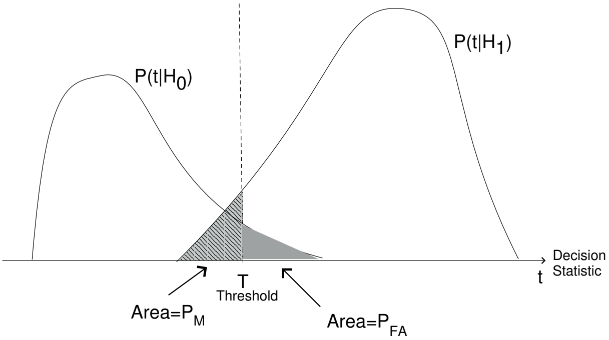


Fig. 8. Testing between two statistical hypotheses.

6.1.1. *Spread-spectrum modulation*

The spread-spectrum scheme is given by

$$x = \begin{cases} s + a : & m = 0 \\ s - a : & m = 1, \end{cases} \tag{33}$$

and the original s is unknown to the detector. Equation (33) is a special case of (7). Then $p_0 = \mathcal{N}(a, \sigma_s^2 + \sigma_w^2)$ and $p_1 = \mathcal{N}(-a, \sigma_s^2 + \sigma_w^2)$. Both p.d.f.'s are shown in Fig. 9. They are hard to distinguish when $a^2 << \sigma_s^2 + \sigma_w^2$. This corresponds to the common case of a strong host-to-watermark ratio; detection performance is poor. More precisely, $P_e = Q(d/2)$, where

$$d = \sqrt{\frac{(2a)^2}{\sigma_s^2 + \sigma_w^2}} = 2\sqrt{(WNR^{-1} + WHR^{-1})^{-1}}$$

is a normalized distance between the two p.d.f.'s, and $Q(t) = \int_t^\infty (2\pi)^{-1/2} e^{-x^2/2}\, dx$ is the Q function. Observe that $P_e \to \frac{1}{2}$ as $WHR \to 0$, i.e. detection becomes completely unreliable.

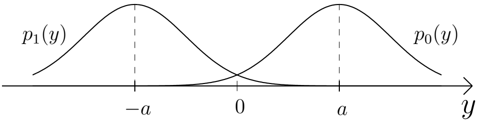


Fig. 9. Rival p.d.f.'s for detection of $m \in \{0, 1\}$, using a simple SSM code.

For detection with the host signal known to the detector (private watermarking), we have $P_e = Q(d/2)$ again, where $d = \sqrt{\frac{(2a)^2}{\sigma_w^2}} = 2\sqrt{WNR}$. This performance is achieved independently of the value of WHR, and so detection performance is much improved when $WHR << WNR$. In fact SSM is an ideal modulation scheme for this problem.

6.1.2. *Scalar QIM*

Assume again that S is unknown to the detector. Consider the distortion-compensated scalar QIM scheme (20). The rival distributions of Y are shown in Fig. 10 for $\sigma_w^2 = 0$ and for $\sigma_w^2 = 1$. The p.d.f.'s in the second case are the same as the p.d.f.'s in the case $\sigma_w^2 = 0$, convolved with the Gaussian noise p.d.f., $\mathcal{N}(0, \sigma_w^2)$. Observe that

(1) The perturbation due to embedding (quantization noise) is limited between $-\frac{\Delta}{2}$ and $\frac{\Delta}{2}$. Under Bennett's high-rate model for quantization noise, this perturbation is uniformly distributed between $-\frac{\Delta}{2}$ and $\frac{\Delta}{2}$, and the distortion due to embedding is $D_1 := \mathbb{E}(X-S)^2 = \frac{\Delta^2}{12}$.[d] Equivalently, given D_1, we select

$$\Delta = \sqrt{12D_1}. \tag{34}$$

Also $WNR = \frac{\Delta^2/12}{\sigma_w^2}$.

(2) For large σ_s^2, we can view the p.d.f.'s as quasi-periodic, with period equal to $\frac{\Delta}{\alpha}$. Roughly speaking, the ability to discriminate between p_0 and p_1 depends on the overlap between the support sets of p_0 and p_1, and fairly little on σ_s^2.

(3) The rounded "pulses" that make up the p.d.f.'s p_0 and p_1 are given by the convolution of a rectangular pulse of width $(1-\alpha)\Delta/\alpha$, with the $\mathcal{N}(0, \sigma_w^2)$ p.d.f..

(4) For good discrimination between p_0 and p_1, the pulses should have relatively small overlap.

(5) In the absence of attacker's noise ($\sigma_w^2 = 0$), the best choice for α would be 1, in which case we obtain error-free detection.

(6) For $\sigma_w^2 > 0$, the choice of α is a tradeoff between embedding distortion and detection performance. The tradeoff is determined by the value of the parameters Δ and α of the embedding function (19).

[d] The uniform quantization model is *exact* for any value of Δ if a uniformly distributed external dither is applied [48]. For the problem at hand, this means that d_0 is randomized uniformly over $[0, \Delta]$ and that we keep $|d_1 - d_0| = \frac{\Delta}{2}$.

(7) For large σ_s^2, little information is lost by reducing Y to the test statistic

$$\tilde{Y} := \alpha Y \bmod \Delta. \tag{35}$$

This nonlinear transformation "folds" the p.d.f.'s into an interval of width Δ. The p.d.f.'s of $\tilde{Y}$ under H_0 and H_1 are shown in Fig. 11 for two values of α. The minimum-distance decoding rule (23) is equivalent to

$$\hat{m} = \operatorname*{argmin}_{m \in \{0,1\}} |\tilde{y} - d_m| \tag{36}$$

where $|d_1 - d_0| = \frac{\Delta}{2}$.

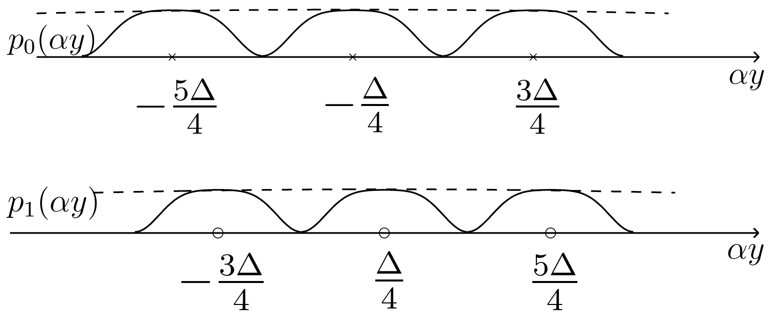


Fig. 10. Rival p.d.f.'s for detection of $m \in \{0, 1\}$ based on unprocessed data Y, using scalar QIM with $WNR = 0.1$ and $\alpha = \frac{WNR}{1+WNR}$.

6.2. *Modulo additive noise channel*

The advantage of the processing (35) of the data Y is that it lends itself to a detection test that is simple, good, and independent of the exact statistics of S. From (8), (21), (35), note that

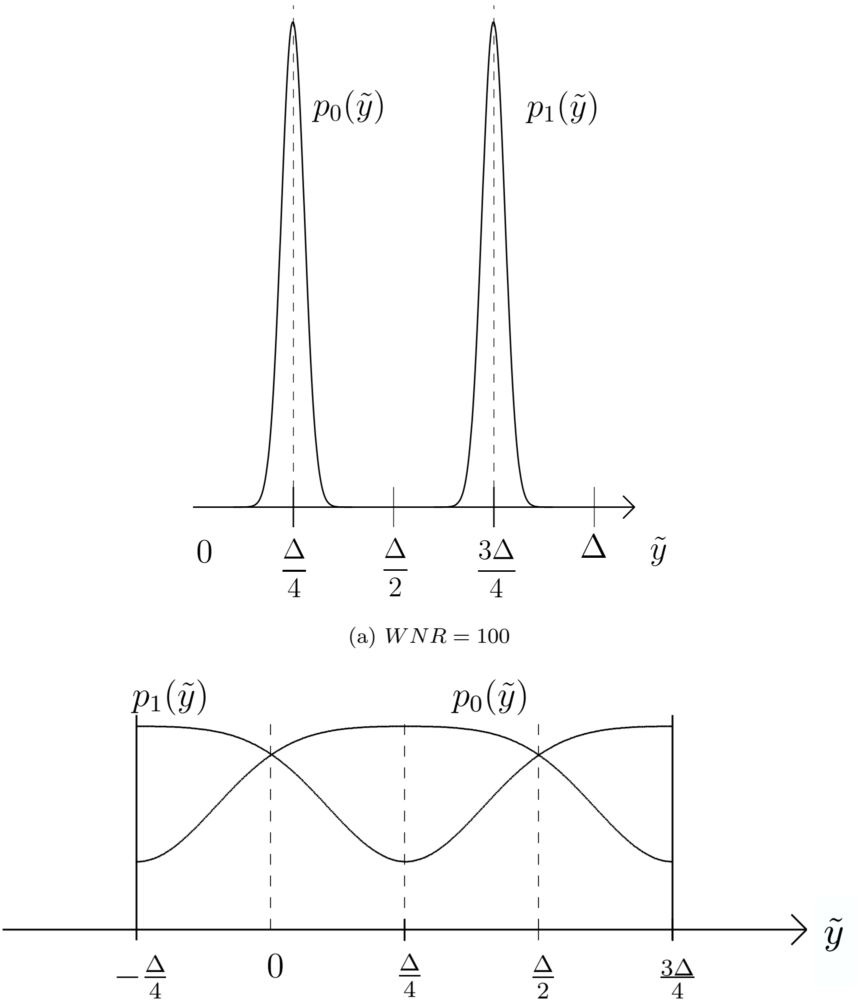
$$\tilde{y} = (d_m + \tilde{e} + \tilde{w}) \bmod \Delta \tag{37}$$

where

$$\tilde{E} := \alpha X_{sym} \left(S - \frac{d_m}{\alpha} \right) \bmod \Delta \tag{38}$$

is termed *self-noise*, and

$$\tilde{W} := \alpha W \bmod \Delta \tag{39}$$



(a) $WNR = 100$

(b) $WNR = 0.1$

Fig. 11. Rival p.d.f.'s for detection of $m \in \{0, 1\}$ based on transformed $\tilde{Y}$, using scalar QIM with $\alpha = \frac{WNR}{1+WNR}$.

is the *aliased attacker's noise*. Indeed the p.d.f. of $\tilde{W}$ is an aliased version of $p_{\alpha W}$:

$$p_{\tilde{W}}(\tilde{w}) = \sum_{k=-\infty}^{\infty} p_{\alpha W}(\alpha \tilde{w} + k\Delta), \quad 0 \le \tilde{w} \le \Delta. \tag{40}$$

Note that $\tilde{E} = 0$ for $(1-\alpha)\frac{\Delta}{2} < |\tilde{E}| < (1+\alpha)\frac{\Delta}{2}$. Under the uniform quantization noise model, $p_{\tilde{E}}$ is a rectangular pulse of width $(1-\alpha)\Delta$ centered at 0:

$$p_{\tilde{E}}(\tilde{e}) = \frac{1}{\Delta(1-\alpha)} 1_{\{|\tilde{e}| \leq \frac{\Delta}{2}(1-\alpha)\}}.$$

Under hypothesis H_i, $i = 0, 1$, the data $\tilde{Y}$ is the sum of an information-bearing offset d_i and a noise V equal to the sum of the self-noise and the aliased attacker's noise:

$$V = \tilde{E} + \tilde{W} \bmod \Delta. \tag{41}$$

Since $\tilde{E}$ and $\tilde{W}$ are statistically independent, the p.d.f. of V is the circular convolution of the p.d.f.'s of $\tilde{E}$ and $\tilde{W}$:

$$\begin{aligned} p_V(v) &= (p_{\tilde{E}} \star p_{\tilde{W}})(v) \\ ] &= \int p_{\tilde{E}}(\tilde{e}) p_{\tilde{W}}(v - \tilde{e})\, d\tilde{e}, \quad 0 \leq v \leq \Delta. \end{aligned} \tag{42}$$

Therefore the p.d.f. of $\tilde{Y}$ under H_i takes the form

$$q_i(\tilde{y}) = p_V(\tilde{y} - d_i), \quad i = 0, 1 \tag{43}$$

The rival p.d.f.'s $q_i(\tilde{y}), i = 0, 1$ are simply translates of p_V. The detector must decide between the two hypotheses

$$\begin{cases} H_0 \ : \ \tilde{Y} = d_0 + V \\ H_1 \ : \ \tilde{Y} = d_1 + V \end{cases} \tag{44}$$

The role of α as a tradeoff between self-noise and attacker's noise appears clearly in this formulation of the detection problem. For small α, the self-noise $\tilde{E}$ dominates the attacker's aliased noise $\tilde{W}$. For $\alpha = 1$, the self-noise is zero, and the attacker's noise dominates. Equation (44) defines a *Modulo Additive Noise* (MAN) channel, diagrammed in Fig. 12.

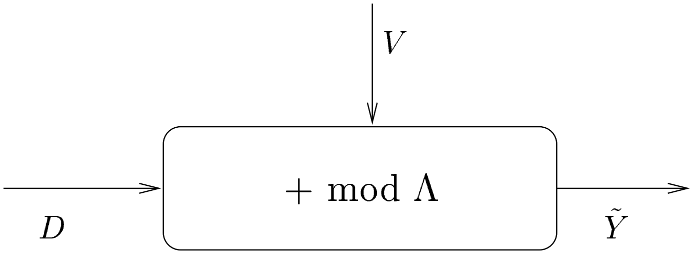


Fig. 12. Modulo Additive Noise Channel.

As an alternative to the simple minimum-distance detector (36), we study the theoretically optimal ML detector. The ML detector based on the transformed data $\tilde{Y}$ and the statistical model above is

$$\frac{p_V(\tilde{y}-d_1)}{p_V(\tilde{y}-d_0)} \underset{H_0}{\overset{H_1}{\gtrless}} 1. \tag{45}$$

It coincides with the minimum-distance detection rule (23) if the attacker's noise distribution p_W is unimodal and symmetric.

The probability of error for the optimal test (45) is

$$\tilde{P}_e = \frac{1}{2}\int \min(q_0(\tilde{y}), q_1(\tilde{y}))\, d\tilde{y}. \tag{46}$$

If the noise distribution $p_W(w)$ is symmetric around $w=0$, so is $p_{\tilde{W}}(\tilde{w})$. The two rival p.d.f.'s, $q_0(\tilde{y})$ and $q_1(\tilde{y})$, have means $\mu_0 = d_0$ and $\mu_1 = d_1$ respectively, and common variance σ_v^2. For moderate-to-large WNR, we have

$$\sigma_v^2 \approx (1-\alpha)^2\frac{\Delta^2}{12} + \alpha^2\sigma_{\tilde{w}}^2.$$

So the "generalized SNR" for detection is given by

$$\begin{aligned} GSNR &:= \frac{(\mu_1-\mu_0)^2}{\sigma_v^2} \\ &\approx \frac{(d_1-d_0)^2}{\frac{1}{12}(1-\alpha)^2\Delta^2 + \alpha^2\sigma_{\tilde{w}}^2} \end{aligned} \tag{47}$$

where $|d_1 - d_0| = \frac{\Delta}{2}$. The value of α that maximizes $GSNR$ is given by a nonlinear equation. (Note that $\sigma_{\tilde{w}}^2$ is a decreasing function of Δ and tends to σ_w^2 if $\Delta >> \alpha\sigma_w$.) A reasonable approximation for α that maximizes GSNR is

$$\alpha_{\max - GSNR} \approx \frac{\Delta^2/12}{\Delta^2/12 + \sigma_w^2} = \frac{WNR}{WNR+1} \tag{48}$$

whence $\max_\alpha GSNR \approx 3(WNR+1)$. The actual maximizing α is slightly lower than the right side of (48) because $\sigma_{\tilde{w}}^2 \geq \sigma_w^2$.

While GSNR is often useful as a rough measure of separation of the p.d.f.'s q_0 and q_1, it does not necessarily serve as an accurate predictor of detection performance. Fig. 13 plots $GSNR$ and $\tilde{P}_e$ as a function of α, for three different values of WNR. Note that the optimal α is slightly different under the $GSNR$ and $\tilde{P}_e$ criteria.

Quite interesting is the performance gap relative to private watermarking, which bounds the performance of any public watermarking scheme. In this case the spread-spectrum scheme (33) yields an error probability $P_e = Q(\sqrt{WNR})$ which is typically smaller than the QIM error probabilities by a factor of 2–3 when WNR ranges from 0.2 to 5, see Fig. 13. The performance loss is quite small, considering that the QIM detector does not known the host signal.

6.3. *Probability of error – Vector case*

The previous two subsections have quantified the benefits of QIM in terms of probability of error for binary detection based on a single observation. This subsection considers the more realistic case of N observations and studies two approximations to the probability of error.

Assume we have a host data vector $S^N = \{S_1, S_2, \cdots, S_N\}$ and we mark each component S_i using the spread-spectrum and QIM techniques. Moreover,

(a) S^N is Gaussian with mean zero and covariance matrix R_{S^N};

(b) the marked signal X^N is corrupted by additive white Gaussian noise W^N with mean zero and variance σ_w^2.

6.3.1. *Spread spectrum modulation*

For the spread-spectrum scheme, (33) generalizes to

$$x^N = \begin{cases} s^N + a^N : & m = 0 \\ s^N - a^N : & m = 1 \end{cases} \tag{49}$$

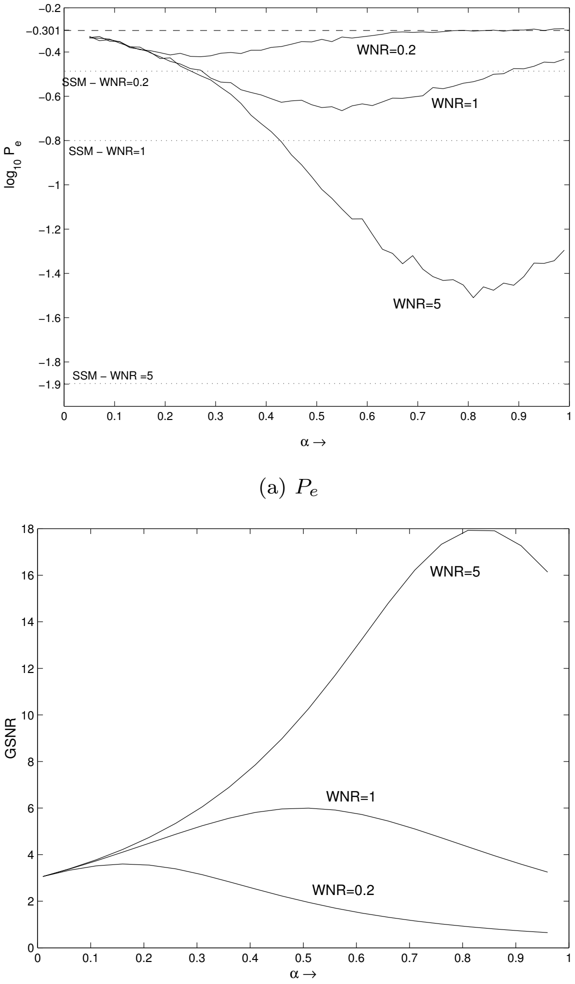
where the spread sequence a^N is known to the detector. For blind watermarking we have

$$p_0 = \mathcal{N}(a^N, R_{S^N} + \sigma_w^2 I_N), \quad p_1 = \mathcal{N}(-a^N, R_{s^N} + \sigma_w^2 I_N).$$

The LRT takes the form

$$a^{NT}(R_{S^N} + \sigma_w^2 I_N)^{-1} y^N - \frac{1}{2} a^{NT}(R_{S^N} + \sigma_w^2 I_N)^{-1} a^N \underset{H_0}{\overset{H_1}{\gtrless}} 0 \tag{50}$$

and the probability of error of the test (50) is $P_e = Q(d/2)$, where $d^2 = 4a^{NT}(R_{S^N} + \sigma_w^2 I_N)^{-1} a^N$ is the GSNR for the detector.



(a) P_e

(b) $GSNR$

Fig. 13. Generalized SNR and probability of error P_e for binary detection based on one single sample. The variable on the horizontal axis is the tradeoff parameter α for QIM. For comparison, P_e for the private and public SSM schemes is given by the ordinate of the dotted horizontal lines.

For private watermarking we have

$$p_0 = \mathcal{N}(a^N, \sigma_w^2 I_N), \quad p_1 = \mathcal{N}(-a^N, \sigma_w^2 I_N).$$

Then $P_e = Q(d/2)$ where $d^2 = \frac{\|2a^N\|^2}{\sigma_w^2} = 4\,WNR$.

6.3.2. *Scalar QIM*

For the scalar QIM scheme, let $d_i^N = \{d_{i,1}, d_{i,2}, \cdots, d_{i,N}\}$ for $i = 0, 1$. We assume again that $d_{i,n} \in \{\pm\frac{\Delta}{4}\}$ and that the noise p.d.f. $p_W(w)$ is symmetric around $w = 0$. Equation (19) generalizes to

$$x^N = \begin{cases} Q_0(\alpha s^N) + (1-\alpha)s^N : & m = 0 \\ Q_1(\alpha s^N) + (1-\alpha)s^N : & m = 1 \end{cases} \tag{51}$$

where each Q_i is viewed as a vector quantizer, in this case simply a product of scalar quantizers:

$$(Q_i(s^N))_n = Q(s_n - d_{i,n}) + d_{i,n}, \quad 1 \le n \le N,\, i = 0, 1.$$

Without loss of generality, we shall assume $d_{0,n} \equiv \frac{\Delta}{4}$ and $d_{1,n} \equiv \frac{3\Delta}{4}$.

The first step at the receiver is to compute the transformed data

$$\tilde{Y}_n = \alpha Y_n \bmod \Delta, \quad 1 \le n \le N. \tag{52}$$

Under the uniform quantization noise model, the preprocessed data $\{\tilde{Y}_n, 1 \le n \le N\}$ are mutually independent, even though there may be dependencies between the host signal samples $\{S_n\}$. The detector must decide between the two hypotheses

$$\begin{cases} H_0 & : \tilde{Y}^N = d_0^N + V^N \\ H_1 & : \tilde{Y}^N = d_1^N + V^N \end{cases} \tag{53}$$

where the samples $V_n, 1 \le n \le N$, are independent and identically distributed (i.i.d.) with p.d.f. p_V given in (42). The ML detector based on $\tilde{Y}^N$ and the statistical model above is

$$\tilde{L}(\tilde{y}^N) = \prod_{n=1}^{N} \frac{p_V(\tilde{y}_n - d_{1,n})}{p_V(\tilde{y}_n - d_{0,n})} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \tag{54}$$

which coincides with the minimum-distance detector (27) in some cases. Similarly to (36), the minimum-distance detection rule may be written in the form

$$\hat{m} = \underset{m \in \{0,1\}}{\operatorname{argmin}} \sum_{n=1}^{N} |\tilde{y}_n - d_{m,n}|^2. \tag{55}$$

The probability of error is given by

$$\tilde{P}_e = \frac{1}{2} \int_{[0,\Delta]^N} \min(q_0^N(\tilde{y}^N), q_1^N(\tilde{y}^N))\, d\tilde{y}^N. \tag{56}$$

It may in principle be computed numerically, using integration over the N-dimensional cube $[0, \Delta]^N$. Unfortunately such methods are impractical even for relatively small N. Monte-Carlo simulations are an alternative, but are time-consuming and do not necessarily provide analytical insights. Two analytic methods for approximating $\tilde{P}_e$ are considered next.

6.3.3. *Gaussian approximation*

One may easily derive the generalized SNR at the detector, as was done in Sec. 6.2. Formula (47) generalizes to

$$\begin{aligned} GSNR &\approx \frac{\|d_1 - d_0\|^2}{\frac{1}{12}(1-\alpha)^2\Delta^2 + \alpha^2\sigma_{\tilde{w}}^2} \\ &= \frac{N\Delta^2/4}{\frac{1}{12}(1-\alpha)^2\Delta^2 + \alpha^2\sigma_{\tilde{w}}^2}. \end{aligned} \tag{57}$$

If the noise V was Gaussian, the probability of error would be given by

$$\tilde{P}_e = Q(\sqrt{GSNR}/2). \tag{58}$$

However V is non-Gaussian, and (58) is generally a poor approximation to the actual $\tilde{P}_e$.

6.3.4. *Large deviations*

If $GSNR$ is large (as is always the case for sufficiently large N), the performance of the detection test is dominated by rare events (whose frequency of occurrence is determined by the tails of the p.d.f.'s q_0^N and q_1^N), and Gaussian approximations of these tails are usually severely inaccurate. The usual approach to such problems in the detection literature is based on large deviations theory [40]. The following upper bound on $\tilde{P}_e$ holds for any N:

$$\tilde{P}_e \leq \frac{1}{2}e^{-NB(q_0,q_1)}$$

where

$$B(q_0, q_1) = -\ln \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \sqrt{q_0(\tilde{y})q_1(\tilde{y})}\, d\tilde{y} \tag{59}$$

is the so-called Bhattacharyya coefficient, or Bhattacharyya distance between the p.d.f.'s q_0 and q_1. Moreover, the bound is tight in the exponent:[e]

$$\lim_{N\to\infty}[-\frac{1}{N}\ln\tilde{P}_e] = B(q_0, q_1).$$

Hence $B(q_0, q_1)$ is a more useful predictor of detection performance than is $GSNR$. It is easy to compute (via numerical 1-D integration in (59)) and can be used to determine how large N should be to guarantee a prescribed probability of error.

The Bhattacharyya coefficient $B(q_0, q_1)$ depends on the QIM parameter α via q_0 and q_1. The dependency of $B(q_0, q_1)$ on α is shown in Fig. 14. The approach above can be generalized to lattice QIM [35].

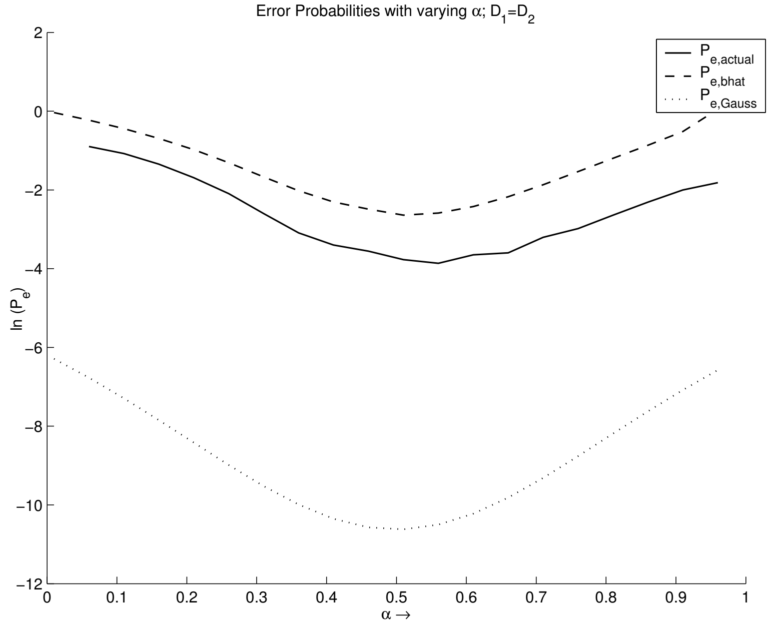


Fig. 14. $\tilde{P}_e$ and its upper bound based on the Bhattacharyya coefficient $B(q_0, q_1)$ for binary detection based on $N = 15$ samples. Also shown is the Gaussian approximation to $\tilde{P}_e$, which is overoptimistic by several orders of magnitude. The variable on the horizontal axis is the QIM tradeoff parameter α.

[e] In general, a Chernoff bound with optimal Chernoff exponent is tight. However, due to the symmetry of p_V and the fact that q_0 and q_1 are translates of p_V, the optimal Chernoff exponent is $\frac{1}{2}$, and thus the optimal bound is the Bhattacharyya bound.

6.4. *Multiple codewords*

In the case of $|\mathcal{M}| > 2$, calculation of the probability of error

$$P_e = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} Pr[Y^N \notin \mathcal{Y}_m | m \text{ sent}]$$

presents difficulties if $\mathcal{M}$ is large. Fortunately, useful bounds on P_e can be derived.

Assume equally likely codewords. For linear codes, the conditional error probability $Pr[Y^N \notin \mathcal{Y}_m | m \text{ sent}]$ is independent of the message m that was sent. Thus we may arbitrarily select $m = 0$ and write

$$P_e = Pr[Y^N \notin \mathcal{Y}_0 | m = 0].$$

A useful upper bound on P_e can sometimes be obtained using the union bound:

$$P_e \leq (|\mathcal{M}| - 1) \max_{i \neq j \in \mathcal{M}} P_{e|i,j} \tag{60}$$

where $P_{e|i,j}$ is the probability of error for a binary test between hypotheses H_i and H_j. Such bounds are typically useful at low bit rates.

Assume for simplicity that a scalar QIM system is used, with Bhattacharyya distance $B(q_0, q_1)$ given in (59). The code $\mathcal{C} = \Lambda/\Lambda'$ is a subset of $\{-\frac{\Delta}{4}, \frac{\Delta}{4}\}^N$. Assume this code has minimum Hamming distance equal to d_H (typically proportional to N), we obtain

$$\tilde{P}_{e|i,j} \leq e^{-d_H B(q_0, q_1)}.$$

If the number of messages grows with N in a subexponential fashion ($\frac{1}{N} \log |\mathcal{M}| \to 0$ as $N \to \infty$), the total probability of error exponent is determined by the worst-case pair of hypotheses:

$$P_e \leq (|\mathcal{M}| - 1) e^{-d_H B(q_0, q_1)}.$$

It is thus desirable to use codes with good minimum-distance properties.

In the case where $|\mathcal{M}|$ grows exponentially with N, the union bound (60) may be loose; if $\log |\mathcal{M}| \geq d_H B(q_0, q_1)$, the bound becomes trivial (≥ 1). Finding better bounds in this case is a topic of current research [19,27].

7. Data-Hiding Capacity

After analyzing probability of decoding error for binning schemes, we turn our attention to a closely related problem, namely what is the maximal rate of a code that allows reliable transmission ($P_e \to 0$ as $N \to \infty$). In other words, we wish to determine a Shannon capacity for data hiding [39].

The rate of the data hiding code $(\mathcal{M}, f_N, \phi_N)$ is $R = \frac{1}{N} \log |\mathcal{M}|$, and the average probability of error is

$$P_{e,N} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} Pr[\phi_N(Y^N, K^N) \neq m \mid M = m]. \tag{61}$$

A rate R is said to be achievable for distortion D_1 and for a class of attack channels $\mathcal{A}^N, N \geq 1$, if there is a sequence of codes subject to distortion D_1, with rate R, such that $P_{e,N} \to 0$ as $N \to \infty$, for any sequence of attacks in $\mathcal{A}^N$. The *data-hiding capacity* $C(D_1, \{\mathcal{A}^N\})$ is then defined as the supremum of all achievable rates[f] for distortion D_1 and attacks in the class $\{\mathcal{A}^N\}$.

7.1. *Finite alphabets*

For simplicity of the exposition, consider the average distortion constraints (1) and (3), and assume the host signal and the attack channel are memoryless. Then

$$A^N(y^N|x^N) = \prod_{i=1}^{N} A(y_i|x_i).$$

The data-hiding capacity defined above turns out to be the solution of a certain mutual-information game and is given in the theorem below. Let $U \in \mathcal{U}$ be an auxiliary random variable such that $(U, S) \to X \to Y$ forms a Markov chain. Let $\mathcal{Q}(D_1)$ be the set of *covert channels* Q that satisfy the constraint

$$\sum_{x,s,k,u} d_1(s,x) Q(x,u|s,k) p(s,k) \leq D_1, \tag{62}$$

$\mathcal{A}(D_2)$ be the set of attack channels A that satisfy the constraint

$$\sum_{s,x,k,y} d_2(x,y) A(y|x) p(x|s,k) p(s,k) \leq D_2, \tag{63}$$

and $\mathcal{A}$ be an arbitrary subset of $\mathcal{A}(D_2)$.

[f] Often NR is termed *payload* of the code in the watermarking literature. Some authors use the word capacity as a synonym for payload. This can create confusion because R is an attribute of the particular code considered, not Shannon's (code-independent) capacity.

Theorem 1: [39] Assume the attacker knows the encoding function f_N and the decoder knows f_N and the attack channel A. A rate R is achievable for distortion D_1 and attacks in the class $\mathcal{A}$ if and only if $R < C$, where C is given by

$$C := C(D_1, \mathcal{A}) = \max_{Q(x,u|s,k) \in \mathcal{Q}(D_1)} \min_{A(y|x) \in \mathcal{A}} J(Q, A) \tag{64}$$

where $|\mathcal{U}| \leq |\mathcal{X}||\Omega| + 1$, Ω is the support set of $p(s,k)$, and

$$J(Q, A) = I(U; Y|K) - I(U; S|K) \tag{65}$$

where $I(X;Y|Z) \overset{\triangle}{=} \sum_{x,y,z} p(x,y,z) \log \frac{p(x,y|z)}{p(x|z)p(y|z)}$ denotes conditional mutual information [11].

Gel'fand–Pinsker. The capacity result (64) is closely related to an important result by Gel'fand and Pinsker [21] in 1980. They derived the capacity of a memoryless channel whose state is known to the encoder but not to the decoder. The encoder may exploit the state information using a binning technique, as discussed below. The role of the channel state is analogous to the role of the host signal in blind data hiding. Key differences with the Gel'fand–Pinsker problem include the existence of distortion constraints, the availability of K^N at both the encoder and decoder, and the fact that the attack channel is unknown to the encoder – whence the minimization over A in (64).

Binning Schemes. In principle, the capacity bound can be approached using a *random binning* coding technique [11,21], which exemplifies the role of the covert channel Q, see Fig. 15. A size–$2^{N(I(U;Y,K)-\epsilon)}$ codebook $\mathcal{C}$ is constructed for the variable U^N by randomly sampling the capacity-achieving distribution $p(u^N)$, and partitioning the samples into $|\mathcal{M}|$ equal-size subsets (lists). The actual embedding of a message $m \in \mathcal{M}$ proceeds as follows: first identify an element $u^N(m)$ from the list of elements indexed by m in the codebook $\mathcal{C}$, in such a way that $u^N(m)$ is statistically typical with the current (s^N, k^N), then generate watermarked data x^N according to the p.m.f. $p(x^N|u^N(m), s^N, k^N)$. The decoder finds $\hat{u}^N$ that is statistically typical with (y^N, k^N), and obtains $\hat{m}$ as the index of the list to which $\hat{u}^N$ belongs. However, memory and computational requirements grow exponentially with block length N, and so such approaches are known to be infeasible in practice. Developing structured binning schemes that approach the capacity bound is a active research area [4,23,48,7,5,18,6]. This problem is closely related to the problem of developing good nested lattice codes in Euclidean spaces which was introduced in Sec. 5.3 and will be further developed in Sec. 8.

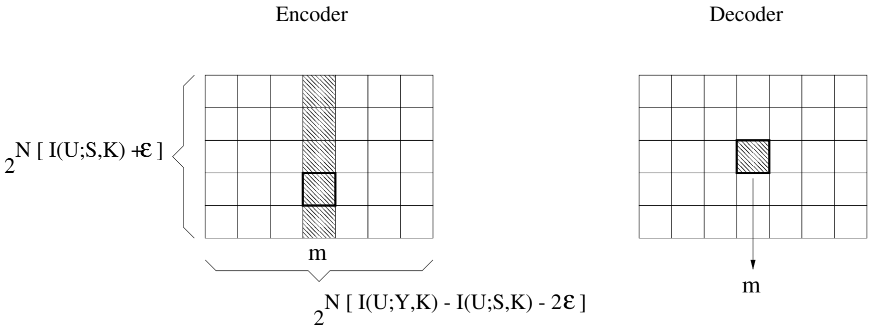


Fig. 15. Random binning technique.

Attack Channels With Memory. Recently, Somekh-Baruch and Merhav [42] have shown that the capacity formula (64) holds under milder assumptions on the attacks and decoder. They assume the a.s. distortion constraints (2) and (4). The decoder does not know the attack channel A^N, which is any channel that satisfies (4) (A^N has arbitrary memory). Capacity can again be achieved using a random binning scheme similar to the one described above, and a particular universal decoder based on the method of types [24,11], i.e., based on the empirical first-order statistics of the pairs (u^N, y^N), for all possible codewords $u^N \in \mathcal{C}$.

7.2. *Gaussian channels*

Theorem 1 can be generalized to the case of infinite alphabets $\mathcal{S}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{U}$, $\mathcal{K}$. The case of Gaussian S and squared–error distortion measure is of considerable practical and theoretical interest, as it becomes possible to explicitly compute the distributions that achieve capacity, leading to insightful results. We refer to this case as the Gaussian channel. Let $\mathcal{S} = \mathcal{X} = \mathcal{Y}$ be the set $\mathbb{R}$ of real numbers, and $d_1(x,y) = d_2(x,y) = (x-y)^2$ be the squared-error metric. Also let $S \sim \mathcal{N}(0, \sigma^2)$, meaning that S follows a Gaussian distribution with mean 0 and variance σ^2.

A remarkable result is that *the data-hiding capacity is the same for both blind and nonblind data hiding problems.* Under the average distortion constraints (1) and (5), we obtain [37]

$$C = C_G(\sigma^2, D_1, D_2) \triangleq \begin{cases} \frac{1}{2}\log\left(1 + \frac{D_1}{D}\right) & : \text{ if } D_1 \le D_2 < \sigma^2, \\ 0 & : \text{ if } D_2 \ge \sigma^2 \end{cases} \tag{66}$$

where $D \triangleq \frac{\sigma^2(D_2 - D_1)}{\sigma^2 - D_2}$. When $D_2 < \sigma^2$, the optimal distributions turn out to be Gaussian test channels [11,37,36], see Fig. 16.

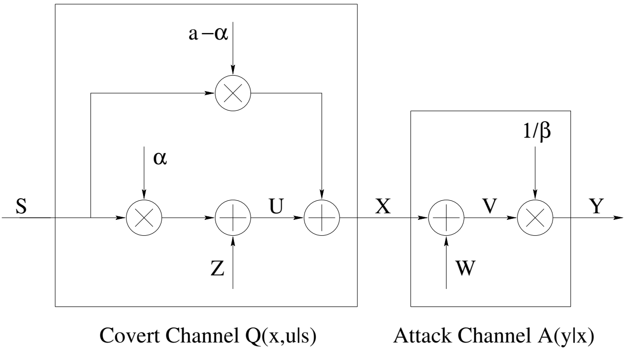


Fig. 16. Optimal data–hiding and attack strategies for Gaussian host data $S \sim \mathcal{N}(0, \sigma^2)$. Here $Z \sim \mathcal{N}(0, aD_1)$ and $W \sim \mathcal{N}(0, \beta(D_2 - D_1))$ are mutually independent random variables, where $a = 1 - D_1/\sigma^2$ and $\beta = \frac{\sigma^2}{\sigma^2 - D_2}$. The optimal channels $p(x|s)$ and $A(y|x)$ are Gaussian test channels with distortion levels D_1 and $D_2 - D_1$, respectively. For public data hiding, $\alpha = \frac{aD_1}{aD_1 + D}$; for public data hiding, one may choose $\alpha = a$.

Closely related to this result is one derived by Costa [10] in 1983 for communications on an additive white Gaussian noise channel (with power D_2) in the presence of an i.i.d. Gaussian interference (with power σ^2) that is known at the encoder but not at the decoder. When the channel input power is constrained not to exceed D_1, Costa showed that the capacity of the channel is exactly the same as if the interference was also known to the decoder: $C = \frac{1}{2} \log \left(1 + \frac{D_1}{D_2}\right)$. The analogy to the data hiding problem is remarkable: the host signal S^N plays the role of the known interference. Capacity in the data hiding problem is slightly lower than in the Costa problem because the optimal Gaussian attack is not additive; however, the gap vanishes in the low-distortion limit ($D_1/\sigma^2 \to 0$ and $D_2/\sigma^2 \to 0$). In this case, we have

$$\alpha = \frac{WNR}{1 + WNR} \tag{67}$$

which admits an elegant MMSE (minimum mean squared error) interpretation [20]; also see (48).

Additional extensions of Costa's result have recently appeared [9,48,47]. In particular, the capacity formula $C = \frac{1}{2} \log \left(1 + \frac{D_1}{D_2}\right)$ is still valid if the interference S^N is *any finite-power sequence*, for *any* values of D_1 and D_2.

Similarly to the finite-alphabet case [42], the capacity for the following two data hiding games are identical: (i) the game with average distortion constraint (3) and memoryless attack channel, known to the decoder, and (ii) the game subject to the a.s. distortion constraint (4) with a decoder uninformed about the attack channel [9].

The optimal decoding rule for Fig. 16 is a minimum-distance decoding rule:

$$\hat{u}^N = \underset{u^N \in \mathcal{C}}{\operatorname{argmax}} \, p\left(u^N | y^N\right) = \underset{u^N \in \mathcal{C}}{\operatorname{argmin}} \, \|u^N - \gamma y^N\|^2 \tag{68}$$

where $\gamma \sim \alpha$ as $D_1/\sigma^2 \to 0$ and $D_2/\sigma^2 \to 0$. For large N, we have $\|u^N\|^2 \sim N\sigma_u^2$, and (68) is asymptotically equivalent to a correlation rule:

$$\hat{u}^N \sim \underset{u^N \in \mathcal{C}}{\operatorname{argmax}} < u^N, y^N > . \tag{69}$$

This rule is remarkable in its simplicity and robustness. For instance (69) is also optimal if the attacker is allowed to scale the output of the Gaussian channel by an arbitrary factor, because all correlations are scaled by the same factor. Also (69) turns out to be the optimal universal decoding rule in Cohen and Lapidoth's setup [9].

The property that capacity is the same whether or not S^N is known at the decoder is illustrated in Fig. 17 using sphere-packing arguments. Assume that $D_1, D_2 << \sigma^2$. With overwhelming probability, the scaled codewords $\frac{1}{\alpha}U^N$ live in a large sphere of radius $\sqrt{N\sigma^2(1+\epsilon)}$ centered at 0. The encoder in the random binning construction selects a scaled codeword $\frac{1}{\alpha}U^N$ inside the medium-size sphere of radius $\sqrt{ND_1/\alpha^2}$ centered at S^N. There are approximately 2^{NC} codewords (one for each possible message m) within this medium-size sphere. The received data vector Y^N lies within a small sphere of radius $\sqrt{N\sigma_v^2}$ centered at $\frac{1}{\alpha}U^N$. Decoding by joint typicality means decoding Y^N to the center of the closest small sphere. To yield a vanishing probability of error, the small spheres should have statistically negligible overlap. The number of distinguishable messages, 2^{NC}, is independent of the size of the large sphere ($N\sigma^2$).

8. Capacity of Constrained Systems

While the theory above provides fundamental limits for reliable data hiding, it does not say how to construct practical codes. The codes used to prove the capacity theorems are random codes which cannot be used in practice due to the exponential complexity of the storage and encoding and decoding procedures.

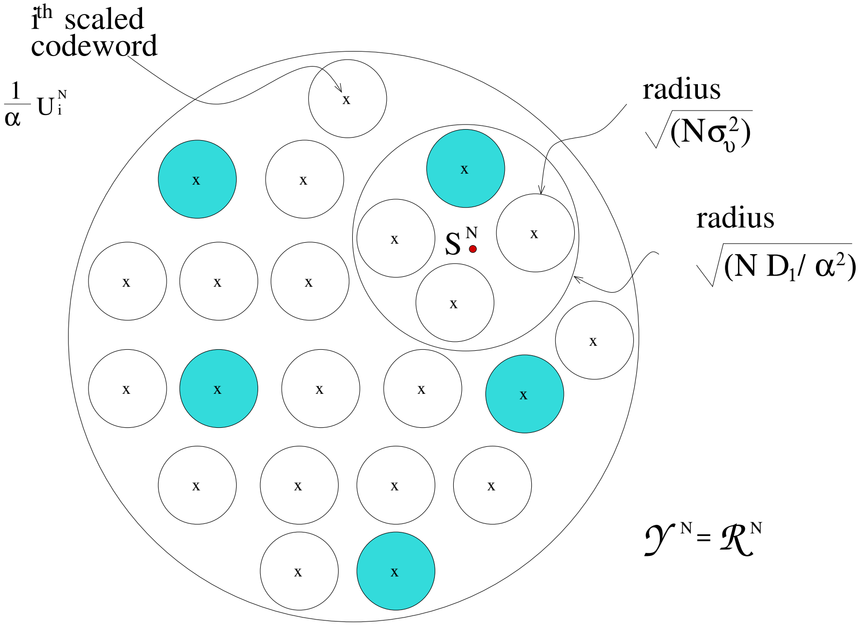

i^th scaled codeword
$\frac{1}{\alpha} U_i^N$
radius $\sqrt{(N\sigma_\upsilon^2)}$
radius $\sqrt{(N D_1/\alpha^2)}$
S^N
x
$\mathcal{Y}^N = \mathcal{R}^N$

Fig. 17. Sphere-packing interpretation of blind Gaussian information hiding. Shaded spheres are indexed by the same message m.

The codes mentioned in Sec. 5.3.2 are practical, but is their performance good enough to approach the unconstrained capacity (66)? Recently Erez and Zamir proved that the answer is yes [19]. Roughly speaking, this requires the use of lattices with nearly spherical Voronoi cells. The information-bearing sequence U^N selected by the lattice encoder (28) plays the same role as the sequence U^N in the random-binning technique of Sec. 7.

For any practical lattice code, one would like to quantify the performance gap relative to an unconstrained system. To illustrate this problem, we consider the case of scalar quantizers.

8.1. *Capacity of scalar QIM systems*

Equation (53) describes the transmission of two possible length-N codewords d_0^N and d_1^N over the MAN channel of Fig. 12. The channel adds independent samples $V_1, \cdots, V_N$ to the input codewords. The addition is modulo Δ, the step size of the scalar quantizer. Referring to (41), the noise V has two parts: self-noise due to quantization and aliased attacker's noise.

The tradeoff parameter α controls the probability distribution of V. If we want to transmit many codewords (as described in Sec. 6.4), what is the maximum rate of reliable transmission?

The answer is given by analyzing the MAN channel of Fig. 12. The maximum rate of reliable transmission for scalar QIM is

$$R_\alpha^{\mathrm{S-QIM}} = \max_{p_D} I(D; \tilde{Y}) \tag{70}$$

where p_D is a probability distribution over the input alphabet $\mathcal{D}$. If the codeword letters are in the binary alphabet $\mathcal{D} = \{\pm\frac{\Delta}{4}\}$ (as was assumed in Sec. 6.3.2), the maximizing distribution is symmetric: $p_D(-\frac{\Delta}{4}) = p_D(\frac{\Delta}{4}) = \frac{1}{2}$.

The value of α that maximizes $R_\alpha^{\mathrm{S-QIM}}$ is obtained numerically and is not the same as (67). A good approximation proposed by Eggers *et al* [17] is $\alpha_{\mathrm{opt}}^{\mathrm{S-QIM}} = \sqrt{\frac{WNR}{WNR+2.71}}$. Both the exact value and its approximation are close to (67) for $WNR \geq 1$. Fig. 18 shows capacity as a function of WNR for scalar QIM and compares it with the capacity expression (66) for unconstrained systems. The gap is approximately 2 dB at a rate of 0.5 bit/sample.

Since the input to the MAN channel is binary-valued, capacity cannot exceed 1 bit/sample. This restriction can be eliminated by allowing the size of the input alphabet $\mathcal{D}$ to be greater than two. The largest nontrivial input alphabet is $[0, \Delta]$. The capacity of this less constrained system can be evaluated using (70). The maximizing p_D is uniform over $[0, \Delta]$. The capacity improvement over binary alphabets is insignificant at rates below 0.7 bit/sample. The gap to capacity is equal to the shaping gain of scalar quantizers, $\frac{1}{2}\log_2 \frac{2\pi e}{12} \approx 0.254$ bit, at high WNR's [19].

Further improvements can be obtained by replacing scalar quantizers with L-dimensional vector quantizers (as explained in Sec. 5.3) and evaluating the capacity formula (70), where p_D is now a probability distribution over the Voronoi cell $\mathcal{V}$.

8.2. *Capacity of sparse QIM systems*

It is easy to relate the capacities of QIM and sparse QIM systems (Sec. 5.2). We have

$$R_{\alpha,\tau}^{\mathrm{sparse}}(WNR) = \tau R_\alpha^{\mathrm{S-QIM}}(WNR/\tau), \quad 0 < \alpha, \tau \leq 1 \tag{71}$$

and thus $C_\tau^{\mathrm{sparse}}(WNR) \stackrel{\triangle}{=} \max_\alpha R_{\alpha,\tau}^{\mathrm{sparse}}(WNR) = \tau C^{\mathrm{S-QIM}}(WNR/\tau)$.

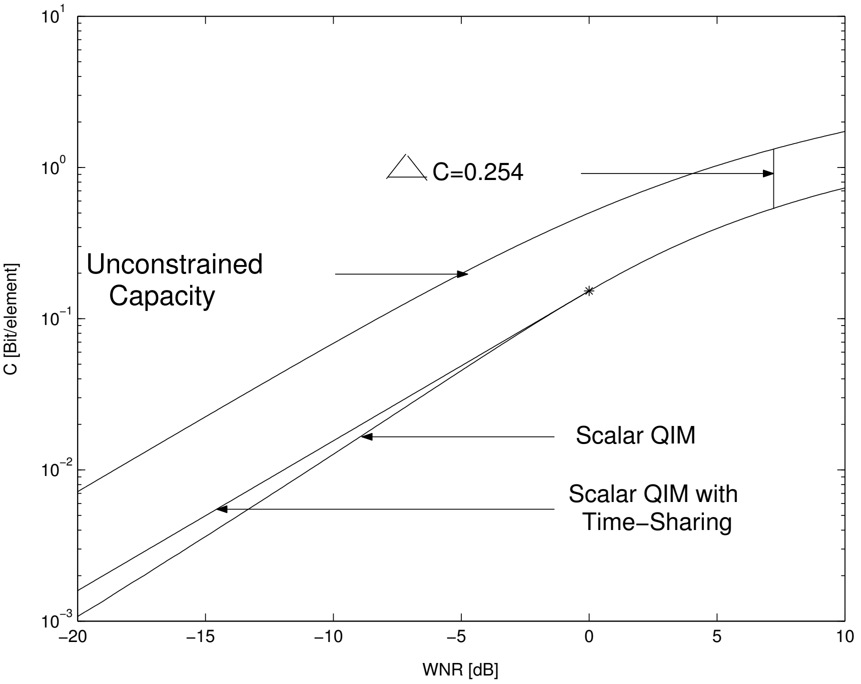


Fig. 18. Capacity *vs* WNR for scalar QIM.

Based on numerical experiments on scalar quantizers, Eggers *et al* [17] observed the following properties:

(1) For WNR above a certain critical value WNR^*, the optimal sparsity factor is $\tau = 1$, i.e., the system is the same as a standard nonsparse QIM system.

(2) For WNR below WNR^*, the optimal τ is less than one, i.e., sparse QIM systems outperform their nonsparse counterparts.

Interestingly, this property is related to information-theoretic time-sharing ideas: the curve $C^{\mathrm{S-QIM}}(WNR)$ is nonconvex at low $WNR's$, and the curve $C^{\mathrm{sparse}}(WNR)$ the upper convex envelope of $C^{\mathrm{S-QIM}}(WNR)$. Thus

$$C^{\mathrm{sparse}}(WNR) = \begin{cases} \frac{WNR}{WNR^*} C^{\mathrm{S-QIM}}(WNR^*) & : \ WNR \le WNR^* \\ C^{\mathrm{S-QIM}}(WNR) & : \ \text{else} \end{cases} \tag{72}$$

is a straight line for $0 \le WNR \le WNR^*$ and coincides with $C^{\mathrm{S-QIM}}(WNR)$ beyond WNR^*. Here WNR^* is the unique solution to the nonlinear equation

$$\frac{d}{dWNR} \ln C^{\mathrm{S-QIM}}(WNR)\bigg|_{WNR=WNR^*} = \frac{1}{WNR^*}.$$

In conclusion, sparse QIM methods are advantageous at low WNR but not at high WNR.

If higher-dimensional quantizers are used, the resulting capacity curve $C_L^{\mathrm{QIM}}(WNR)$ approaches the actual capacity function $C(WNR) = \frac{1}{2}\log_2(1+WNR)$ which is convex and thus cannot be improved by convexification.

8.3. *Parallel Gaussian channels*

Real-world signals such as images do not follow i.i.d. Gaussian models; however they can be decomposed into approximately independent Gaussian components [36]. Data-hiding capacity can be evaluated by solving a certain power-allocation problem, as described below.

Assume S^N is a collection of K independent sources S_k, $1 \le k \le K$, each producing N_k i.i.d. Gaussian random variables from the distribution $\mathcal{N}\left(0, \sigma_k^2\right)$, where $\sum_{k=1}^K N_k = N$. Thus, we have K parallel Gaussian channels, with samples $\{S_k\,(n)\}$, and rates $r_k = N_k/N$, $1 \le k \le K$. The distortion metric is squared error. Let

$$d_{1k} = \frac{1}{N_k}\sum_{n=1}^{N_k} \mathbb{E}\left[X_k(n) - S_k(n)\right]^2 \quad \text{and} \quad d_{2k} = \frac{1}{N_k}\sum_{n=1}^{N_k} \mathbb{E}\left[Y_k(n) - S_k(n)\right]^2 \tag{73}$$

be the distortions introduced by the embedder and the attacker in channel k, respectively. We have distortion constraints

$$\sum_{k=1}^K r_k d_{1k} \le D_1 \quad \text{and} \quad \sum_{k=1}^K r_k d_{2k} \le D_2. \tag{74}$$

As in the Gaussian case, capacity is the same for both blind and nonblind data hiding [37,36]:

$$C = \max_{\{d_{1k}\}} \min_{\{d_{2k}\}} \sum_{k=1}^K r_k C_G(\sigma_k^2, d_{1k}, d_{2k}) \tag{75}$$

where the maximization and minimization over power allocations are subject to the distortion constraints (74). The capacity-achieving distributions are product distributions, i.e., the K channels are decoupled.

9. Desynchronization Attacks

In addition to noise attacks, an attacker may introduce delays and scaling factors (fixed or time-varying) in an attempt to desynchronize the decoder. The perceptual effects of such operations are normally quite weak, but the effects on decoding performance can be devastating. Thus one can ask three basic questions:

(1) How does the performance of the basic QIM decoders degrade under such operations?
(2) What is the capacity of the data-hiding systems if the distortion metric does not penalize delays and scaling factors?[g]
(3) How can one improve the basic QIM decoder to better cope with desynchronization attacks?

This line of research has recently gained some interest. For simplicity, we consider five simple desynchronization attacks.

(1) **Offset.** Let $x_\theta(n) = \theta + x(n)$.
(2) **Amplitude scaling.** Let $x_\theta(n) = \theta x(n)$.
(3) **Cyclic Delay.** Denote by x_θ the signal x cyclically shifted by θ, i.e., $x_\theta(n) = x(n - \theta \bmod N)$ if θ is an integer. For noninteger θ, we use the more general formula $x_\theta(n) = \sum_{i=1}^{N} x(i)\varphi(n-i)$ where $\varphi(t) = \frac{\sin \pi t}{N \sin \pi t/N}$ is the periodic sinc interpolating function.
(4) **Erasures.** Samples $x(n)$ are erased with a certain probability, resulting in a shortened received sequence.
(5) **Insertions.** New values are inserted in the sequence $x(n)$ resulting in a longer received sequence.

Performance of Basic QIM Decoders. The noise V at the decoder is still a weighted average of quantization noise and attacker's noise, however a new term is added to the attacker's noise in the first three cases.

(1) For an offset attack, the new term is simply the offset θ. If w is zero-mean, the mean-squared error (MSE) of the attack noise is increased from $D_w = \frac{1}{N}\|w\|^2$ to $D_w + \theta^2$, which is significant if $|\theta| > \sqrt{D_w}$.
(2) For an amplitude scaling attack, the new term is equal to $(\theta - 1)x(n)$. If $w(n)$ is independent of $x(n)$, the MSE of the attack noise becomes $D_w + (\theta - 1)^2 \frac{1}{N}\|x\|^2$. This effect is significant if $|\theta - 1|$ exceeds the root noise-to-host power ratio, $\frac{\|w\|}{\|x\|}$.

[g] The squared-error metric penalizes delays and scaling factors, unlike some perceptual metrics.

(3) For a cyclic delay, the MSE of the attack noise is asymptotic to $D_w + \theta^2 \frac{1}{N}\|x'\|^2$ as $\theta \to 0$, where $x'(n) = \sum_{i=0}^{N-1} x(i)\varphi'(n-i)$ denotes the sampled derivative of the signal x. This effect is significant if $|\theta - 1| > \frac{\|w\|}{\|x'\|}$.
For θ a nonzero integer, the probability of error is close to 1, except if a cyclic code is used [35].

(4) Erasures and insertions can have a similar catastrophic effect.

Therefore, the effect of even mild desynchronization attacks on unsuspecting QIM decoders can be devastating.

Capacity. As is the case with more traditional communication problems [24], desynchronization attacks have only a benign effect on capacity [39,36]. The poor performance of basic QIM decoders under desynchronization attacks should thus be attributed to the suboptimality of these decoders rather than a fundamental performance limit.

Improved QIM Systems. Several ideas are being developed in the literature to better cope with desynchronization attacks. These include the use of pilot sequences [17,31] for estimating desynchronization parameters, cyclic codes for coping with integer delays [35], Reed-Solomon codes for coping with an equal number of insertions and deletions [41], and synchronization codes for coping with more general insertions, deletions and substitutions [16,8].

10. Data Hiding in Images

Several papers, including [8,41,30], have recently studied quantization-based codes for embedding data in images. The paper [30] directly illustrates the parallel-Gaussian channel theory, as described below.

Wavelet image coefficients are assumed to be Gaussian with zero means and location-dependent variances. The coefficients are conditionally independent, given the variance field. They are grouped into channels containing coefficients with similar variances. This yields a parallel Gaussian model, as described in Sec. 8.3.

The embedding algorithm is outlined in Table 2. The attacker and decoder are assumed to know the variances of the wavelet coefficients of the host data.[h] The optimal attack is Gaussian, with power allocation $\{d_{2k}\}$

[h] Hence this scheme is *semi-blind*; at the decoder S^N is unknown, but the corresponding variances are known. In practice, the variances would be estimated from the received data.

Table 2. Quantization-Based Embedding Algorithm.

Step 1	Apply a discrete wavelet transform to the host image.
Step 2	Estimate the variances of the wavelet coefficients at each scale and location.
Step 3	Group wavelet coefficients with similar variances, and treat each group as a channel.
Step 4	Solve the power allocation problem (75), obtaining $\{d_{1k}\}$ and $\{d_{2k}\}$.
Step 5	Choose a target bit rate R_k below $C_k = C_G(\sigma_k^2, d_{1k}, d_{2k})$ for each channel k. For instance, $R_k = 0.1C_k$.
Step 6	Embed data in each channel using a Gaussian code from Sec. 5.3.2.
Step 7	Apply the inverse wavelet transform to the modified wavelet coefficients. This yields the watermarked image.

derived in Step 4 of Table 2. The decoder performs MAP decoding in each channel, according to (68).

Results are given for the 512×512 image Lena, using Daubechies' length-8 orthogonal wavelets and 64 parallel Gaussian channels. A value of $D_1 = 10$ is chosen such that the embedding distortion is just noticeable. For $D_2 = 5D_1$, the operational probability of bit error, P_{be}, is equal to 0.05. This is vastly better than $P_{be} \approx 0.5$ obtained using uniform bit allocation over the channels, and $P_{be} \approx 0.5$ obtained using Chen and Wornell's original QIM (using $\alpha_k = a_k = 1$ instead of the optimal $\alpha_k = \frac{d_{1k}}{d_{1k}+d_{2k}}$). The total bit rate is 398 bits, 10% of NC.

The paper [41] showed how to combine QIM schemes with desynchronization codes. For instance, they embedded 6,301 bits in the image *Lena* and tampered with the image in various ways (cropping, resizing, substitutions, compression, noise, etc.) All 6,301 bits could be successfully decoded.

11. Authentication

So far we have focused on coding problems, in which the decoder knows that one of $|\mathcal{M}|$ possible messages is embedded in the data, and attempts to reliably decode the message. As discussed in Sec. 3.1, the problem is somewhat different when the decoder is possibly presented with a nonmarked signal. The simplest such problem arises when the receiver must perform a binary decision: is the received signal marked (using a known signature) or not. The primary application of this problem is signal authentication.

The basic hypothesis testing setup is given by (30), where H_0 and H_1 are respectively the "unmarked" and "marked" hypotheses. The challenge is to design an embedding code so that the rival p.d.f.'s p_0 and p_1 are as dissimilar as possible.

It turns out that QIM codes outperform SSM codes in this context as well. A signal is marked by fixing a dither sequence d^N and using e.g., scalar QIM with this particular sequence. The probability of error of such schemes has been analyzed in [27]. Such schemes make it also possible to reconstruct the original signal with a guaranteed accuracy [28].

12. Discussion

This paper has reviewed some basic theory for data hiding and examined the connection between this theory and design criteria for good practical codes. Information theory and game theory play a central role in the analysis and allow us to study the tradeoffs between embedding distortion, attack distortion, and embedding rate. These tradeoffs have been evaluated using probability of error and capacity analyses. Related aspects of these problems may be found in recent papers on the characterization of error exponents [29,34,26], on optimal design of the key distribution [38], and on optimal estimation of attack channel parameters [31,33].

These methods can be applied to closely related problems such as fingerprinting [1,32,45,44] and steganography [2,15,39].

Acknowledgments

This work was supported by the US National Foundation under grants CDA 96-24396, CCR 00-81268, and CCR 02-08809. Thanks are due to M. Kesal and A. Goteti for helpful discussions and for making the figures.

References

1. D. Boneh and J. Shaw, "Collusion–Secure Fingerprinting for Digital Data," *IEEE Trans. Info. Thy*, Vol. 44, No. 5, pp. 1897—1905, 1998.
2. C. Cachin, "An Information–Theoretic Model for Steganography," *Proc. 1998 Workshop on Information Hiding*, Portland, Oregon, Lecture Notes in Computer Sciences, Springer–Verlag, 1998.
3. B. Chen and G. W. Wornell, "An Information–Theoretic Approach to the Design of Robust Digital Watermarking Systems," *Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Phoenix, AZ, March 1999.
4. B. Chen and G. W. Wornell, "Quantization Index Modulation Methods: A Class of Provably Good Methods for Digital Watermarking and Information Embedding," *IEEE Trans. Info. Thy*, Vol. 47, No. 4, pp. 1423—1443, May 2001.

5. J. Chou, S. Pradhan, L. El Ghaoui and K. Ramchandran, "A Robust Optimization Solution to the Data Hiding Problem using Distributed Source Coding Principles," *Proc. SPIE*, Vol. 3971, San Jose, CA, Jan. 2000.
6. J. Chou, S. S. Pradhan and Ramchandran, "On the Duality Between Distributed Source Coding and Data Hiding," *Proc. 33rd Asilomar Conf.*, Pacific Grove, pp. 1503—1507, Oct. 1999.
7. J. Chou, S. S. Pradhan and Ramchandran, "Turbo Coded Trellis-Based Constructions for Data Embedding: Channel Coding with Side Information," *Proc. 35th Asilomar Conf.*, Pacific Grove, pp. 305–309, Nov. 2001.
8. J. Chou and K. Ramchandran, "Robust turbo-based data hiding for image and video sources," *Proc. IEEE Int. Conf. on Image Processing*, Rochester, NY, 2002.
9. A. S. Cohen and A. Lapidoth, "The Gaussian Watermarking Game," *IEEE Trans. Info. Thy*, Vol. 48, No. 6, pp. 1639—1667, June 2002.
10. M. Costa, "Writing on Dirty Paper," *IEEE Trans. Info. Thy*, Vol. 29, No. 3, pp. 439—441, May 1983.
11. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, 1991.
12. I. J. Cox, J. Killian, F. T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. Image Proc.*, Vol. 6, No. 12, pp. 1673—1687, Dec. 1997.
13. I. J. Cox, M. L. Miller and J. A. Bloom, *Digital Watermarking*, Morgan-Kaufmann, San Francisco, 2002.
14. I. J. Cox, M. L. Miller and A. L. McKellips, "Watermarking as Communications with Side Information," *Proceedings IEEE*, Special Issue on Identification and Protection of Multimedia Information, Vol. 87, No. 7, pp. 1127—1141, July 1999.
15. O. Dabeer, K. Sullivan, U. Madhow, S. Chandrasekharan, and B. S. Manjunath, "Detection of Hiding in the Least Significant Bit," *Proc. CISS'03*, Baltimore, MD, Mar. 2003.
16. M. C. Davey and D. J. C. Mackay, "Reliable Communication over Channels with Insertions, Deletions and Substitutions," *IEEE Trans. on Information Theory*, Vol. 47, No. 2, pp. 687–698, Feb. 2001.
17. J. J. Eggers, R. Bäuml, R. Tzschoppe and B. Girod, "Scalar Costa Scheme for Information Embedding," *IEEE Transactions on Signal Processing*, Special issue on Data Hiding, Vol. 51, No. 4, pp. 1003—1019, Apr. 2003.
18. J. J. Eggers, J. K. Su and B. Girod, "A Blind Watermarking Scheme Based on Structured Codebooks," *Proc. IEE Secure Images and Image Authentication*, London, UK, Apr. 2000.
19. U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log(1 + SNR)$ on the AWGN Channel with Lattice Encoding and Decoding," *preprint*, May 2001; revised, Sep. 2003.
20. G. D. Forney, Jr., "On the Role of MMSE Estimation in Approaching the Information-Theoretic Limits of Linear Gaussian Channels: Shannon Meets Wiener," *Proc. Allerton Conf.*, Monticello, IL, Oct. 2003.
21. S. I. Gel'fand and M. S. Pinsker, "Coding for Channel with Random Parameters," *Problems of Control and Information Theory*, Vol. 9, No. 1, pp. 19—31, 1980.

22. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer, 1992.
23. M. Kesal, K. M. Mıhçak, R. Kötter and P. Moulin, "Iteratively Decodable Codes for Watermarking Applications," *Proc. 2nd Symposium on Turbo Codes and Related Topics*, Brest, France, Sep. 2000.
24. A. Lapidoth and P. Narayan, "Reliable Communication Under Channel Uncertainty," *IEEE Trans. Info. Thy*, Vol. 44, No. 6, pp. 2148—2177, Oct. 1998.
25. G.-I. Lin, *Digital Watermarking of Still Images Using a Modified Dither Modulation Algorithm*, M. S. thesis, U. of Illinois at Urbana–Champaign, Dept. of Electrical and Computer Engineering, 2000.
26. T. Liu and P. Moulin, "Error exponents for one-bit watermarking," *Proc. ICASSP*, Hong Kong, Apr. 2003.
27. T. Liu and P. Moulin, "Error exponents for watermarking game with squared-error constraints," *Proc. Int. Symp. on Info Theory*, Yokohama, Japan, July 2003.
28. E. Martinian and G. W. Wornell, "Authentication with Distortion Constraints", *Proc. IEEE Int. Conf. on Image Processing*, pp. II.17—20, Rochester, NY, 2002.
29. N. Merhav, "On Random Coding Error Exponents of Watermarking Codes," *IEEE Trans. Info Thy*, Vol. 46, No. 2, pp. 420—430, Mar. 2000.
30. M. K. Mıhçak and P. Moulin, "Information-Embedding Codes Matched to Local Gaussian Image Models," *Proc. IEEE Int. Conf. on Im. Proc.*, Rochester, NY, 2002.
31. P. Moulin, "Embedded-Signal Design for Channel Parameter Estimation. Part II: Quantization Embedding," *Proc. IEEE Statistical Signal Processing Workshop*, St Louis, MO, Sep. 2003.
32. P. Moulin and A. Briassouli, "The Gaussian Fingerprinting Game," *Proc. CISS'02*, Princeton, NJ, March 2002.
33. P. Moulin and A. Ivanović, "The Fisher Information Game for Optimal Design of Synchronization Patterns in Blind Watermarking," *Proc. IEEE Int. Conf. on Image Processing*, Thessaloniki, Greece, Oct. 2001.
34. P. Moulin and A. Ivanović, "The Zero-Rate Spread-Spectrum Watermarking Game," *IEEE Transactions on Signal Processing*, Vol. 51, No. 4, pp. 1098—1117, Apr. 2003.
35. P. Moulin and R. Koetter, "Theory and Codes for Data-Hiding," *in preparation.*
36. P. Moulin and M. K. Mıhçak, "A Framework for Evaluating the Data-Hiding Capacity of Image Sources," *IEEE Trans. on Image Processing*, Vol. 11, No. 9, pp. 1029–1042, Sep. 2002.
37. P. Moulin and M. K. Mıhçak, "The Parallel-Gaussian Watermarking Game," to appear in *IEEE Trans. Info. Thy*, Feb. 2004.
38. P. Moulin and J. A. O'Sullivan, "Optimal Key Design in Information-Embedding Systems," *Proc. CISS'02*, Princeton, NJ, March 2002.
39. P. Moulin and J. A. O'Sullivan, "Information-Theoretic Analysis of Information Hiding," *IEEE Trans. on Information Theory*, Vol. 49, No. 3, pp. 563—593, 2003.

40. H. V. Poor, *An Introduction to Detection and Estimation Theory*, Springer-Verlag, 1994.
41. K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath and S. Chandrasekaran, "Robust Image-Adaptive Data Hiding Using Erasure and Error Correction," *IEEE Trans. on Image Processing*, Vol. 13, No. 12, pp. 1627–1639, December 2004.
42. A. Somekh-Baruch and N. Merhav, "On the Error Exponent and Capacity Games of Private Watermarking Systems," *IEEE Trans. Info. Thy*, Vol. 49, No. 3, pp. 537—562, March 2003.
43. A. Somekh-Baruch and N. Merhav, "On the Capacity Game of Private Fingerprinting Systems Under Collusion Attacks," *Proc. IEEE Int. Symp. on Info. Theory*, Yokohama, Japan, p. 191, July 2003.
44. A. Somekh-Baruch and N. Merhav, "On the Capacity Game of Public Watermarking Systems," *IEEE Trans. Info. Thy*, Vol. 50, No. 3, pp. 511–524, March 2004.
45. Z. J. Wang, M. Wu, H. Zhao, W. Trappe, and K.J.R. Liu, "Resistance of Orthogonal Gaussian Fingerprints to Collusion Attacks," *Proc. ICASSP'03*, Hong Kong, April 2003.
46. R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual Watermarks for Digital Images and Video," *Proceedings IEEE*, Special Issue on Identification and Protection of Multimedia Information, Vol. 87, No. 7, pp. 1108—1126, July 1999.
47. W. Yu *et al.*, "Writing on Colored Paper," *Proc. IEEE Int. Symp. on Info. Thy*, p. 302, Washington, D.C., 2001.
48. R. Zamir, S. Shamai (Shitz), and U. Erez, "Nested Linear/Lattice Codes for Structured Multiterminal Binning," *IEEE Trans. Info. Thy*, Vol. 48, No. 6, pp. 1250—1276, June 2002.

IMAGE STEGANOGRAPHY AND STEGANALYSIS: CONCEPTS AND PRACTICE

Mehdi Kharrazi[1], Husrev T. Sencar[2] and Nasir Memon[2]
[1]*Department of Electrical and Computer Engineering*
[2]*Department of Computer and Information Science*
Polytechnic University, Brooklyn, NY 11201, USA
E-mails: {mehdi, taha, memon}@isis.poly.edu

In the last few years, we have seen many new and powerful steganography and steganalysis techniques reported in the literature. In the following tutorial we go over some general concepts and ideas that apply to steganography and steganalysis. We review and discuss the notions of steganographic security and capacity. Some of the more recent image steganography and steganalysis techniques are analyzed with this perspective, and their contributions are highlighted.

1. Introduction

Steganography refers to the science of “invisible” communication. Unlike cryptography, where the goal is to secure communications from an eavesdropper, steganographic techniques strive to hide the very presence of the message itself from an observer. The general idea of hiding some information in digital content has a wider class of applications that go beyond steganography, Fig. 1. The techniques involved in such applications are collectively referred to as *information hiding*. For example, an image printed on a document could be annotated by metadata that could lead a user to its high resolution version. In general, metadata provides additional information about an image. Although metadata can also be stored in the file header of a digital image, this approach has many limitations. Usually, when a file is transformed to another format (e.g., from TIFF to JPEG or to BMP), the metadata is lost. Similarly, cropping or any other form of image manipulation destroys the metadata. Finally, metadata can only be attached to an image as long as the image exists in the digital form and is lost once the image is printed. Information hiding allows the metadata to travel with the image regardless of the file format and image state (digital or analog).

A special case of information hiding is *digital watermarking*. Digital watermarking is the process of embedding information into digital multimedia content such that the information (the watermark) can later be extracted or detected for a variety of purposes including copy prevention and control. Digital watermarking has become an active and important area of research, and development and commercialization of watermarking techniques is being deemed essential to help address some of the challenges faced by the rapid proliferation of digital content. The key difference between information hiding and watermarking is the absence of an active adversary. In watermarking applications like copyright protection and authentication, there is an active adversary that would attempt to remove, invalidate or forge watermarks. In information hiding there is no such active adversary as there is no value associated with the act of removing the information hidden in the content. Nevertheless, information hiding techniques need to be robust against accidental distortions.

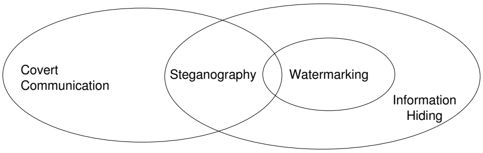


Fig. 1. Relationship of steganography to related fields.

Unlike information hiding and digital watermarking, the main goal of steganography is to communicate securely in a completely undetectable manner. Although steganography is an ancient art, first used against the persian by the romans, it has evolved much through the years.

In the following tutorial we focus on some general concepts and ideas that apply across the field of steganography. The rest of this tutorial is organized as follows: in section 2 we first define the problem which steganography tries to address and introduce to the reader some terminologies commonly used in the field. In section 3 we go over different approaches in defining security. In section 4, the notion of steganographic capacity is discussed, section 5 goes over some embedding techniques, and in sections 6 some steganalysis techniques are reviewed. We conclude in section 7.

2. General Concepts

In this section we go over the concepts and definitions used in the field of steganography. We first start by going over the framework in which steganography is usually presented and then go over some definitions.

The modern formulation of steganography is often given in terms of the *prisoner's problem* [41] where Alice and Bob are two inmates who wish to communicate in order to hatch an escape plan. However, all communication between them is examined by the warden, Wendy, who will put them in solitary confinement at the slightest suspicion of covert communication. Specifically, in the general model for steganography, illustrated in Fig. 2, we have Alice wishing to send a secret message m to Bob. In order to do so, she "embeds" m into a *cover-object* c, and obtains a *stego-object* s. The stego-object s is then sent through the public channel. Thus we have the following definitions:

Cover-object: refers to the object used as the carrier to embed messages into. Many different objects have been employed to embed messages into for example images, audio, and video as well as file structures, and html pages to name a few.

Stego-object: refers to the object which is carrying a hidden message. So given a cover object and a message, the goal of the steganographer is to produce a stego object which would carry the message.

In a *pure steganography* framework, the technique for embedding the message is unknown to Wendy and shared as a secret between Alice and Bob. However, it is generally considered that the algorithm in use is not secret but only the key used by the algorithm is kept as a secret between the two parties, this assumption is also known as Kerchoff's principle in the field of cryptography. The secret key, for example, can be a password used to seed a pseudo-random number generator to select pixel locations in an image cover-object for embedding the secret message (possibly encrypted). Wendy has no knowledge about the secret key that Alice and Bob share, although she is aware of the algorithm that they could be employing for embedding messages.

The warden Wendy who is free to examine all messages exchanged between Alice and Bob can be passive or active. A *passive* warden simply examines the message and tries to determine if it potentially contains a hidden message. If it appears that it does, she suppresses the message and/or takes appropriate action, else she lets the message through without any action. An *active* warden, on the other hand, can alter messages deliberately, even

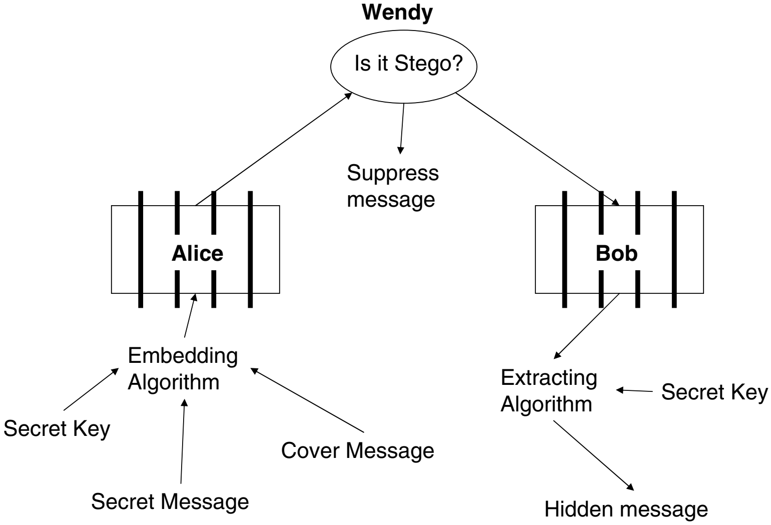


Fig. 2. General model for steganography.

though she does not see any trace of a hidden message, in order to foil any secret communication that can nevertheless be occurring between Alice and Bob. The amount of change the warden is allowed to make depends on the model being used and the cover-objects being employed. For example, with images, it would make sense that the warden is allowed to make changes as long as she does not alter significantly the subjective visual quality of a suspected stego-image. In this tutorial we assume that no changes are made to the stego-object by the warden Wendy.

Wendy should not be able to distinguish in any sense between cover-objects (objects not containing any secret message) and stego-objects (objects containing a secret message). In this context, *steganalysis* refers to the body of techniques that aid Wendy in distinguishing between cover-objects and stego-objects. It should be noted that Wendy has to make this distinction without any knowledge of the secret key which Alice and Bob may be sharing and sometimes even without any knowledge of the specific algorithm that they might be using for embedding the secret message. Hence steganalysis is inherently a difficult problem. However, it should also be noted that Wendy does not have to glean anything about the contents of the secret message m. Just determining the existence of a hidden message is enough. This fact makes her job a bit easier.

The development of techniques for steganography and the wide-spread availability of tools for the same have led to an increased interest in steganalysis techniques. The last two years, for example, have seen many new and powerful steganalysis techniques reported in the literature. Many of such techniques are specific to different embedding methods and indeed have shown to be quite effective in this regard. We will review these techniques in the coming sections.

3. Steganographic Security

In steganography, unlike other forms of communications, one's awareness of the underlying communication between the sender and receiver defeats the whole purpose. Therefore, the first requirement of a steganographic system is its *undetectability.* In other words, a steganographic system is considered to be *insecure*, if the warden Wendy is able to differentiate between cover-objects and stego-objects.

There have been various approaches in defining and evaluating the *security* of a steganographic system. Zollner et al. [50] were among the first to address the undetectability aspect of steganographical systems. They provide an analysis to show that information theoretically secure steganography is possible if embedding operation has a random nature and the embedded message is independent from both the cover-object and stego-object. These conditions, however, ensure undetectability against an attacker who knows the stego-object but has no information available about the indeterministic embedding operation. That is, Wendy has no access to the statistics, distribution, or conditional distribution of the cover-object.

On the other hand, [21,38] approached steganographic security from a complexity theoretic point of view. Based on cryptographic principles, they propose the design of encryption-decryption functions for steganographic embedding and detection. In this setting, the underlying distribution of the cover-objects is known by the attacker, and *undetectability* is defined in a conditional sense as the inability of a polynomial-time attacker (Wendy) to distinguish the stego-object from a cover-object. This model assumes that stego-object is a distorted version of the cover-object, however, it does not attempt to probabilistically characterize the stego object.

In [7], Cachin defined the first steganographic security measure that quantifies the information theoretic security of a stegosystem. His model assigns probability distributions to cover-object and stego-object under which they are produced. Then, the task of Wendy is to decide whether the observed object is produced according to known cover-object distribution or not. In the best case scenario, Wendy also knows the distribution of stego-object and makes a decision by performing a binary hypothesis test. Consequently, the detectability of a stegosystem is based on relative entropy between the probability distributions of the cover-object and stego-object, denoted by P_c and P_s, respectively, i.e.,

$$D(P_c||P_s) = \int P_c \log \frac{P_c}{P_s}. \tag{1}$$

From this equation, we note that $D(P_c||P_s)$ increases with the ratio $\frac{Pc}{Ps}$ which in turn means that the reliability of steganalysis detector will also increase. Accordingly, a stego technique is said to be perfectly secure if $D(P_c||P_s) = 0$ (P_c and P_s are equal), and ϵ-secure if the relative entropy between P_c and P_s is at most ϵ, $D(P_c||P_s) \leq \epsilon$. Perfectly secure algorithms are shown to exist, although they are impractical [7]. However, it should be noted that this definition of security is based on the assumption that the cover-object and stego-object are independent, identically distributed (i.i.d.) vectors of random variables.

Since Wendy uses hypothesis testing in distinguishing between stego-objects and cover-objects, she will make two types of errors, namely, type-I and type-II errors. A type-I error, with probability α occurs, when a cover-object is mistaken for a stego-object (false alarm rate), and a type-II error, with probability β, occurs when a stego-object is mistaken for a cover-object (miss rate). Thus bounds on these error probabilities can be computed using relative entropy, thereby relating steganographic security to detection error probabilities. Cachin [7] obtains these bounds utilizing the facts that deterministic processing can not increase the relative entropy between two distributions, say, P_c and P_s, and hypothesis testing is a form of processing by a binary function that yields α (P(detect message present | message absent)) and β (P(detect message absent | message present)). Then, the relative entropy between distributions P_c and P_s and binary relative entropy of two distributions with parameters (α,$1-\alpha$) and (β, $1-\beta$) need to satisfy

$$d(\alpha, \beta) \leq D(P_c||P_s), \tag{2}$$

where $d(\alpha, \beta)$ is expressed as

$$d(\alpha, \beta) = \alpha \log \frac{\alpha}{1-\beta} + (1-\alpha) \log \frac{1-\alpha}{\beta}. \tag{3}$$

Then, for an ϵ-secure stegosystem we have

$$d(\alpha, \beta) \leq \epsilon. \tag{4}$$

Consequently, when the false alarm rate is set to zero ($\alpha = 0$), the miss rate is lower bounded as $\beta \geq 2^{-\epsilon}$. It should be noted that the probability of detection error for Wendy is defined as

$$P_e = \alpha P(\text{message absent}) + \beta P(\text{message present}). \tag{5}$$

Based on above equations, for a perfectly secure stegosystem, $\alpha + \beta = 1$, and when a cover-object is equally likely to undergo embedding operation, then $P_e = \frac{1}{2}$. Hence, Wendy's decisions are unreliable.

As one can observe, there are several shortcomings in the above definition of security. While the ϵ-secure definition may work for random bit streams (with no inherent statistical structure), for real-life cover-objects such as audio, image, and video, it seems to fail. This is because, real-life cover-objects have a rich statistical structure in terms of correlation, higher-order dependence, etc. By exploiting these structures, it is possible to design good steganalysis detectors even if the first order probability distribution is preserved (i.e., $\epsilon = 0$) during the embedding process. If we approximate the probability distribution functions using histograms, then, examples such as [20] show that it is possible to design good steganalysis detectors even if the histograms of the cover image and the stego image are the same.

Consider the following embedding example. Let X and Y be two binary random variables such that $P(X = 0) = P(Y = 0) = 1/2$, and let them represent the host and covert message, respectively. Let the embedding function be given by the following:

$$Z = X + Y \bmod 2. \tag{6}$$

We then observe that $D(P_Z || P_X) = 0$ but $E(X - Z)^2 = 1$. Therefore the non-zero mean squared error value may give away enough information to a steganalysis detector even though $D(.) = 0$.

One attempt to overcome the limitations of i.i.d. cover-object model was made by Wang et al. [45] where they extended Cachin's results to multivariate Gaussian case, assuming that cover-object and stego-object are vectors of length N with distributions $P_{\mathbf{c}^N}$ and $P_{\mathbf{s}^N}$, respectively. In the multivariate case, similar to i.i.d. case, undetectability condition requires that the distribution of cover-object is preserved after embedding. However, when this is not possible, the degree of detectability of a stegosystem will depend on the deviation from the underlying distribution and the covariance structure of the cover-object. If the cover-object is jointly Gaussian with zero mean and covariance matrix $R_{\mathbf{c}^N}$, among all distributions (with zero mean and covariance matrix $R_{\mathbf{s}^N}$) the Gaussian distribution for the stego-object minimizes the relative entropy. Then, the detectability of stegosystem can be quantified based on the relative entropy as

$$D(P_{\mathbf{c}^N}||P_{\mathbf{s}^N}) = \frac{1}{2}\left(tr(\hat{R}) - \log(\hat{R} + I_N)\right) \approx \frac{1}{4}tr(\hat{R}^2) \tag{7}$$

where tr(.) denotes the trace of a matrix, I_N is the $N \times N$ identity matrix, and $\hat{R} = R_{\mathbf{c}^N} R_{\mathbf{s}^N}^{-1} - I_N$. Consequently, Wendy's detection error probability, P_e can be lower bounded as [45]

$$P_e > \frac{1}{2}\exp^{-\frac{D(P_{\mathbf{c}^N}||P_{\mathbf{s}^N}) + D(P_{\mathbf{s}^N}||P_{\mathbf{c}^N})}{2}} \tag{8}$$

assuming both hypotheses are equally likely, i.e., $P_e = \frac{1}{2}\alpha + \frac{1}{2}\beta$.

Although [45] addressed the inherent limitation of the ϵ-secure notion of Cachin, [7], by considering non-white cover-objects, due to analytical tractability purposes they limited their analysis to cover-objects that are generated by a Gaussian stationary process. However, as stated before, this is not true for many real-life cover-objects. One approach to rectify this problem is to probabilistically model the cover-objects or their transformed versions or some perceptually significant features of the cover-object and put a constraint that the relative entropy computed using the nth order joint probability distributions must be less than, say, ϵ_n and then force the embedding technique to preserve this constraint. But, it may then be possible, at least in theory, to use $(n+1)$th order statistics for successful steganalysis. This line of thought clearly poses several interesting issues:

- Practicality of preserving nth order joint probability distribution during embedding for medium to large values of n.
- Behavior of ϵ_n depends on the cover message as well as the embedding algorithm. If it varies monotonically with n then, for a desired target value, say, $\epsilon = \epsilon^*$, it may be possible to pre-compute a value of $n = n^*$ that achieves this target.

Of course, even if these nth order distributions are preserved, there is no guarantee that embedding induced perceptual distortions will be acceptable. If such distortions are significant, then it is not even necessary to use a statistical detector for steganalysis!

From a practical point of view, Katzenbeisser et al. [23] propose the idea of using an indistinguishability test to define the security of a stegosystem.

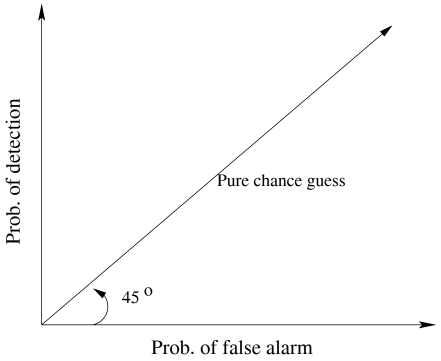


Fig. 3. Detector ROC plane. (Figure taken from [11].)

In their model, Wendy has access to cover-object and stego-object generation mechanisms and uses them consecutively to learn the statistical features of both objects to distinguish between them, rather than assuming their true probability distributions are available. In a similar manner, Chandramouli et al. [11] propose an alternative measure for steganographic security. Their definition is based on the false alarm probability (α), the detection probability ($1-\beta$), and the steganalysis detector's receiver operating characteristic (ROC) which is a plot of α versus $1-\beta$. Points on the ROC curve represent the achievable performance of the steganalysis detector. The average error probability of steganalysis detection is as defined in Eq. (5). Assuming P(message present)=P(message absent) and setting $\alpha = 1-\beta$, then $P_e = 1/2$ and ROC curve takes the form shown in Fig. 3. That is, the detector makes purely random guesses when it operates or forced to operate on the 45 degree line in the ROC plane. Then, the steganographic security can be defined in terms of the deviation of the steganalysis detector's operation curve from the 45 degree ROC line. Correspondingly, a stegosystem can be defined to be $\gamma_{\mathcal{D}}$-secure with respect to a steganalysis detector $\mathcal{D}$ when $|1-\beta_{\mathcal{D}}-\alpha_{\mathcal{D}}| \leq \gamma_{\mathcal{D}}$ where $0 \leq \gamma_{\mathcal{D}} \leq 1$ and $\gamma_{\mathcal{D}} = 0$ refers to the perfect security condition, similar to the ϵ-security notion of Cachin [7].

4. Steganographic Capacity

Steganographic capacity refers to the maximum amount (rate) of information that can be embedded into a cover-object and then can be reliably recovered from the stego-object (or a distorted version), under the constraints of undetectability, perceptual intactness and robustness, depending on whether Wendy is active or passive. Compared to data hiding systems, stegosystems have the added core requirement of undetectability. Therefore, the steganographic embedding operation needs to preserve the statistical properties of the cover-object, in addition to its perceptual quality. On the other hand, if Wendy suspects of a covert communication but cannot reliably make a decision, she may choose to modify the stego-object before delivering it. This setting of steganography very much resembles to data hiding problem, and corresponding results on data hiding capacity can be adapted to steganography [31].

As discussed in the previous section, the degree of undetectability of a stegosystem is measured in terms of a distance between probability distributions $P_{\mathbf{c}^N}$ and $P_{\mathbf{s}^N}$, i.e., $D(P_{\mathbf{c}^N}||P_{\mathbf{s}^N}) \leq \epsilon$ where $\epsilon = 0$ is the perfect security condition. Let $d(\mathbf{c}^N, \mathbf{s}^N)$ be a perceptual distance measure defined between cover-object $\mathbf{c}^N$ and stego-object $\mathbf{s}^N$. When the warden is passive, the steganographic capacity C_p of a perfectly secure stegosystem with embedding distortion limited to P is defined, in terms of random vectors $\mathbf{s}^N$ and $\mathbf{c}^N$, as

$$C_p = \{\sup H(\mathbf{s}^N|\mathbf{c}^N) : P_{\mathbf{c}^N} = P_{\mathbf{s}^N} \text{ and } \frac{1}{N}E[d(\mathbf{c}^N, \mathbf{s}^N)] \leq P\} \tag{9}$$

where $E[.]$ denotes the expected value and supremum is taken over all $P_{\mathbf{s}^N|\mathbf{c}^N}$ for the given constraints. In [31], Moulin et al. discuss code generation (embedding) for a perfectly secure stegosystem with binary i.i.d. cover-object and Hamming distortion measure, and provide capacity results. However, generalization of such techniques to real life cover-objects is not possible due to two reasons. First is the simplistic i.i.d. assumption, and second is the utilized distortion measure as there is no trivial relation between bit error rate and reconstruction quality.

In order to be able to design practical stegosystems, the perfect security condition in Eq. (9) can be relaxed by replacing it with the ϵ-security notion. One way to exploit this is by identifying the perceptually significant and insignificant parts of the cover-object $\mathbf{c}^N$, and preserving the statistics of the significant component while utilizing the insignificant component for embedding. For this, let there be a function g(.) such that $d(\mathbf{c}^N, g(\mathbf{c}^N)) \approx 0$ and $g(\mathbf{c}^N) = g(\mathbf{s}^N)$. Then, Eq. (9) can be modified as

$$C_p = \{\sup H(\mathbf{s}^N|\mathbf{c}^N) : P_{g(\mathbf{c}^N)} = P_{g(\mathbf{s}^N)} \text{ and } \frac{1}{N}E[(d(\mathbf{c}^N, \mathbf{s}^N)] \leq P\} \tag{10}$$

where $D(P_{\mathbf{c}^N}||P_{\mathbf{s}^N}) \leq \epsilon$. This approach requires statistical modelling of the cover-object or of some features of it, which will be modified during embedding. For example, [48,16,10] observe the statistical regularity between pairs of sample values in an image, and provide a framework for (ϵ-secure) embedding in least significant bit (LSB) layer. Similarly, Sallee [39] models AC components of DCT coefficients by Generalized Cauchy distribution and uses this model for embedding. In the same manner, wavelet transformed image coefficients can be marginally modelled by Generalized Laplacian distribution [42]. This approach, in general, suffers due to the difficulty in modelling the correlation structure via higher order joint distributions which is needed to ensure ϵ-security.

In the presence of an active warden, the steganographic capacity can be determined based on the solution of data hiding capacity with the inclusion of undetectability or ϵ-security condition. Data hiding capacity has been the subject of many research works, see, [5,13,29,49,12,8,30,36,37,9] and references therein, where the problem is viewed as a channel communication scenario with side information at the encoder. Accordingly, the solution for the data hiding capacity requires consideration of an auxiliary random variable u that serves as a random codebook shared by both embedder and detector. Let the distorted stego-object be denoted by y, and assume cover-object and stego-object are distorted by amounts P and D during embedding operation and attack, respectively. Since undetectability is the central issue in steganography, we consider the additional constraint of $P_c = P_s$. Then, the steganographic capacity for the active warden case, C_a, is derived, in terms of i.i.d. random variables c, u, s, and y, as

$$C_a = \{\sup I(u,y) - I(u,c) : P_c = P_s, E[(d(c,s)] \leq P, \text{and} E[(d(s,y)] \leq D\} \tag{11}$$

where supremum is taken over all distributions $P_{u|c}$ and all embedding functions under the given constraints. The computation of the steganographic capacity of practical stegosystems, using Equations (9)-(11), still remains to be an open problem due to lack of true statistical models and for reasons of analytical tractability.

Chandramouli et al. [10], from a practical point of view, make an alternative definition of steganographic capacity based on the γ-security notion given in the previous section [11]. They define steganographic capacity from a detection theoretic perspective, rather than information theoretic, as the maximum message size that can be embedded so that a steganalysis detector is only able to make a perfectly random guess about the presence/absence of a covert message. This indicates that the steganographic capacity in the presence of steganalysis varies with respect to the steganalysis detector. Therefore, its formulation must involve parameters of the embedding function as well as that of the steganalysis detector. Assuming N is the number of message carrying symbols, and $\alpha_{\mathcal{D}}^{(N)}$ and $1 - \beta_{\mathcal{D}}^{(N)}$ are the corresponding false alarm and detection probabilities for a steganalysis detector $\mathcal{D}$, the steganographic capacity is defined as

$$N_{\gamma}^{*} = \{\max N \text{ subject to } |1 - \beta_{\mathcal{D}}^{(N)} - \alpha_{\mathcal{D}}^{(N)}| \leq \gamma_{\mathcal{D}}\} \text{ symbols.} \tag{12}$$

Based on this definition, [10] provide an analysis on the capacity of LSB steganography and investigate under what conditions an observer can distinguish between stego-images and cover-images.

5. Techniques for Image Steganography

Given the proliferation of digital images, and given the high degree of redundancy present in a digital representation of an image (despite compression), there has been an increased interest in using digital images as cover-objects for the purpose of steganography. Therefore we have limited our discussion to the case of images for the rest of this tutorial. We should also note that there have been much more work on embedding techniques which make use of the transform domain or more specifically JPEG images due to their wide popularity. Thus to an attacker the fact that an image other than that of JPEG format is being transferred between two entities could hint of suspicious activity.

There have been a number of image steganography algorithm proposed, these algorithm could be categorized in a number of ways:

- Spatial or Transform, depending on redundancies used from either domain for the embedding process.
- Model based or ad-hoc, if the algorithm models statistical properties before embedding and preserves them, or otherwise.
- Active or Passive Warden, based on whether the design of embedder-detector pair takes into account the presence of an active attacker.

In what follows we go over algorithms classified into 3 different sections, based on the more important characteristics of each embedding technique. Some of the techniques which we will discuss below have been successfully broken by steganalysis attacks, which we will go over in Section 6.

5.1. *Spatial domain embedding*

The best widely known steganography algorithm is based on modifying the least significant bit layer of images, hence known as the *LSB technique.* This technique makes use of the fact that the least significant bits in an image could be thought of random noise and changes to them would not have any effect on the image. This is evident by looking at Fig. 4. Although the image seems unchanged visually after the LSBs are modified, the statistical properties of the image changes significantly. We will discuss in the next section of this tutorial how these statistical changes could be used to detect stego images created using the LSB method.

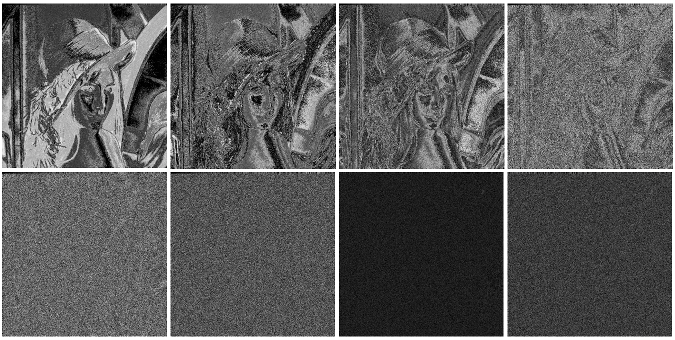


Fig. 4. Bitplane decomposition of image Lena.

In the LSB technique, the LSB of the pixels is replaced by the message to be sent. The message bits are permuted before embedding, this has the effect of distributing the bits evenly, thus on average only half of the LSB's will be modified. Popular steganographic tools based on LSB embedding [14,33,40], vary in their approach for hiding information. Some algorithms change LSB of pixels visited in a random walk, others modify pixels in certain areas of images, or instead of just changing the last bit they increment or decrement the pixel value [40].

Fridrich et al. [18] proposed another approach for embedding in spatial domain. In their method, noise that statistically resemble common processing distortion, e.g., scanner noise, or digital camera noise, is introduced to pixels on a random walk. The noise is produced by a pseudo random noise generator using a shared key. A *parity function* is designed to embed and detect the message signal modulated by the generated noise.

5.2. *Transform domain embedding*

Another category for embedding techniques for which a number of algorithms have been proposed is the transform domain embedding category. Most of the work in this category has been concentrated on making use of redundancies in the DCT (discrete cosine transform) domain, which is used in JPEG compression. But there has been other algorithms which make use of other transform domains such as the frequency domain [1].

Embedding in DCT domain is simply done by altering the DCT coefficients, for example by changing the least significant bit of each coefficient. One of the constraints of embedding in DCT domain is that many of the 64 coefficients are equal to zero, and changing too many zeros to non-zero values will have an effect on the compression rate. That is why the number of bits one could embed in DCT domain, is less that the number of bits one could embed by the LSB method. Also the embedding capacity becomes dependent on the image type used in the case of DCT embedding, since depending on the texture of image the number of non-zero DCT coefficients will vary.

Although changing the DCT coefficients will cause unnoticeable visual artifices, they do cause detectable statistical changes. In the next section, we will discuss techniques that exploit these statistical anomalies for steganalysis. In order to minimize statistical artifacts left after the embedding process, different methods for altering the DCT coefficients have been proposed, we will discuss two of the more interesting of these methods, namely the F5 [46] and Outguess [32] algorithms.

F5 [46] embedding algorithm was proposed by Westfeld as the latest in a series of algorithms, which embed messages by modifying the DCT coefficients. For a review of jsteg, F3 and F4 algorithms that F5 is built on, please refer to [46]. F5 has two important features, first it permutes the DCT coefficients before embedding, and second it employs matrix embedding.

The first operation, namely permuting the DCT coefficients has the effect of spreading the changed coefficients evenly over the entire image. The importance of this operation becomes evident when a small message is used. Let's say we are embedding a message of size m, then if no permutation is done and coefficients are selected in the order they appear, then only the first m coefficients are used. Thus the first part of the image get's fully changed after embedding, and the rest of the image remains unchanged. This could facilitate attacks on the algorithm since the amount of change is not uniform over the entire image. On the other hand when permutation is done, the message is spread uniformly over the image thus the distortion effects of embedding is spread equally and uniformly over the entire image.

The second operation done by F5 is matrix embedding. The goal of matrix embedding is to minimize the amount of change made to the DCT coefficients. Westfeld [46], takes n DCT coefficients and hashes them to k bits. If the hash value equals to the message bits then the next n coefficients are chosen and so on. Otherwise one of the n coefficients is modified and the hash is recalculated. The modifications are constrained by the fact that the resulting n DCT coefficients should not have a hamming distance of more than d_{max} from the original n DCT coefficients. This process is repeated until the hash value matches the message bits. So then given an image, the optimal values for k and n could be selected.

Outguess [32], which was proposed by Provos, is another embedding algorithm which embeds messages in the DCT domain. Outguess goes about the embedding process in two separate steps. First it identifies the redundant DCT coefficients which have minimal effect on the cover image, and then depending on the information obtained in the first step, chooses bits in which it would embed the message. We should note that at the time Outguess was proposed, one of its goals was to overcome steganalysis attacks which look at changes in the DCT histograms after embedding. So Provos, proposed a solution in which some of the DCT coefficients are left unchanged in the embedding process, afterwards these remaining coefficients are adjusted in order to preserve the original histogram of DCT coefficients. As we will see in the steganalysis section both F5 [46], and Outguess [32] embedding techniques have been successfully attacked.

As mentioned before, another transform domain which has been used for embedding is the frequency domain. Alturki et al. [1] propose quantizing the coefficients in the frequency domain in order to embed messages. They first decorrelate the image by scrambling the pixels randomly, which in effect whitens the frequency domain of the image and increases the number of transform coefficients in the frequency domain thus increasing the embedding capacity. As evident from Fig. 5, the result is a salt and pepper image where its probability distribution function resembles a gaussian distribution. The frequency coefficients are then quantized to even or odd multiples of the quantization step size to embed zeros or ones. Then the inverse FFT of the signal is taken and descrambled. The resulting image would be visually incomparable to the original image. But statistically the image changes and as the authors show in their work, the result of the embedding operation is the addition of a gaussian noise to the image.

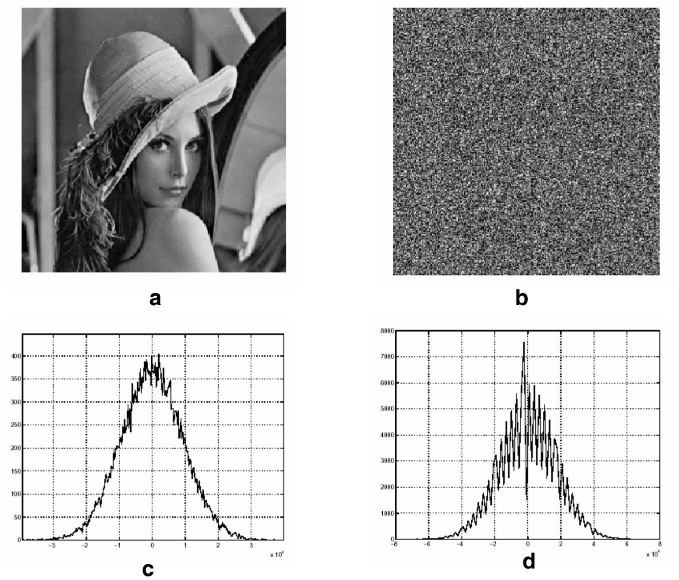


Fig. 5. Frequency domain embedding. a) Original image, b) scrambled image, c) histogram of DFT coefficients, and d) histogram of DFT coefficients after quantization. (Figure taken from [1].)

5.3. *Model based techniques*

Unlike techniques discussed in the two previous subsections, model based techniques try to model statistical properties of an image, and preserve them in the embedding process. For example Sallee [39] proposes a method which breaks down transformed image coefficients into two parts, and replaces the perceptually insignificant component with the coded message signal. Initially, the marginal statistics of quantized (non-zero) AC DCT coefficients are modelled with a parametric density function. For this, a low precision histogram of each frequency channel is obtained, and the model is fit to each histogram by determining the corresponding model parameters. Sallee defines the offset value of coefficient within a histogram bin as a *symbol* and computes the corresponding *symbol probabilities* from the relative frequencies of symbols (offset values of coefficients in all histogram bins).

In the heart of the embedding operation is a non-adaptive arithmetic decoder which takes as input the message signal and decodes it with respect to measured symbol probabilities. Then, the entropy decoded message is embedded by specifying new bin offsets for each coefficient. In other words, the coefficients in each histogram bin are modified with respect to embedding rule, while the global histogram and symbol probabilities are preserved. Extraction, on the other hand, is similar to embedding. That is, model parameters are determined to measure symbol probabilities and to obtain the embedded symbol sequence (decoded message). (It should be noted that the obtained model parameters and the symbol probabilities are the same both at the embedder and detector.) The embedded message is extracted by entropy encoding the symbol sequence.

Another model based technique was proposed by Radhakrishnan et al. [35], in which the message signal is processed so that it would exhibit the properties of an arbitrary cover signal, they call this approach data masking. As argued if Alice wants to send an encrypted message to Bob, the warden Wendy would be able to detect such a message as an encrypted stream since it would exhibit properties of randomness. In order for a secure channel to achieve covertness, it is necessary to preprocess the encrypted stream at the end points to remove randomness such that the resulting stream defeats statistical tests for randomness and the stream is reversible at the other end.

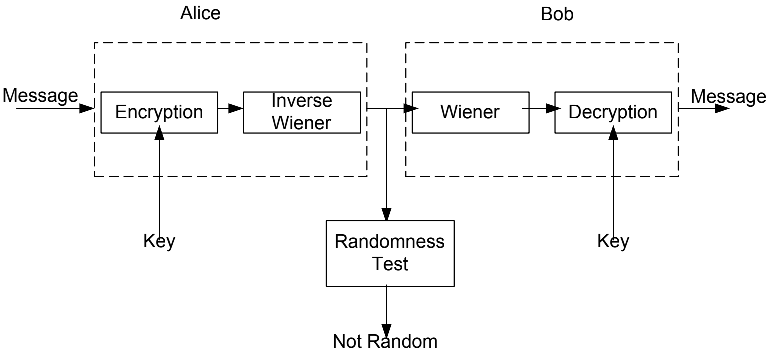


Fig. 6. Proposed System for Secure and Covert Communication. (Figure taken from [35].)

The authors propose Inverse Wiener filtering as a solution to remove randomness from cipher streams as shown in Fig 6. Let us consider the cipher stream as samples from a wide sense stationary (WSS) process, E. We would like to transform this input process with high degree of randomness to another stationary process, A, with more correlation between samples by using a linear filter, H. It is well known that the power spectrum of a WSS input, $A(w)$, to a linear time invariant system will have the output with the power spectrum $E(w)$ expressed as

$$E(w) = |H(w)|^2 A(w). \tag{13}$$

If $E(w)$ is a white noise process, then $H(w)$ is the whitening filter or Wiener filter. Since the encrypted stream is random, its power spectral density is flat and resembles the power spectral density of a white noise process. Then, the desired Wiener filter can be obtained by spectral factorization of $(E(w)/A(w))$ followed by selection of poles and zeros to obtain the minimum phase solution for $H(w)$. The authors discuss how the above method could be used with audio as cover-object in [35], and more recently with images as cover-object in [34].

6. Steganalysis

There are two approaches to the problem of steganalysis, one is to come up with a steganalysis method specific to a particular steganographic algorithm. The other is developing techniques which are independent of the steganographic algorithm to be analyzed. Each of the two approaches has its own advantages and disadvantages. A steganalysis technique specific to an embedding method would give very good results when tested only on that embedding method, and might fail on all other steganographic algorithms. On the other hand, a steganalysis method which is independent of the embedding algorithm might preform less accurately overall but still provide acceptable results on new embedding algorithms. These two approaches will be discussed below and we will go over a few of the proposed techniques for each approach.

Before we proceed, one should note that steganalysis algorithms in essence are called successful if they can detect the presence of a message, and the message itself does not have to be decoded. Indeed, the latter can be very hard if the message is encrypted using strong cryptography. However, recently there have been methods proposed in the literature which in addition to detecting the presence of a message are also able to estimate the size of the embedded message with great accuracy. We consider these aspects to be extraneous and only focus on the ability to detect the presence of a message.

6.1. *Technique specific steganalysis*

We first look at steganalysis techniques that are designed with a particular steganographic embedding algorithm in mind. As opposed to the previous section, where the embedding algorithms were categorized depending on the approach taken in the embedding process, here we categorize the steganographic algorithms depending on the type of image they operate on, which includes Raw images (for example bmp format), Palette based images (for example GIF images), and finally JPEG images.

6.1.1. *Raw images*

Raw images are widely used with the simple LSB embedding method, where the message is embedded in a subset of the LSB (least significant bit) plane of the image, possibly after encryption. An early approach to LSB steganalysis was presented in [48] by Westfeld et al. They note that LSB embedding induces a partitioning of image pixels into Pairs of Values (PoV's) that get mapped to one another. For example the value 2 gets mapped to 3 on LSB flipping and likewise 3 gets mapped to 2. So (2, 3) forms a PoV. Now LSB embedding causes the frequency of individual elements of a PoV to flatten out with respect to one another. So for example if an image has 50 pixels that have a value 2 and 100 pixels that have a value 3, then after LSB embedding of the entire LSB plane the expected frequencies of 2 and 3 are 75 and 75 respectively. This of course is when the entire LSB plane is modified. However, as long as the embedded message is large enough, there will be a statistically discernible flattening of PoV distributions and this fact is exploited by their steganalysis technique.

The length constraint, on the other hand, turns out to be the main limitation of their technique. LSB embedding can only be reliably detected when the message length becomes comparable with the number of pixels in the image. In the case where message placement is known, shorter messages can be detected. But requiring knowledge of message placement is too strong an assumption as one of the key factors playing in the favor of Alice and Bob is the fact that the secret message is hidden in a location unknown to Wendy.

A more direct approach for LSB steganalysis that analytically estimates the length of an LSB embedded message in an image was proposed by Dumitrescu et al. [16]. Their technique is based on an important statistical identity related to certain sets of pixels in an image. This identity is very sensitive to LSB embedding, and the change in the identity can quantify the length of the embedded message. This technique is described in detail below, where our description is adopted from [16].

Consider the partition of an image into pairs of horizontally adjacent pixels. Let $\mathcal{P}$ be the set of all these pixel pairs. Define the subsets X, Y

and Z of $\mathcal{P}$ as follows:

- X is the set of pairs $(u, v) \in \mathcal{P}$ such that v is even and $u < v$, or v is odd and $u > v$.
- Y is the set of pairs $(u, v) \in \mathcal{P}$ such that v is even and $u > v$, or v is odd and $u < v$.
- Z is the subset of pairs $(u, v) \in \mathcal{P}$ such that $u = v$.

After having made the above definitions, the authors make the assumption that statistically we will have

$$|X| = |Y|. \tag{14}$$

This assumption is true for natural images as the gradient of intensity function in any direction is equally likely to be positive or negative.

Furthermore, they partition the set Y into two subsets W and V, with W being the set of pairs in $\mathcal{P}$ of the form $(2k, 2k+1)$ or $(2k+1, 2k)$, and $V = Y - W$. Then $\mathcal{P} = X \cup W \cup V \cup Z$. They call the sets X, V, W and Z as *primary sets.*

When LSB embedding is done pixel values get modified and so does the membership of pixel pairs in the primary sets. More specifically, given a pixel pair (u, v), they identify the following four situations:

00) both values u and v remain unmodified;
01) only v is modified;
10) only u is modified;
11) both u and v are modified.

The corresponding change of membership in the primary sets is shown in Fig. 7.

By some simple algebraic manipulations, the authors finally arrive at the equation

$$0.5\gamma p^2 + (2|X'| - |\mathcal{P}|)p + |Y'| - |X'| = 0. \tag{15}$$

where $\gamma = |W| + |Z| = |W'| + |Z'|$. The above equation allows one to estimate p, i.e., the length of the embedded message, based on X', Y', W', Z' which can all be measured from the image being examined for possible steganography. Of course it should be noted that we cannot have $\gamma = 0$, the probability of which for natural images is very small.

In fact, the pairs based steganalysis described above was inspired by an effectively identical technique, although from a very different approach,

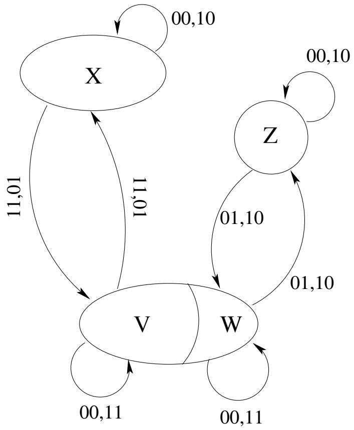


Fig. 7. State transition diagram for sets X, V, W, Z under LSB flipping. (Figure taken from [16].)

called RS-Steganalysis by Fridrich et al. in [19] that had first provided remarkable detection accuracy and message length estimation even for short messages. However, RS-Steganalysis does not offer a direct analytical explanation that can account for its success. It is based more on empirical observations and their modelling. It is interesting to see that the Pair's based steganalysis technique essentially ends up with exactly the same steganalyzer as RS-Steganalysis.

Although the above techniques are for gray scale images, they are applicable to color images by considering each color plane as a gray scale image. A steganalysis technique that directly analyzes color images for LSB embedding and yields high detection rates even for short messages was proposed by Fridrich et al. [17]. They define pixels that are "close" in color intensity to be pixels that have a difference of not more than one count in any of the three color planes. They then show that the ratio of "close" colors to the total number of unique colors increases significantly when a new message of a selected length is embedded in a cover image as opposed to when the same message is embedded in a stego-image (that is an image already carrying a LSB encoded message). It is this difference that enables them to distinguish cover-images from stego-images for the case of LSB steganography.

In contrast to the simple LSB method discussed, Hide [40] increments or decrements the sample value in order to change the LSB value. Thus the techniques previously discussed for LSB embedding with bit flipping do

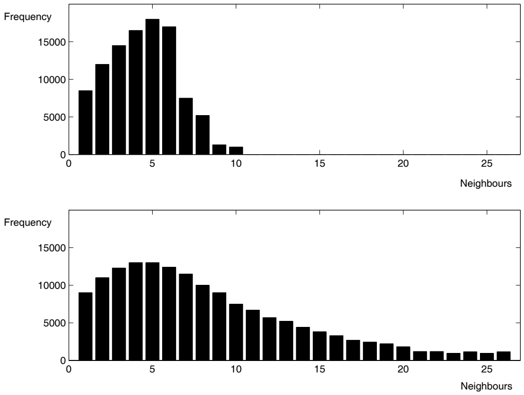


Fig. 8. Neighborhood histogram of a cover image (top) and stego image with 40 KB message embedded (bottom). (Figure taken from [47].)

not detect Hide. In order to detect embedded messages by Hide, Westfeld [47] proposes a similar steganalysis attack as Fridrich et al. [17] where it is argued that since the values are incremented or decremented, 26 neighboring colors for each color value could be created, whereas in a natural image there are 4 to 5 neighboring colors on average. Thus by looking at the neighborhood histogram representing the number of neighbors in one axis and the frequency in the other one would be able to say if the image carries a message. This is clearly seen in Fig 8.

6.1.2. *Palette based images*

Palette based images, like GIF images, are another popular class of images for which there have been a number of steganography methods proposed [27,24,28]. Perhaps some of the earliest steganalysis work in this regard was reported by Johnson et al. [22]. They mainly look at palette tables in GIF images and anomalies caused therein by common stego-tools that perform LSB embedding in GIF images. Since pixel values in a palette image are represented by indices into a color look-up table which contains the actual color RGB value, even minor modifications to these indices can result in annoying artifacts. Visual inspection or simple statistics from such stego-images can yield enough tell-tale evidence to discriminate between stego and cover-images.

In order to minimize the distortion caused by embedding, EzStego [27] first sorts the color pallet so that the color differences between consecutive colors is minimized. It then embeds the message bits in the LSB of the color indices in the sorted pallet. Since pixels which are modified due to the embedding process get mapped neighboring colors in the palette, which are now similar, visual artifacts are minimal and hard to notice. To detect EzStego, Fridrich [20] argues that a vector consisting of color pairs, obtained after sorting the pallet, has considerable structure due to the fact there are a small number of colors in pallet images. But the embedding process will disturb this structure, thus after the embedding the entropy of the color pair vector will increase. The entropy would be maximal when the maximum length message is embedded in to the GIF image. Another steganalysis techniques for EzStego were proposed by Westfeld [48], but the technique discussed above provides a much higher detection rate and a more accurate estimate of the message lengths.

6.1.3. *JPEG images*

JPEG images are the third category of images which are used routinely as cover medium. Many steganalysis attacks have been proposed for steganography algorithms [32,43,46] which employ this category of images. Fridrich [20] has proposed attacks on the F5 and Outguess algorithms, both of which were covered in the previous section. F5 [46] embeds bits in the DCT coefficients using matrix embedding so that for a given message the number of changes made to the cover image is minimized, at the same time it spreads the message over the cover image. But F5 does alter the histogram of DCT coefficients. Fridrich proposes a simple technique to estimate the original histogram so that the number of changes and length of the embedded message could be estimated. The original histogram is simply estimated by cropping the JPEG image by 4 columns and then re-compressing the image using the same quantization table as used before. As is evident in Fig 9, the resulting DCT coefficient histogram would be a very good estimate of the original histogram.

Intuitively, effect of the cropping operation could be reasoned as follows. In a natural image, characteristics are expected to change smoothly with respect to spatial coordinates. That is, image features computed in a portion of image will not change significantly by a slight shift in the computation window. In the same manner, the statistics of the DCT coefficients computed from a shifted partitioning of an image should remain roughly unchanged. However, since in F5, DCT coefficients are tailored by the embedder, cropping of the image (shift in the partitioning) will spoil the

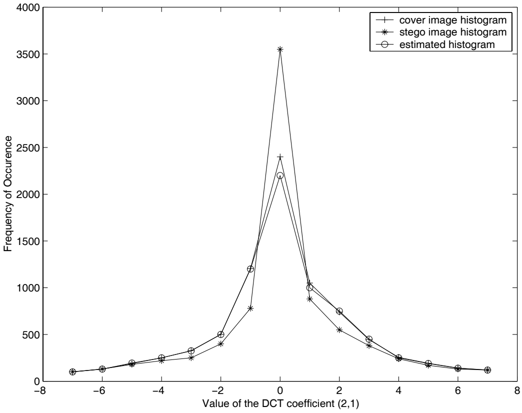


Fig. 9. The effect of F5 embedding on the histogram of the DCT coefficient (2,1). (Figure taken from [20].)

structure created by embedding process, thereby, the coefficient statistics will vary and estimate the original structure.

A second technique proposed by Fridrich [20] deals with the Outguess [32] embedding program. Outguess first embeds information in LSB of the DCT coefficients by making a random walk, leaving some coefficients unchanged. Then it adjusts the remaining coefficients in order to preserve the original histogram of DCT coefficients. Thus the previous steganalysis method where the original histogram is estimated will not be effective. On the other hand when embedding messages in a clean image, noise is introduced in the DCT coefficients, therefore increasing the spatial discontinuities along the 8x8 JPEG blocks. Given a stego image if a message is embedded in the image again there is partial cancellation of changes made to the LSBs of DCT coefficients, thus the increase in discontinuities will be smaller. This increase or lack of increase in the discontinuities is used to estimate the message size which is being carried by a stego image. In a related work Wang et al. [44] use a statistical approach and show how embedding in DCT domain affects differently the distribution of neighboring pixels which are inside blocks or across blocks. These differences could be used to distinguish between clean and stego images.

6.2. *Universal steganalysis*

The steganalysis techniques described above were all specific to a particular embedding algorithm. A more general class of steganalysis techniques pioneered independently by Avcibas et al. [2,3,4] and Farid et al. [25,26], are designed to work with any steganographic embedding algorithm, even an unknown algorithm. Such techniques have subsequently been called *Universal Steganalysis* or *Blind Steganalysis Techniques.* Such approaches essentially design a classifier based on a training set of cover-objects and stego-objects obtained from a variety of different embedding algorithms. Classification is done based on some inherent "features" of typical natural images which can get violated when an image undergoes some embedding process. Hence, designing a feature classification based universal steganalysis technique consists of tackling two independent problems. The first is to find and calculate features which are able to capture statistical changes introduced in the image after the embedding process. The second is coming up with a strong classification algorithm which is able to maximize the distinction captured by the features and achieve high classification accuracy.

Typically, a good feature should be accurate, monotonic, and consistent in capturing statistical signatures left by the embedding process. Detection accuracy can be interpreted as the ability of the measure to detect the presence of a hidden message with minimum error on average. Similarly, detection monotonicity signifies that the features should ideally be monotonic in their relationship to the embedded message size. Finally, detection consistency relates to the feature's ability to provide consistently accurate detection for a large set of steganography techniques and image types. This implies that the feature should be independent on the type and variety of images supplied to it.

In [4] Avcibas et al. develop a discriminator for cover images and stego images, using an appropriate set of Image Quality Metrics (IQM's). Objective image quality measures have been utilized in coding artifact evaluation, performance prediction of vision algorithms, quality loss due to sensor inadequacy etc. In [4] they are used not as predictors of subjective image quality or algorithmic performance, but specifically as a steganalysis tool, that is, as features used in distinguishing cover-objects from stego-objects.

To select quality metrics to be used for steganalysis, the authors use Analysis of Variance (ANOVA) techniques. They arrive at a ranking of IQM's based on their F-scores in the ANOVA tests to identify the ones that responded most consistently and strongly to message embedding. The idea is to seek IQM's that are sensitive specifically to steganography effects, that is, those measures for which the variability in score data can be explained better because of some treatment rather than as random variations due to the image set. The rationale of using several quality measures is

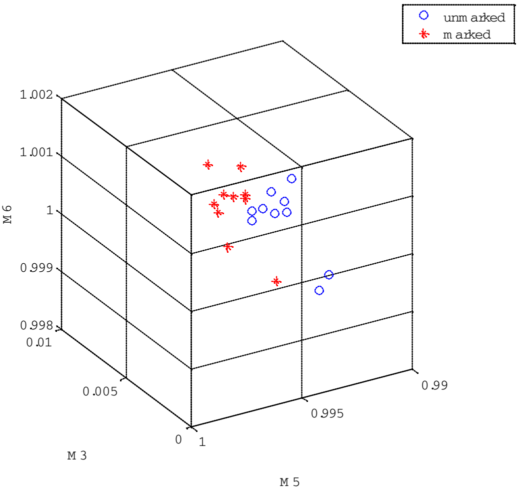


Fig. 10. Scatter plot of 3 image quality measures showing separation of marked and unmarked images. (Figure taken from [4].)

that different measures respond with differing sensitivities to artifacts and distortions. For example, measures like mean-square-error respond more to additive noise, whereas others such as spectral phase or mean square HVS-weighted (Human Visual System) error are more sensitive to pure blur; while the gradient measure reacts to distortions concentrated around edges and textures. Similarly embedding techniques affect different aspects of images. Fig. 10 shows separation in the feature plane between stego images and cover images, for 3 example quality metrics.

A second technique proposed by Avcibas et al. [2] looks at seventh and eight bit planes of an image and calculates several binary similarity measures. The approach is based on the fact that correlation between contiguous bit-planes is affected after a message is embedded in the image. The authors conjecture that correlation between the contiguous bit planes decreases after a message is embedded in the image. In order to capture the effect made by different embedding algorithms several features are calculated. Using the obtained features a MMSE linear predictor is obtained which is used to classify a given image as either a cover image or an image containing hidden messages.

A different approach is taken by Farid et al. [25,26] for feature extraction from images. The authors argue that most of the specific steganalysis techniques concentrate on first order statistics, i.e. histogram of DCT coefficients, but simple counter measures could keep the first order statistics intact thus making the steganalysis technique useless. So they propose building a model for natural images by using higher order statistics and then show that images with messages embedded in them deviate from this model. Quadratic mirror filters (QMF) are used to decompose the image, after which higher order statistics such as mean, variance, skewness, and kurtosis are calculated for each subband. Additionally the same statistics are calculated for the error obtained from an optimal linear predictor of coefficient magnitudes of each subband, as the second part of the feature set.

In all of the above methods, the calculated features are used to train a classifier, which in turn is used to classify clean and stego images. Different classifiers have been employed by different authors, Avcibas et al. use a MMSE Linear predictor, whereas Farid et al. [25,26] use a Fisher linear discriminant [15] and also a Support Vector Machine (SVM) [6] classifier. SVM classifiers seem to have much better performance in terms of classification accuracy compared to linear classifiers since they are able to classify non-linearly separable features. All of the above authors have reported good accuracy results in classifying images as clean or containing hidden messages after training with a classifier. Although, direct comparison might be hard as is in many classification problems, due to the fact that the way experiments are setup or conducted vary.

7. Conclusion

The past few years have seen an increasing interest in using images as cover media for steganographic communication. There have been a multitude of public domain tools, albeit many being ad-hoc and naive, available for image based steganography. Given this fact, detection of covert communications that utilize images has become an important issue. In this tutorial we have reviewed some fundamental notions related to steganography and steganalysis.

Although we covered a number of security and capacity definitions, there has been no work successfully formulating the relationship between the two from the practical point of view. For example it is understood that as less information is embedded in a cover-object the more secure the system will be. But due to difficulties in statistical modelling of image features, the security versus capacity trade-off has not been theoretically explored and quantified within an analytical framework.

We also reviewed a number of embedding algorithms starting with the earliest algorithm proposed which was the LSB technique. At some point LSB seemed to be unbreakable but as natural images were better understood and newer models were created LSB gave way to new and more powerful algorithms which try to minimize changes to image statistics. But with further improvement in understanding of the statistical regularities and redundancies of natural images, most of these algorithms have also been successfully steganalysed.

In term of steganalysis, as discussed earlier, there are two approaches, technique specific or universal steganalysis. Although finding attacks specific to an embedding method are helpful in coming up with better embedding methods, their practical usage seems to be limited. Since given an image we may not know the embedding technique being used, or even we might be unfamiliar with the embedding technique. Thus universal steganalysis techniques seem to be the real solution since they should be able to detect stego images even when a new embedding technique is being employed.

Acknowledgment

This work was supported by AFRL Grant No. F30602-03-C-0091.

References

1. F. Alturki and R. Mersereau, "Secure blind image steganographic technique using discrete fourier transformation," *IEEE International Conference on Image Processing, Thessaloniki, Greece.*, 2001.
2. I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics." *Security and Watermarking of Multimedia Contents, San Jose, Ca.*, Feruary 2001.
3. I. Avcibas, N. Memon, and B. Sankur, "Image steganalysis with binary similarity measures." *IEEE International Conference on Image Processing, Rochester, New York.*, September 2002.
4. I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics." *IEEE transactions on Image Processing*, January 2003.
5. R. J. Barron, B. Chen, and G. W. Wornell, "The duality between information embedding and source coding with side information and its implications - applications," *IEEE Transactions on Information Theory.*
6. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery.*, pp. 2:121–167, 1998.
7. C. Cachin, "An information-theoretic model for steganography," *2nd International Workshop Information Hiding*, vol. LNCS 1525, pp. 306–318, 1998.

8. R. Chandramouli, “Data hiding capacity in the presence of an imperfectly known channel,” *SPIE Proceedings of Security and Watermarking of Multimedia Contents II*, vol. 4314, pp. 517–522, 2001.
9. R. Chandramouli, “Watermarking capacity in the presence of multiple watermarks and partially known channel,” *SPIE Multimedia Systems and Applications IV*, vol. 4518, pp. 210–215, Aug. 2001.
10. R. Chandramouli and N. Memon, “Analysis of lsb image steganography techniques,” *IEEE International Conference on Image Processing*, vol. 3, pp. 1019–1022, 2001.
11. R. Chandramouli and N. Memon, “Steganography capacity: A steganalysis perspective,” *SPIE Security and Watermarking of Multimedia Contents V*, vol. 5020, 2003.
12. J. Chou, S. S. Pradhan, L. E. Ghaoui, and K. Ramchandran, “A robust optimization solution to the data hiding problem using distributed source coding principles,” *Proceedings SPIE: Image and Video Communications and Processing*, vol. 3974, 2000.
13. A. Cohen and A. Lapidoth, “On the gaussian watermarking game,” *International Symposium on Information Theory*, June 2000.
14. F. Collin, “Encryptpic,” *http://www.winsite.com/bin/Info?500000033023.*
15. R. Duda and P. Hart, “Pattern classification and scene analysis,” *John Wiley and Sons.*, 1973.
16. S. Dumitrescu, X. Wu, and N. Memon, “On steganalysis of random lsb embedding in continuous-tone images,” *IEEE International Conference on Image Processing, Rochester, New York*, September 2002.
17. J. Fridrich, R. Du, and L. Meng, “Steganalysis of lsb encoding in color images,” *IEEE International Conference on Multimedia and Expo*, *New York*, August 2000.
18. J. Fridrich and M. Goljan, “Digital image steganography using stochastic modulation,” *SPIE Symposium on Electronic Imaging, San Jose, CA*, 2003.
19. J. Fridrich, M. Goljan, and R. Du, “Detecting lsb steganography in color and gray-scale images,” *IEEE Multimedia Special Issue on Security*, pp. 22–28, October-November 2001.
20. J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, “Quantitive steganalysis of digital images: Estimating the secret message lenght,” *ACM Multimedia Systems Journal, Special issue on Multimedia Security*, 2003.
21. N. J. Hopper, J. Langford, and L. von Ahn, “Provably secure steganography,” *Advances in Cryptology: CRYPTO 2002, August*, 2002.
22. N. F. Johnson and S. Jajodia, “Steganalysis of images created using current steganography software,” *in David Aucsmith (Eds.): Information Hiding, LNCS 1525, Springer-Verlag Berlin Heidelberg*, pp. 32–47, 1998.
23. S. Katzenbeisser and F. A. P. Petitcolas, “Defining security in steganographic systems,” *Proceedings of the SPIE vol. 4675, Security and Watermarking of Multimedia Contents IV.*, pp. 50–56, 2002.
24. M. Kwan, “Gifshuffle,” *http://www.darkside.com.au/gifshuffle/.*
25. S. Lyu and H. Farid, “Detecting hidden messages using higher-order statistics

and support vector machines," *5th International Workshop on Information Hiding*, 2002.

26. S. Lyu and H. Farid, "Steganalysis using color wavelet statistics and one-class support vector machines," *SPIE Symposium on Electronic Imaging, San Jose, CA*, 2004.
27. R. Machado, "Ezstego," *http://www.stego.com*, 2001.
28. C. Moroney, "Hide and seek," *http://www.rugeley.demon.co.uk/security/hdsk50.zip.*
29. P. Moulin and M. Mihcak, "A framework for evaluating the data-hiding capacity of image sources," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1029–1042, September 2002.
30. P. Moulin and J. Sullivan, "Information theoretic analysis of information hiding," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 563–593, 2003.
31. P. Moulin and Y. Wang, "New results on steganography," *Proc. of CISS*, 2004.
32. N. Provos, "Defending against statistical steganalysis," *10th USENIX Security Symposium*, 2001.
33. G. Pulcini, "Stegotif," *http://www.geocities.com/SiliconValley/9210/gfree.html.*
34. R. Radhakrishnan, M. Kharrazi, and N. Memon, "Data masking: A new approach for steganography?" *The Journal of VLSI Signal Processing*, vol. 41, no. 3, November 2005.
35. R. Radhakrishnan, K. Shanmugasundaram, and N. Memon, "Data masking: A secure-covert channel paradigm," *IEEE Multimedia Signal Processing, St. Thomas, US Virgin Islands*, 2002.
36. M. Ramkumar and A. Akansu, "Information theoretic bounds for data hiding in compressed images," *IEEE 2nd Workshop on Multimedia Signal Processing*, pp. 267–272, Dec. 1998.
37. M. Ramkumar and A. Akansu, "Theoretical capacity measures for data hiding in compressed images," *SPIE Multimedia Systems and Application*, vol. 3528, pp. 482–492, 1998.
38. L. Reyzin and S. Russell, "More efficient provably secure steganography," 2003.
39. P. Sallee, "Model-based steganography," *International Workshop on Digital Watermarking, Seoul, Korea.*, 2003.
40. T. Sharp, "Hide 2.1, 2001," *http://www.sharpthoughts.org.*
41. G. Simmons, "The prisoners problem and the subliminal channel," *CRYPTO*, pp. 51–67, 1983.
42. E. P. Simoncelli, "Modeling the joint statistics of images in the wavelet domain," *Proceedings of the 44th Annual Meeting*, 1999.
43. D. Upham, "Jpeg-jsteg," *ftp://ftp.funet.fi/pub/crypt/steganography/jpeg-jsteg-v4.diff.gz.*
44. Y. Wang and P. Moulin, "Steganalysis of block-dct image steganography," *IEEE Workshop On Statistical Signal Processing*, 2003.

45. Y. Wang and P. Moulin, "Steganalysis of block-structured text," *Proceedings of SPIE*, 2004.
46. A. Westfeld, "F5a steganographic algorithm: High capacity despite better steganalysis," *4th International Workshop on Information Hiding*, 2001.
47. A. Westfeld, "Detecting low embedding rates," *5th International Workshop on Information Hiding*, pp. 324–339, 2002.
48. A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," *3rd International Workshop on Information Hiding*, 1999.
49. R. Zamir, S. Shamai, and U. Erez, "Nested lattice/linear for structured multiterminal binning," *IEEE Transactions on Information Theory*, 2002.
50. J. Zollner, H. Federrath, H. Klimant, A. Pfitzman, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf, "Modeling the security of steganographic systems," *2nd Information Hiding Workshop*, pp. 345–355, April 1998.

This page intentionally left blank

THE APRIORI ALGORITHM – A TUTORIAL

Markus Hegland
CMA, Australian National University
John Dedman Building, Canberra ACT 0200, Australia
E-mail: Markus.Hegland@anu.edu.au

Association rules are "if-then rules" with two measures which quantify the support and confidence of the rule for a given data set. Having their origin in market basket analysis, association rules are now one of the most popular tools in data mining. This popularity is to a large part due to the availability of efficient algorithms. The first and arguably most influential algorithm for efficient association rule discovery is Apriori.

In the following we will review basic concepts of association rule discovery including support, confidence, the apriori property, constraints and parallel algorithms. The core consists of a review of the most important algorithms for association rule discovery. Some familiarity with concepts like predicates, probability, expectation and random variables is assumed.

1. Introduction

Large amounts of data have been collected routinely in the course of day-to-day management in business, administration, banking, the delivery of social and health services, environmental protection, security and in politics. Such data is primarily used for accounting and for management of the customer base. Typically, management data sets are very large and constantly growing and contain a large number of complex features. While these data sets reflect properties of the managed subjects and relations, and are thus potentially of some use to their owner, they often have relatively low information density. One requires robust, simple and computationally efficient tools to extract information from such data sets. The development and understanding of such tools is the core business of data mining. These tools are based on ideas from computer science, mathematics and statistics.

The introduction of association rule mining in 1993 by Agrawal, Imielinski and Swami [2] and, in particular, the development of an efficient algorithm by Agrawal and Srikant [3] and by Mannila, Toivonen and Verkamo [13] marked a shift of the focus in the young discipline of data mining onto rules and data bases. Consequently, besides involving the traditional statistical and machine learning community, data mining now attracted researchers with a variety of skills ranging from computer science, mathematics, science, to business and administration. The urgent need for computational tools to extract information from data bases and for manpower to apply these tools has allowed a diverse community to settle in this new area. The data analysis aspect of data mining is more exploratory than in statistics and consequently, the mathematical roots of probability are somewhat less prominent in data mining than in statistics. Computationally, however, data mining frequently requires the solution of large and complex search and optimisation problems and it is here where mathematical methods can assist most. This is particularly the case for association rule mining which requires searching large data bases for complex rules. Mathematical *modelling* is required in order to generalise the original techniques used in market basket analysis to a wide variety of applications. Mathematical *analysis* provides insights into the performance of the algorithms.

An association rule is an *implication* or *if-then-rule* which is supported by data. The motivation given in [2] for the development of association rules is *market basket analysis* which deals with the contents of point-of-sale transactions of large retailers. A typical association rule resulting from such a study could be “90 percent of all customers who buy bread and butter also buy milk”. Insights into customer behaviour may also be obtained through customer surveys, but the analysis of the transactional data has the advantage of being much cheaper and covering all current customers. Compared to customer surveys, the analysis of transactional data does have some severe limitations, however. For example, point-of-sale data typically does not contain any information about personal interests, age and occupation of customers. Nonetheless, market basket analysis can provide new insights into customer behaviour and has led to higher profits through better customer relations, customer retention, better product placements, product development and fraud detection.

Market basket analysis is not limited to retail shopping but has also been applied in other business areas including

- credit card transactions,
- telecommunication service purchases,
- banking services,
- insurance claims, and
- medical patient histories.

Association rule mining generalises market basket analysis and is used in many other areas including genomics, text data analysis and Internet intrusion detection. For motivation we will in the following mostly focus on retail market basket analysis.

When a customer passes through a point of sale, the contents of his market basket are registered. This results in large collections of market basket data which provide information about which items were sold and, in particular, which combinations of items were sold. The small toy example in the table of figure 1 shall illustrate this further. Each row corresponds to a market basket or transaction containing popular retail items. An inspection of the table reveals that:

market basket id	market basket content
1	orange juice, soda water
2	milk, orange juice, bread
3	orange juice, butter
4	orange juice, bread, soda water
5	bread

Fig. 1. Five grocery market baskets.

- Four of the five baskets contain orange juice,
- two baskets contain soda water
- half of the baskets which contain orange juice also contain soda
- all the baskets which contain soda also contain juice

These rules are very simple as is typical for association rule mining. Simple rules are understandable and ultimately useful. In a large retail shop there are usually more than 10,000 items on sale and the shop may service thousands of customers every day. Thus the size of the collected data is substantial and even the detection of simple rules like the ones above requires sophisticated algorithms. The efficiency of the algorithms will depend on the particular characteristics of the data sets. An important feature of many retail data sets is that an average market basket only contains a small subset of all items available.

The simplest model for the customers assumes that the customers choose products from the shelves in the shop at random. In this case the choice of each product is independent from any other product. Consequently, association rule discovery will simply recover the likelihoods for any item to be chosen. While it is important to compare the performance of other models with this "null-hypothesis" one would usually find that shoppers do have a more complex approach when they fill the shopping basket (or trolley). They will buy breakfast items, lunch items, dinner items and snacks, party drinks, and Sunday dinners. They will have preferences for cheap items, for (particular) brand items, for high-quality, for freshness, low-fat, special diet and environmentally safe items. Such goals and preferences of the shopper will influence the choices but can not be directly observed. In some sense, market basket analysis should provide information about how the shoppers choose. In order to understand this a bit further consider the case of politicians who vote according to party policy but where we will assume for the moment that the party membership is unknown. Is it possible to see an effect of the party membership in voting data? For a small but real illustrative example consider the US Congress voting records from 1984 [12], see figure 2. The 16 columns of the displayed bit matrix correspond to the 16 votes and the 435 rows to the members of congress. We have simplified the data slightly so that a matrix element is one (pixel set) in the case of votes which contains "voted for", "paired for" and "announced for" and the matrix element is zero in all other cases. The left data matrix in figure 2 is the original data where only the rows and columns have been randomly permuted to remove any information introduced through the way the data was collected. The matrix on the right side is purely random such that each entry is independent and only the total number of entries is maintained. Can you see the difference between the two bit matrices? We found that for most people, the difference between the two matrices is not obvious from visual inspection alone.

Data mining aims to discover patterns in the left bit matrix and thus differences between the two examples. In particular, we will find columns or items which display similar voting patterns and we aim to discover rules relating to the items which hold for a large proportion of members of congress. We will see how many of these rules can be explained by underlying mechanisms (in this case party membership).

In this example the selection of what are rows and what columns is

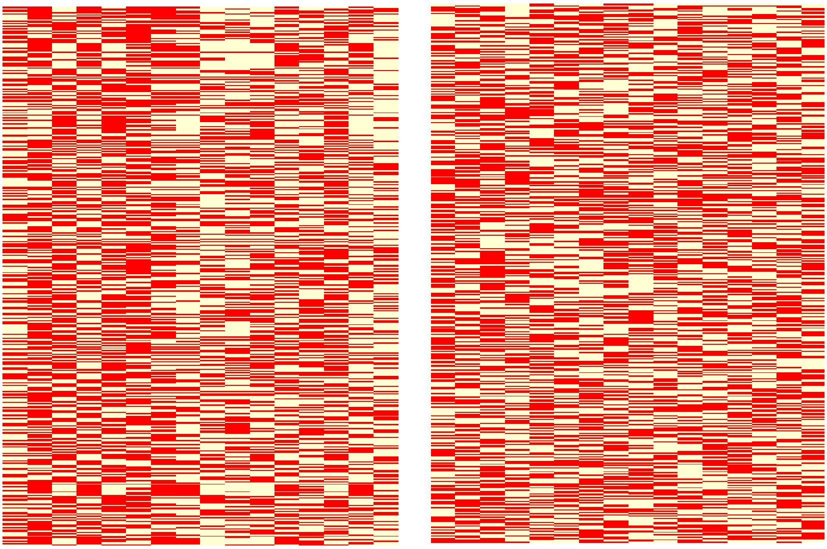


Fig. 2. 1984 US House of Representatives Votes, with 16 items voted on.

somewhat arbitrary. Instead of patterns regarding the items voted on one might be interested in patterns relating the members of Congress. For example one might be interested in statements like “if member x and member y vote yes then member z votes yes as well. Statements like this may reveal some of the interactions between the members of Congress. The duality of observations and objects occurs in other areas of data mining as well and illustrates that data size and data complexity are really two dual concepts which can be interchanged in many cases. This is in particular exploited in some newer association rule discovery algorithms which are based on formal concept analysis [8].

By inspecting the data matrix of the voting example, one finds that the items have received between 34 to 63.5 percent yes votes. Pairs of items have received between 4 and 49 percent yes votes. The pairs with the most yes votes (over 45 percent) are in the columns 2/6, 4/6, 13/15, 13/16 and 15/16. Some rules obtained for these pairs are: 92 percent of the yes votes in column 2 are also yes votes in column 6, 86 percent of the yes votes in column 4 are also yes votes in column 6 and, on the other side, 88 percent of the votes in column 13 are also in column 15 and 89 percent of the yes votes in column 16 are also yes votes in column 15. These figures suggest combinations of items which could be further investigated in terms of causal relationships between the items. Only a careful statistical analysis may provide some certainty on this. This and other issues concerning inference belong to statistics and are beyond the scope of this tutorial which focusses on computational issues.

2. Itemsets and Associations

In this section a formal mathematical model is derived to describe itemsets and associations to provide a framework for the discussion of the apriori algorithm. In order to apply ideas from market basket analysis to other areas one needs a general description of market baskets which can equally describe collections of medical services received by a patient during an episode of care, subsequences of amino acid sequences of a protein, and collections or words or concepts used on web pages. In this general description the items are numbered and a market basket is represented by an indicator vector.

2.1. *The datamodel*

In this subsection a probabilistic model for the data is given along with some simple model examples. For this, we consider the voting data example again.

First, the items are enumerated as $0, \ldots, d-1$. Often, enumeration is done such that the more frequent items correspond to lower numbers but this is not essential. Itemsets are then sets of integers between 0 and $d-1$. The itemsets are represented by bitvectors $x \in \mathbb{X} := \{0,1\}^d$ where item j is present in the corresponding itemset iff the j-th bit is set in x. Consider the "micromarket" with the items juice, bread, milk, cheese and potatoes with item numbers $0, 1, 2, 3$ and 4, respectively. The market basket containing bread, milk and potatoes is then mapped onto the set $\{1,2,4\}$ and is represented by the bitvector $(0,1,1,0,1)$. From the bitvector it is clear which elements are in the market basket or itemset and which are not.

The data is a sequence of itemsets which is represented as a bitmatrix where each row corresponds to an itemset and the columns correspond to the items. For the micromarket example a dataset containing the market baskets {juice,bread, milk}, {potato} and {bread, potatoes} would be represented by the matrix

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

In the congressional voting example mentioned in the previous section the first few rows of matrix are

```
1  0 1 1 1 1  0 1  0 1 0  0  1  1 0 0
1  1 1 0 0 1  0 1  1 1 1  0  1  1 0 0
0  0 0 1 1 1  0 1  1 0 0  0  0  0 1 0
0  1 0 1 0 1  0 1  1 0 1  1  0  0 1 0
```

and they correspond to the following "itemsets" (or sets of yes-votes):

```
1  3  4  5  6  8  10  13  14
1  2  3  6  8  9  10  11  13  14
4  5  6  8  9  15
2  4  6  8  9  11 12  15
1  3  7  9 10  11 13  16.
```

It is assumed that the data matrix $X \in \{0,1\}^{n,d}$ is random and thus the elements $x_j^{(i)}$ are binary random variables. One would in general have to assume correlations between both rows and columns. The correlations between the columns might relate to the type of shopping and customer, e.g., young family with small kids, weekend shopping or shopping for a specific dish. Correlations between the rows might relate to special offers of the retailer, time of the day and week. In the following it will be assumed that the rows are drawn independently from a population of market baskets. Thus it is assumed that there is a probability distribution function $p : \mathbb{X} \rightarrow [0,1]$ with

$$\sum_{x \in \mathbb{X}} p(x) = 1$$

where $\mathbb{X} = \{0,1\}^d$. The probability measure with distribution p is denoted by P and one has $P(A) = \sum_{x \in A} p(x)$.

The data can be represented as an empirical distribution with

$$p_{\mathrm{emp}}(x) = \frac{1}{n} \sum_{i=1}^{n} \delta(x - x^{(i)})$$

where $\delta(x)$ is the indicator function where $\delta(0) = 1$ and $\delta(x) = 0$ if $x \neq 0$. (For simplicity the empty market basket is denoted by 0 instead of $(0, \ldots, 0)$.) All the information derived from the data is stored in p_{emp} but some sort of "smoothing" is required if one would like to generalise insights from the empirical distribution of one itemset collection to another to separate the noise from the signal. Association rule discovery has its own form of smoothing as will be seen in the following.

The task of *frequent itemset mining* consists of finding itemsets which occur frequently in market baskets. For this one recalls that the itemsets are partially ordered with respect to inclusion (the subset relation) and we write $x \leq y$ if the set with representation x is a subset of the set with representation y or $x = y$. With this partial order one defines the *support* of an itemset x to be

$$s(x) = P(\{z \mid x \leq z\}) \tag{1}$$

which is also called the *anticummulative distribution function* of the probability P. The support is a function $s : \mathbb{X} \to [0,1]$ and $s(0) = 1$. By construction, the support is *antimonotone*, i.e., if $x \leq y$ then $p(x) \geq p(y)$. This antimonotonicity is the basis for efficient algorithms to find all *frequent itemsets* which are defined as itemsets for which $s(x) \geq \sigma_0 > 0$ for some user defined σ_0.

Equation (1) can be reformulated in terms of $p(x)$ as

$$s(x) = \sum_{z \geq x} p(z). \tag{2}$$

For given supports $s(x)$, this is a linear system of equations which can be solved recursively using $s(e) = p(e)$ (where $e = (1, \ldots, 1)$ is the maximal itemset) and

$$p(x) = s(x) - \sum_{z > x} p(z).$$

It follows that the support function $s(x)$ provides an alternative description of the probability measure P which is equivalent to p. However, for many examples the form of $s(x)$ turns out to be simpler. In the cases of market baskets it is highly unlikely, that market baskets contain large numbers of items and so approximations with $s(x) = 0$ for x with a large number of nonzero components will usually produce good approximations of the itemset distribution $p(x)$. This leads to an effective smoothing mechanism for association rule discovery where the minimal support σ_0 acts as a smoothing parameter which in principle could be determined from a test data set or with crossvalidation.

2.1.1. *Example: The random shopper*

In the simplest case all the bits (items) in x are chosen independently with probability p_0. We call this the case of the "random shopper" as it corresponds to a shopper who fills the market basket with random items. The distribution is in this case

$$p(x) = p_0^{|x|}(1 - p_0)^{d-|x|}$$

where $|x|$ is the number of bits set in x and d is the total number of items available, i.e., number of components of x. As any $z \geq x$ has at least all the bits set which are set in x one gets for the support

$$s(x) = p_0^{|x|}.$$

It follows that the frequent itemsets x are exactly the ones with few items, in particular, where

$$|x| \leq \log(\sigma_0)/\log(p_0).$$

For example, if one is interested in finding itemsets which are supported by one percent of the data records and if the probability of choosing any item is $p_0 = 0.1$ the frequent itemsets are the ones with at most two items. For large shops one would typically have p_0 much smaller. Note that if $p_0 < \sigma_0$ one would not get any frequent itemsets at all.

The random shopper is of course not a realistic model for shopping and one would in particular not expect to draw useful conclusions from the frequent itemsets. Basically, the random shopper is the market basket equivalent of noise. The above discussion, however, might be used to guide the choice of σ_0 to filter out the noise in market basket analysis. In particular, one could choose

$$\sigma_0 = \min_{|x|=1} s(x).$$

Note that in the random shopper case the rhs is just p_0. In this case, all the single items would be frequent.

A slight generalisation of the random shopper example above assumes that the items are selected independently but with different probabilities p_j. In this case one gets

$$p(x) = \prod_{j=1}^{d} p_j^{x_j}(1 - p_j)^{1-x_j}$$

and

$$s(x) = \prod_{j=1}^{d} p_j^{x_j}.$$

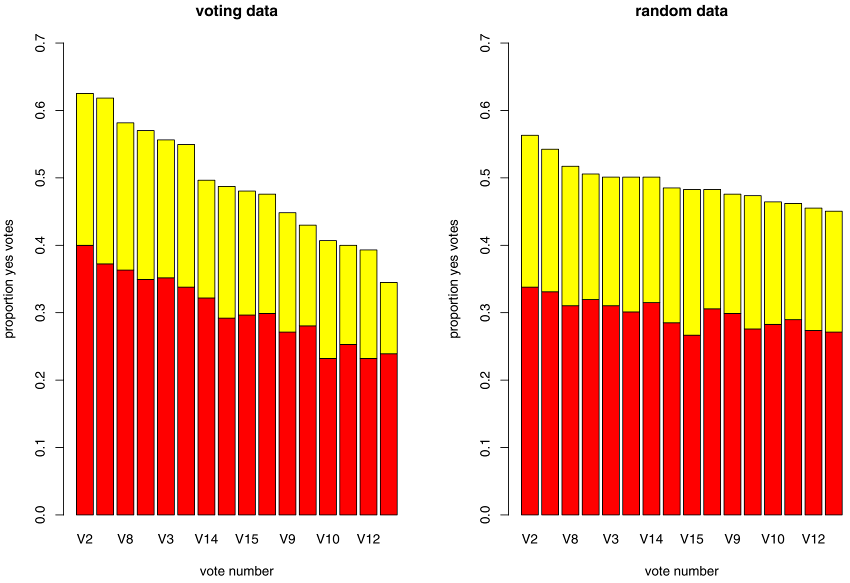


Fig. 3. Supports for all the votes in the US congress data (split by party).

In examples one can often find that the p_j, when sorted, are approximated by *Zipf's law*, i.e,

$$p_j = \frac{\alpha}{j}$$

for some constant α. It follows again that itemsets with few popular items are most likely.

However, this type of structure is not really what is of interest in association rule mining. To illustrate this consider again the case of the US Congress voting data. In figure 3 the support for single itemsets are displayed for the case of the actual data matrix and for a random permutation of all the matrix elements. The supports are between 0.32 and 0.62 for the original data where for the randomly permuted case the supports are between 0.46 and 0.56. Note that these supports are computed from the data, in theory, the permuted case should have constant supports somewhere slightly below 0.5. More interesting than the variation of the supports of single items is the case when 2 items are considered. Supports for the votes displayed in figure 4. are of the form "V2 and Vx". Note that "V2 and V2" is included for reference, even though this itemset has only one item. In the random data case where the vote for any pair {V2,Vx} (where Vx

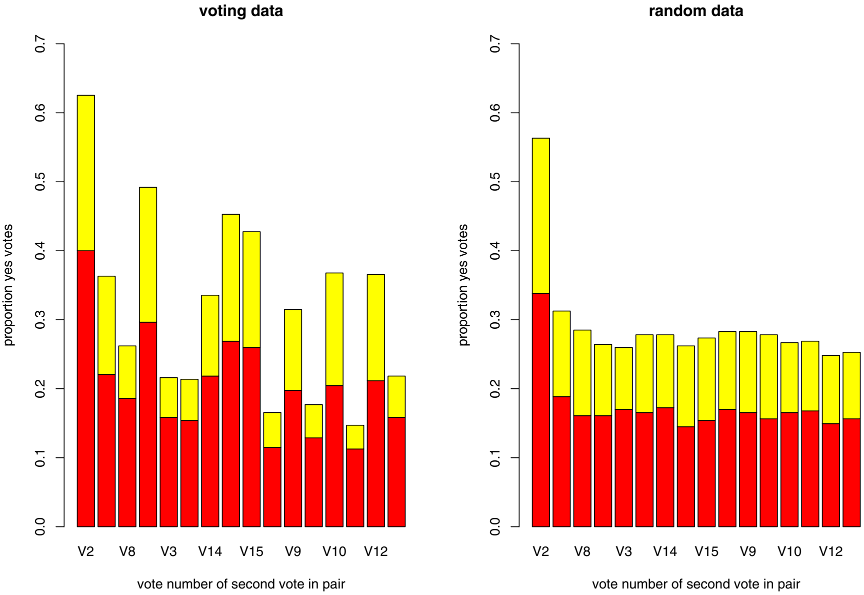


Fig. 4. Supports for pairs of votes in the US congress data (split by party).

is not V2) is the square of the vote for the single item V2 as predicted by the "random shopper theory" above. One notes that some pairs have significantly higher supports than the random ones and others significantly lower supports. This type of behaviour is not captured by the "random shopper" model above even if the case of variable supports for single items are allowed. The following example attempts to model some this behaviour.

2.1.2. *Example: Multiple shopper and item classes*

Consider now the case of some simple structure. Assume that there are two types of shoppers and two types of items. Assume that the items are $x = (x_0, x_1)$ where the vectors x_i correspond to items of class i. In practice, the type of items might have to be discovered as well. Consider that the shoppers of type i have a probability π_i of filling a market basket where here $i = 0, 1$. Assume that it is not known to which type of shopper a market basket belongs. Finally, assume that the difference between the two types of shoppers relates to how likely they are to put items of the two types into their market basket. Let $p_{i,j}$ denote the probability that shopper of type i puts an item of type j into the market basket. In this case the probability distribution is

$$p(x) = \pi_0 \, p_{00}^{|x_0|} p_{01}^{|x_1|} (1 - p_{00})^{d_0 - |x_0|} (1 - p_{01})^{d_1 - |x_1|}$$
$$+ \pi_1 \, p_{10}^{|x_0|} p_{11}^{|x_1|} (1 - p_{10})^{d_0 - |x_0|} (1 - p_{11})^{d_1 - |x_1|}.$$

This is a *mixture model* with two components. Recovery of the parameters from the data of mixture models uses the EM algorithm and is discussed in detail in [14]. Note that $\pi_0 + \pi_1 = 1$.

For frequent itemset mining, however, the support function is considered and similarly to the random shopper case can be shown to be:

$$s(x) = \pi_0 p_{00}^{|x_0|} p_{01}^{|x_1|} + \pi_1 p_{10}^{|x_0|} p_{11}^{|x_1|}.$$

Assume that the shopper of type i is unlikely to purchase items of the other type. Thus p_{00} and p_{11} are much larger than p_{10} and p_{01}. In this case the frequent itemsets are going to be small (as before), moreover, one has either $x_0 = 0$ or $x_1 = 0$, thus, frequent itemsets will only contain items of one type. Thus in this case frequent itemset mining acts as a filter to retrieve "pure" itemsets.

A simple application to the voting data could consider two types of politicians. A question to further study would be how closely these two types correspond with the party lines.

One can now consider generalisations of this case by including more than two types, combining with different probabilities (Zipf's law) for the different items in the same class and even use itemtypes and customer types which overlap. These generalisations lead to graphical models and Bayesian nets [11,5]. The "association rule approach" in these cases distinguishes itself by using support functions, frequent itemsets and in particular, is based on binary data. A statistical approach to this type of data is "discriminant analysis" [7].

2.2. *The size of itemsets*

The size of the itemsets is a key factor in determining the performance of association rule discovery algorithms. The size of an itemset is the equal to the number of bits set, one has

$$|x| = \sum_{i=1} x_i$$

if the components of x are interpreted as integers 0 or 1 this is a real valued random variable or function defined on $\mathbb{X}$. The expectation of a general real random variable f is

$$E(f) = \sum_{x \in \mathbb{X}} p(x) f(x).$$

The expectation is monotone in the sense that $f \geq g \Rightarrow E(f) \geq E(g)$ and the expectation of a constant function is the function value. The *variance* of the random variable corresponding to the function f is

$$\mathrm{var}(f) = E\left((f - E(f))^2\right).$$

For an arbitrary but fixed itemset $x \in \mathbb{X}$ consider the function

$$a_x(z) = \prod_{i=1}^{d} z_i^{x_i}.$$

Thus function takes values which are either 0 or 1 and $a_x(z) = 1$ iff $x \leq z$ as in this case all the components z_i which occur to power 1 in $a_x(z)$ are one. Note that $a_x(z)$ is a monotone function of z and antimonotone function of x. Moreover, one has for the expectation $E(a_x) = s(x)$ and variance $\mathrm{var}(a_x) = s(x)(1 - s(x))$. The term $1 - s(x)$ is the support of the complement of the itemset x. The values for the expectation and variance are obtained directly from the definition of the expectation and the fact that $a_x(z)^2 = a_x(z)$ (holds for any function with values 0 and 1).

Our main example of a random variable is the length of an itemset,

$$f(x) = |x| = \sum_{j=1}^{d} x_j.$$

The average length of itemsets is the expectation of f and one can see that

$$E(|x|) = \sum_{|z|=1} s(z).$$

The variance of the length is also expressed in terms of the support as

$$\mathrm{var}(|x|) = \sum_{|x||z|=1} (s(x \vee z) - s(x)s(z))$$

where $x \vee z$ corresponds to the union of the itemsets x and z.

With the expectation and using the Markov inequality one gets a simple bound on the probability of large itemsets as

$$P(\{x \mid |x| \geq m\}) \leq E(|x|)/m.$$

The expected length is easily obtained directly from the data and this bound gives an easy upper bound on probability of large itemsets. Of course one could just as easily get a histogram for the size of itemsets directly from the data. Using the above equation one gets an estimate for the average support of one-itemsets as $E(|x|)/d$.

Consider now the special example of a random shopper discussed previously. In this case one gets $E(|x|) = dp_0$. The distribution of the length is in this case binomial and one has:

$$P(|x| = r) = \binom{d}{r} p_0^r \, (1 - p_0)^{d-r}.$$

Moreover, for very large d and small p_0 one gets a good approximation using the *Poisson distribution*

$$p(|x| = r) \approx \frac{1}{r!} e^{-\lambda} \lambda^r.$$

with $\lambda = E(|x|)$.

The apriori algorithm which will be discussed in the following works best when long itemsets are unlikely. Thus in order to choose a suitable algorithm, it is important to check if this is the case, e.g., by using the histogram for the length $|x|$.

2.3. *The itemset lattice*

The search for frequent itemsets benefits from a structured search space as the itemsets form a lattice. This lattice is also intuitive and itemsets close to 0 in the lattice are often the ones which are of most interest and lend themselves to interpretation and further discussion.

As $\mathbb{X}$ consists of sets or bitvectors, one has a natural *partial ordering* which is induced by the subset relation. In terms of the bitvectors one can define this component-wise as

$$x \leq y :\Leftrightarrow x_i \leq y_i.$$

Alternatively,

$$x \leq y :\Leftrightarrow (x_i = 1 \Rightarrow y_i = 1, \quad i = 1, \ldots, d)\,.$$

If $x_i = 1$ and $x \leq y$ then it follows that $y_i = 1$. Thus if the corresponding itemset to x contains item i then the itemset corresponding to y has to contain the same item. In other words, the itemset corresponding to x is a subset of the one corresponding to y.

Subsets have at most the same number of elements as their supersets and so if $x \leq y$ then $|x| \leq |y|$.

Bitvectors also allow a total order by interpreting it as an integer $\phi(x)$ by

$$\phi(x) = \sum_{i=0}^{d-1} x_i 2^i.$$

Now as ϕ is a bijection it induces a total order on $\mathbb{X}$ defined as $x \prec y$ iff $\phi(x) < \phi(y)$. This is the *colex order* and the colex order extends the partial order as $x \leq y \Rightarrow x \prec y$.

The partial order has a smallest element which consists of the empty set, corresponding to the bitvector $x = (0, \ldots, 0)$ and a largest element which is just the set of all items Z_d, corresponding to the bitvector $(1, \ldots, 1)$. Furthermore, for each pair $x, y \in \mathbb{X}$ there is a greatest lower bound and a least upper bound. These are just

$$x \vee y = z$$

where $z_i = \max\{x_i, y_i\}$ for $i = 0, \ldots, d-1$ and similarly for $x \wedge y$. Consequently, the partially ordered set $\mathbb{X}$ forms a Boolean lattice. We denote the maximal and minimal elements of $\mathbb{X}$ by e and 0.

In general, partial order is defined by

Definition 1: A partially ordered set $(\mathbb{X}, \leq)$ consists of a set $\mathbb{X}$ with a binary relation $\leq$ such that for all $x, x', x'' \in \mathbb{X}$:

- $x \leq x$ (reflexivity)
- If $x \leq x'$ and $x' \leq x$ then $x = x'$ (antisymmetry)
- If $x \leq x'$ and $x' \leq x\prime\prime$ then $x \leq x''$ (transitivity)

A lattice is a partially ordered set with glb and lub:

Definition 2: A lattice $(\mathbb{X}, \leq)$ is a partially ordered set such that for each pair of elements of $\mathbb{X}$ there is greatest lower bound and a least upper bound.

We will call a lattice *distributive* if the distributive law holds:

$$x \wedge (x' \vee x'') = (x \wedge x') \vee (x \wedge x'').$$

where $x \vee y$ is the maximum of the two elements which contains ones whenever at least one of the two elements and $x \wedge x'$ contains ones where both elements contain a one. Then we can define:

Definition 3: A lattice $(\mathbb{X}, \leq)$ is a *Boolean lattice* if

(1) $(\mathbb{X}, \leq)$ is distributive

(2) It has a *maximal element* e and a *minimal element* 0 such that for all $x \in \mathbb{X}$:

$$0 \leq x \leq e.$$

(3) Each element x as a (unique) *complement* x' such that $x \wedge x' = 0$ and $x \vee x' = e$.

The maximal and minimal elements 0 and e satisfy $0 \vee x = x$ and $e \wedge x = x$. In algebra one considers the properties of the conjunctives $\vee$ and $\wedge$ and a set which has conjunctives which have the properties of a Boolean lattice is called a *Boolean algebra.* We will now consider some of the properties of Boolean algebras.

The smallest nontrivial elements of $\mathbb{X}$ are the *atoms*:

Definition 4: The set of atoms $\mathcal{A}$ of a lattice is defined by

$$\mathcal{A} = \{x \in \mathbb{X} | x \neq 0 \text{ and if } \ x' \leq x \text{ then } x' = x\}.$$

The atoms generate the lattice, in particular, one has:

Lemma 5: *Let $\mathbb{X}$ be a finite Boolean lattice. Then, for each $x \in \mathbb{X}$ one has*

$$x = \bigvee \{z \in \mathcal{A}(\mathbb{X}) \mid z \leq x\}.$$

Proof: Let $A_x := \{z \in \mathcal{A}(B) | z \leq x\}$. Thus x is an upper bound for A_x, i.e., $\bigvee A_x \leq x$.

Now let y be any upper bound for A_x, i.e., $\bigvee A_x \leq y$. We need to show that $x \leq y$.

Consider $x \wedge y'$. If this is 0 then from distributivity one gets $x = (x \wedge y) \vee (x \wedge y') = x \wedge y \leq y$. Conversely, if it is not true that $x \leq y$ then $x \wedge y' > 0$. This happens if what we would like to show doesn't hold.

In this case there is an atom $z \leq x \wedge y'$ and it follows that $z \in A_x$. As y is an upper bound we have $y \geq z$ and so $0 = y \wedge y' \geq x \wedge y'$ which is impossible as we assumed $x \wedge y' > 0$. Thus it follows that $x \leq y$. □

The set of atoms associated with any element is unique, and the Boolean lattice itself is isomorph to the powerset of the set of atoms. This is the key structural theorem of Boolean lattices and is the reason why we can talk about sets (itemsets) in general for association rule discovery.

Theorem 6: A finite Boolean algebra $\mathbb{X}$ is isomorphic to the power set $2^{\mathcal{A}(\mathbb{X})}$ of the set of atoms. The isomorphism is given by

$$\eta : x \in \mathbb{X} \mapsto \{z \in \mathcal{A}(\mathbb{X}) \mid z \leq x\}$$

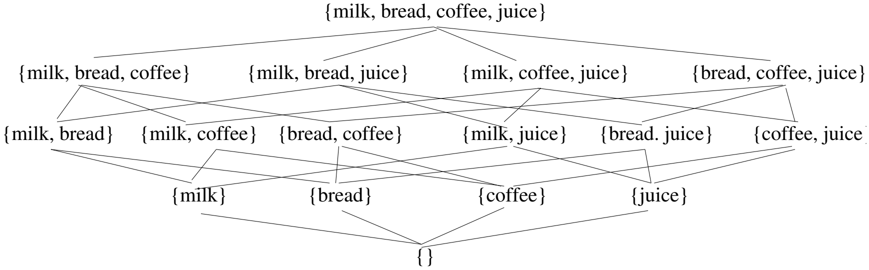


Fig. 5. Lattice of breakfast itemsets.

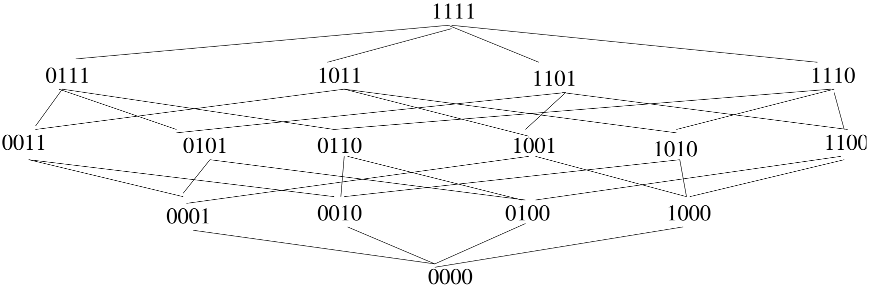


Fig. 6. Lattice of bitvectors.

and the inverse is

$$\eta^{-1} : S \mapsto \bigvee S.$$

In our case the atoms are the d basis vectors $e_1, \ldots, e_d$ and any element of $\mathbb{X}$ can be represented as a set of basis vectors, in particular $x = \sum_{i=1}^{d} \xi_i e_i$ where $\xi_i \in \{0, 1\}$. For the proof of the above theorem and further information on lattices and partially ordered sets see [6]. The significance of the theorem lays in the fact that if $\mathbb{X}$ is an arbitrary Boolean lattice it is equivalent to the powerset of atoms (which can be represented by bitvectors) and so one can find association rules on any Boolean lattice which conceptually generalises the association rule algorithms.

In figure 5 we show the lattice of patterns for a simple market basket case which is just a power set. The corresponding lattice for the bitvectors is in figure 6. We represent the lattice using an undirected graph where the nodes are the elements of $\mathbb{X}$ and edges are introduced between any element and its *covering* elements. A covering element of $x \in \mathbb{X}$ is an $x' \geq x$ such

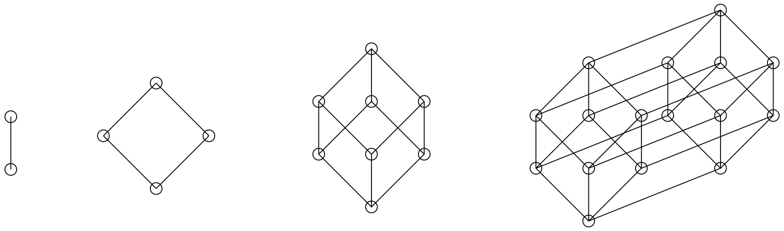


Fig. 7. The first Boolean lattices.

that no element is "in between" x and x', i.e., any element x'' with $x'' \geq x$ and $x' \geq x''$ is either equal to x or to x'.

In Figure 7 we display the graphs of the first few Boolean lattices. We will graph specific lattices with (Hasse) diagrams [6] and later use the positive plane $\mathbb{R}^2_+$ to illustrate general aspects of the lattices.

In the following we may sometimes also refer to the elements x of $\mathbb{X}$ as item sets, market baskets or even patterns depending on the context. As the data set is a finite collection of elements of a lattice the closure of this collection with respect to $\wedge$ and $\vee$ (the inf and sup operators) in the lattice forms again a boolean lattice. The powerset of the set of elements of this lattice is then the sigma algebra which is fundamental to the measure which is defined by the data.

The partial order in the set $\mathbb{X}$ allows us to introduce the *cumulative distribution function* as the probability that we observe a bitvector less than a given $x \in \mathbb{X}$:

$$F(x) = P\left(\{x' | x' \leq x\}\right).$$

By definition, the cumulative distribution function is monotone, i.e.

$$x \leq x' \;\Rightarrow\; F(x) \leq F(x').$$

A second cumulative distribution function is obtained from the dual order as:

$$F^{\partial}(x) = P\left(\{x' | x' \leq x\}\right).$$

It turns out that this dual cumulative distribution function is the one which is more useful in the discussion of association rules and frequent itemsets as one has for the support $s(x)$:

$$s(x) = F^{\partial}(x).$$

From this it follows directly that $s(x)$ is antimonotone, i.e., that

$$x \leq y \Rightarrow s(x) \geq s(y).$$

The aim of frequent itemset mining is to find sets of itemsets, i.e., subsets of $\mathbb{X}$. In particular, one aims to determine

$$L = \{x \mid s(x) \geq \sigma_0\}.$$

From the antimonotonicity of s, it follows that

$$x \in L \ \text{ and } \ x \geq y \Rightarrow y \in L.$$

Such a set is called an *down-set*, *decreasing set* or *order ideal.* This algebraic characterisation will turn out to be crucial for the development and analysis of algorithms.

The set of downsets is a subset of the power set of $\mathbb{X}$, however, there are still a very large number of possible downsets. For example, in the case of $d = 6$, the set $\mathbb{X}$ has 64 elements and the power set has $2^{64} \approx 1.810^{19}$ elements and the number of downsets is 7,828,354. The simplest downsets are generated by one element and are

$$\downarrow x = \{z \mid z \leq x\}$$

For example, one has

$$\mathbb{X} = \downarrow e.$$

Consider the set of itemsets for which the (empirical) support as defined by a data base D is nonzero. This is the set of all itemsets which are at least contained in one data record. It is

$$L^{(0)} = \bigcup_{x \in D} \downarrow x.$$

This formula can be simplified by considering only the maximal elements $D_{\max}$ in D:

$$L^{(0)} = \bigcup_{x \in D_{\max}} \downarrow x.$$

Any general downset can be represented as a union of the "simple downsets" generated by one element. It follows that for some set $Z \subset \mathbb{X}$ of maximal elements one then has

$$L = \bigcup_{x \in Z} \downarrow x.$$

The aim of frequent itemset mining is to determine Z for a given σ_0. Algorithms to determine such Z will be discussed in the next sections. Note that $L \subset L^{(0}$ and that the representation of L can be considerably more complex than the representation of $L^{(0)}$. As illustration, consider the case where e is in the data base. In this case $L^{(0)} = \mathbb{X} =\downarrow e$ but e is usually not going to be a frequent itemset.

2.4. *General search for itemsets and search for rules*

The previous subsections considered the search for itemsets which were frequently occurring in a database. One might be interested in more general characterisations, maybe searching for itemsets for which the income from their sale amounted to some minimum figure or which combined only certain items together. Thus one has a criterion or predicate $a(x)$ which is true for items of interest. It is assumed that the evaluation of this criterion is expensive and requires reading the database. One would now like to find all "interesting" items and would like to do this without having to consider all possible itemsets. This search for interesting itemsets is considerably more challenging. In some cases one finds, however, that the sets to be found are again downsets and then similar algorithms can be employed. Often, however, one has to resort to heuristics and approximate methods.

Once frequent itemsets are available one can find *strong rules.* These rules are of the type "if itemset x is in a record than so is itemset y". Such a rule is written as $x \Rightarrow y$ and is defined by a pair of itemsets $(x, y) \in \mathbb{X}^2$. The proportion of records for which this rule holds is called the *confidence*, it is defined formally as

$$c(x \Rightarrow y) = \frac{s(x \vee y)}{s(x)}.$$

A strong rule is given by a rule $x \Rightarrow y$ for which $x \vee y$ is frequent, i.e., $s(x \vee y) \geq \sigma_0$ for a given $\sigma_0 > 0$ and for which $c(x \Rightarrow y) \geq \gamma_0$ for a given $\gamma_0 > 0$. The constants σ_0, γ_0 are provided by the user and their careful choice is crucial to the detection of sensible rules. From the definition it follows that the confidence is the conditional anticumulative distribution, i.e.,

$$c(a_x \Rightarrow a_y) = F^\delta(y|x)$$

where $F^\delta(y|x) = F^\delta(y \vee x)/F^\delta(x)$ is the conditional cumulative distribution function. Now there are several problems with strong association rules which have been addressed in the literature:

- The straight-forward interpretation of the rules may lead to wrong inferences.
- The number of strong rules found can be very small.
- The number of strong rules can be very large.
- Most of the strong rules found can be inferred from domain knowledge and do not lead to new insights.
- The strong rules found do not lend themselves to any actions and are hard to interpret.

We will address various of these challenges in the following. At this stage the association rule mining problem consists of the following:

$$\boxed{\textit{Find all strong association rules in a given data set } D.}$$

A simple procedure would now visit each frequent itemset z and look at all pairs z_1, z_2 such that $z = z_1 \vee z_2$ and $z_1 \wedge z_2 = 0$ and consider all rules $a_{z_1} \Rightarrow a_{z_2}$. This procedure can be improved by taking into account that

Theorem 7: Let $z = z_1 \vee z_2 = z_3 \vee z_4$ and $z_1 \wedge z_2 = z_3 \wedge z_4 = 0$. Then if $a_{z_1} \Rightarrow a_{z_2}$ is a strong association rule and $z_3 \geq z_1$ then so is $a_{z_3} \Rightarrow a_{z_4}$.

This is basically "the apriori property for the rules" and allows pruning the tree of possible rules quite a lot. The theorem is again used as a necessary condition. We start the algorithm by considering $z = z_1 \vee z_2$ with 1-itemsets for z_2 and looking at all strong rules. Then, if we consider a 2-itemset for z_2 both subsets $y < z_2$ need to be consequents of strong rules in order for z_2 to be a candidate of a consequent. By constructing the consequents taking into account that all their nearest neighbours (their cover in lattice terminology) need to be consequents as well. Due to the interpretability problem one is mostly interested in small consequent itemsets so that this is not really a big consideration. See [16] for efficient algorithms for the direct search for association rules.

3. The Apriori Algorithm

The aim of association rule discovery is the derivation of *if-then-rules* based on the itemsets x defined in the previous subsection. An example of such a rule is "if a market basket contains orange juice then it also contains bread". In this section the Apriori algorithm to find all frequent itemsets is discussed. The classical Apriori algorithm as suggested by Agrawal et al. in [3] is one of the most important data mining algorithms. It uses a breadth first search approach, first finding all frequent 1-itemsets, and then discovering 2-itemsets and continues by finding increasingly larger frequent itemsets. The three subsections of this section consider first the problem of the determination of the support of any itemset and the storage of the data in memory, then the actual apriori algorithm and finally the estimation of the size of candidate itemsets which allows the prediction of computational time as the size of the candidates is a determining factor in the complexity of the apriori algorithm.

3.1. *Time complexity – Computing supports*

The data is a sequence $x^{(1)}, \ldots, x^{(n)}$ of binary vectors. We can thus represent the data as a n by d binary matrix. The number of nonzero elements is $\sum_{i=1}^{n} |x^{(i)}|$. This is approximated by the n times expected length, i.e., $nE(|x|)$. So the proportion of nonzero elements is $E(|x|)/d$. This can be very small, especially for the case of market baskets, where out of often more than 10,000 items usually less than 100 items are purchased. Thus less than one percent of all the items are nonzero. In this case it makes sense to store the matrix in a sparse format. Here we will consider two ways to store the matrix, either by rows or by columns. The matrix corresponding to an earlier example is

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} .$$

First we discuss the *horizontal organisation.* A row is represented simply by the indices of the nonzero elements. And the matrix is represented as a tuple of rows. For example, the above matrix is represented as

$$\big[(1,2)\ (1,3,4)\ (1,5)\ (1,2,4)\ (5)\big]$$

In practice we also need pointers which tell us where the row starts if contiguous locations are used in memory.

Now assume that we have any row x and a a_z and would like to find out if x supports a_z, i.e., if $z \leq x$. If both the vectors are represented in the sparse format this means that we would like to find out if the indices of z are a subset of the indices of x. There are several different ways to do this and we will choose the one which uses an auxiliary bitvector $v \in \mathbb{X}$ (in full format) which is initialised to zero. The proposed algorithm has 3 steps:

(1) Expand x into a bitvector v: $v \leftarrow x$.
(2) Extract the value of v for the elements of z, i.e., $v[z]$. If they are all nonzero, i.e., if $v[z] = e$ then $z \leq x$.
(3) Set v to zero again, i.e., $v \leftarrow 0$.

We assume that the time per nonzero element for all the steps is the same τ and we get for the time:

$$T = (2|x| + |z|)\tau.$$

In practice we will have to determine if $a_z(x)$ holds for m_k different vectors z which have all the same length k. Rather than doing the above algorithm m_k times one can extract x once and one so gets the algorithm

(1) Extract x into v: $v \leftarrow x$.
(2) For all $j = 1, \ldots, m_k$ check if $v[z^{(j)}] = e$, i.e., if $z^{(j)} \leq x$.
(3) Set v to zero again, i.e., $v \leftarrow 0$.

With the same assumptions as above we get ($|z^{(j)}| = k$) for the time:

$$T = (2|x| + m_k k)\tau.$$

Finally, running this algorithm for all the rows $x^{(i)}$ and vectors $z^{(j)}$ of different lengths, one gets the total time

$$T = \sum_k (2 \sum_{i=1}^{n} |x^{(i)}| + m_k k n)\tau$$

and the expected time is

$$E(T) = \sum_k (2E(|x|) + m_k k) n\tau.$$

Note that the sum over k is for k between one and d but only the k for which $m_k > 0$ need to be considered. The complexity has two parts. The first part is proportional to $E(|x|)n$ which corresponds to the number of data points times the average complexity of each data point. This part thus encapsulates the data dependency. The second part is proportional to $m_k k n$ where the factor $m_k k$ refers to the complexity of the search space which has to be visited for each record n. For $k = 1$ we have $m_1 = 1$ as we need to consider all the components. Thus the second part is larger than $dn\tau$, in fact, we would probably have to consider all the pairs so that it would be larger than $d^2 n\tau$ which is much larger than the first part as

$2E(|x|) \leq 2d$. Thus the major cost is due to the search through the possible patterns and one typically has a good approximation

$$E(T) \approx \sum_k m_k k n \tau.$$

An alternative is based on the *vertical organisation* where the binary matrix (or Boolean relational table) is stored column-wise. This may require slightly less storage as the row wise storage as we only needs pointers to each column and one typically has more rows than columns. In this vertical storage scheme the matrix considered earlier would be represented as

$$\big[(1,2,3,4)\ (1,4)\ (2)\ (2,4)\ (3,5)\big]$$

The storage savings in the vertical format however, are offset by extra storage costs for an auxiliary vector with n elements.

For any a_z the algorithm considers only the columns for which the components z_j are one. The algorithm determines the intersection (or elementwise product) of all the columns j with $z_j = 1$. This is done by using the auxiliary array v which holds the current intersection. We initially set it to the first column j with $z_j = 1$, later extract all the values at the points defined by the nonzero elements for the next column j' for which $z_{j'} = 1$, then zero the original ones in v and finally set the extracted values into the v. More concisely, we have the algorithm, where x_j stands for the whole column j in the data matrix.

```
(1) Get $j$ such that $z_j = 1$, mark as visited
(2) Extract column $x_j$ into $v$: $v \leftarrow x_j$
(3) Repeat until no nonzero elements in $z$ unvisited:
    (a) Get unvisited $j$ such that $z_j = 1$, mark as visited
    (b) Extract elements of $v$ corresponding to $x_j$, i.e., $w \leftarrow v[x_j]$
    (c) Set $v$ to zero, $v \leftarrow 0$
    (d) Set $v$ to $w$, $v \leftarrow w$
(4) Get the support $s(a_z) = |w|$
```

So we access v three times for each column, once for the extraction of elements, once for setting it to zero and once for resetting the elements. Thus for the determination of the support of z in the data base we have the time complexity of

$$T = 3\tau \sum_{j=1}^{d} \sum_{i=1}^{n} x_j^{(i)} z_j.$$

A more careful analysis shows that this is actually an upper bound for the complexity. Now this is done for m_k arrays $z^{(s,k)}$ of size k and for all k. Thus we get the total time for the determination of the support of all a_z to be

$$T = 3\tau \sum_k \sum_{s=1}^{m_k} \sum_{j=1}^{d} \sum_{i=1}^{n} x_j^{(i)} z_j^{(s,k)}.$$

We can get a simple upper bound for this using $x_j^{(i)} \leq 1$ as

$$T \leq 3 \sum_k m_k k n \tau$$

because $\sum_{j=1}^{d} z_j^{(s,k)} = k$. This is roughly 3 times what we got for the previous algorithm. However, the $x_j^{(i)}$ are random with an expectation $E(x_j^{(i)})$ which is typically much less than one and have an average expected length of $E(|x|)/d$. If we introduce this into the equation for T we get the approximation

$$E(T) \approx \frac{3E(|x|)}{d} \sum_k m_k k n \tau$$

which can be substantially smaller than the time for the previous algorithm.

Finally, we should point out that there are many other possible algorithms and other possible data formats. Practical experience and more careful analysis shows that one method may be more suitable for one data set where the other is better for another data set. Thus one carefully has to consider the specifics of a data set. Another consideration is also the size k and number m_k of the z considered. It is clear from the above that it is essential to carefully choose the "candidates" a_z for which the support will be determined. This will further be discussed in the next sections. There is one term which occurred in both algorithms above and which characterises the complexity of the search through multiple levels of a_z, it is:

$$C = \sum_{k=1}^{\infty} m_k k.$$

We will use this constant later in the discussion of the efficiency of the search procedures.

3.2. *The algorithm*

Some principles of the apriori algorithm are suggested in [2]. In particular, the authors suggest a breadth-first search algorithm and utilise the

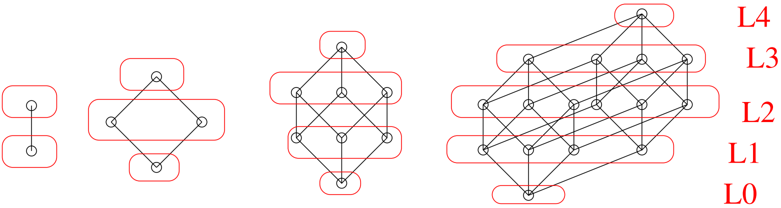


Fig. 8. Level sets of Boolean lattices.

Algorithm 1 Apriori

```
$C_1 = \mathcal{A}(\mathbb{X})$ is the set of all one-itemsets, $k = 1$
while $C_k \neq \emptyset$ do
    scan database to determine support of all $a_y$ with $y \in C_k$
    extract frequent itemsets from $C_k$ into $L_k$
    generate $C_{k+1}$
    $k := k + 1$.
end while
```

apriori principle to avoid unnecessary processing. However, the problem with this early algorithm is that it generates candidate itemsets for each record and also cannot make use of the vertical data organisation. Consequently, two groups of authors suggested at the same conference a new and faster algorithm which determines candidate itemsets before each data base scan [3,13]. This approach has substantially improved performance and is capable of utilising the vertical data organisation. We will discuss this algorithm using the currently accepted term *frequent itemsets.* (This was in the earlier literature called "large itemsets" or "covering itemsets".)

The apriori algorithm visits the lattice of itemsets in a level-wise fashion, see Figure 8 and Algorithm 1. Thus it is a *breadth-first-search* or BFS procedure. At each level the data base is scanned to determine the support of items in the *candidate itemset* C_k. Recall from the last section that the major determining parameter for the complexity of the algorithm is $C = \sum_k m_k k$ where $m_k = |C_k|$.

It is often pointed out that much of the time is spent in dealing with pairs of items. We know that $m_1 = d$ as one needs to consider all single items. Furthermore, one would not have any items which alone are not frequent and so one has $m_2 = d(d-1)/2$. Thus we get the lower bound for C:

$$C \leq m_1 + 2m_2 = d^2.$$

As one sees in practice that this is a large portion of the total computations one has a good approximation $C \approx d^2$. Including also the dependence on the data size we get for the time complexity of apriori:

$$T = O(d^2 n).$$

Thus we have scalability in the data size but quadratic dependence on the dimension or number of attributes.

Consider the first (row-wise) storage where $T \approx d^2 n\tau$. If we have $d = 10,000$ items and $n = 1,000,000$ data records and the speed of the computations is such that $\tau = 1$ns the apriori algorithm would require 10^5 seconds which is around 30 hours, more than one day. Thus the time spent for the algorithm is clearly considerable.

In case of the second (column-wise) storage scheme we have $T \approx 3E(|x|)dn\tau$. Note that in this case for fixed size of the market baskets the complexity is now

$$T = O(dn).$$

If we take the same data set as before and we assume that the average market basket contains 100 items ($E(|x|) = 100$) then the apriori algorithm would require only 300 seconds or five minutes, clearly a big improvement over the row-wise algorithm.

3.3. *Determination and size of the candidate itemsets*

The computational complexity of the apriori algorithm is dominated by data scanning, i.e., evaluation of $a_z(x)$ for data records x. We found earlier that the complexity can be described by the constant $C = \sum_k m_k k$. As m_k is the number of k itemsets, i.e., bitvectors where exactly k bits are one we get $m_k \leq \binom{m}{k}$. On the other hand the apriori algorithm will find the set L_k of the actual frequent itemsets, thus $L_k \subset C_k$ and so $|L_k| \leq m_k$. Thus one gets for the constant C the bounds

$$\sum_k k|L_k| \leq C = \sum_k m_k k \leq \sum_{k=0}^{d} \binom{d}{k} k.$$

The upper bound is hopeless for any large size d and we need to get better bounds. This depends very much on how the candidate itemsets C_k are chosen. We choose C_1 to be the set of all 1-itemsets, and C_2 to be the set of all 2-itemsets so that we get $m_1 = 1$ and $m_2 = d(d-1)/2$.

The apriori algorithm determines alternatively C_k and L_k such that successively the following chain of sets is generated:

$$C_1 = L_1 \to C_2 \to L_2 \to C_3 \to L_3 \to C_4 \to \cdots$$

How should we now choose the C_k? We know, that the sequence L_k satisfies the *apriori property*, which can be reformulated as

Definition 8: If y is a frequent k-itemset (i.e., $y \in L_k$) and if $z \leq y$ then z is a frequent $|z|$-itemset, i.e., $z \in L_{|z|}$.

Thus in extending a sequence $L_1, L_2, \ldots, L_k$ by a C_{k+1} we can choose the candidate itemset such that the extended sequence still satisfies the apriori property. This still leaves a lot of freedom to choose the candidate itemset. In particular, the empty set would always be admissible. We need to find a set which contains L_{k+1}. The apriori algorithm chooses the *largest* set C_{k+1} which satisfies the apriori condition. But is this really necessary? It is if we can find a data set for which the extended sequence is the set of frequent itemsets. This is shown in the next proposition:

Proposition 9: *Let $L_1, \ldots, L_m$ be any sequence of sets of k-itemsets which satisfies the apriori condition. Then there exists a dataset D and a $\sigma > 0$ such that the L_k are frequent itemsets for this dataset with minimal support σ.*

Proof: Set $x^{(i)} \in \bigcup_k L_k,\ i = 1, \ldots, n$ to be sequence of all maximal itemsets, i.e., for any $z \in \bigcup_k L_k$ there is an $x^{(i)}$ such that $z \leq x^{(i)}$ and $x^{(i)} \not\leq x^{(j)}$ for $i \neq j$. Choose $\sigma = 1/n$. Then the L_k are the sets of frequent itemsets for this data set. □

For any collection L_k there might be other data sets as well, the one chosen above is the minimal one. The sequence of the C_k is now characterised by:

(1) $C_1 = L_1$
(2) If $y \in C_k$ and $z \leq y$ then $z \in C_{k'}$ where $k' = |z|$.

In this case we will say that *the sequence C_k satisfies the apriori condition.* It turns out that this characterisation is strong enough to get good upper bounds for the size of $m_k = |C_k|$.

However, before we go any further in the study of bounds for $|C_k|$ we provide a construction of a sequence C_k which satisfies the apriori condition. A first method uses L_k to construct C_{k+1} which it chooses to be the maximal set such that the sequence $L_1, \dots, L_k, C_{k+1}$ satisfies the apriori property. One can see by induction that then the sequence $C_1, \dots, C_{k+1}$ will also satisfy the apriori property. A more general approach constructs $C_{k+1}, \dots, C_{k+p}$ such that $L_1, \dots, L_k, C_{k+1}, \dots, C_{k+p}$ satisfies the apriori property. As p increases the granularity gets larger and this method may work well for larger itemsets. However, choosing larger p also amounts to larger C_k and thus some overhead. We will only discuss the case of $p = 1$ here.

The generation of C_{k+1} is done in two steps. First a slightly larger set is constructed and then all the elements which break the apriori property are removed. For the first step the join operation is used. To explain join let the elements of L_1 (the atoms) be enumerated as $e_1, \dots, e_d$. Any itemset can then be constructed as join of these atoms. We denote a general itemset by

$$e(j_1, \dots, j_k) = e_{j_1} \vee \cdots \vee e_{j_k}$$

where $j_1 < j_2 < \cdots < j_k$. The *join* of any k-itemset with itself is then defined as

$$L_k \bowtie L_k := \{e(j_1, \dots, j_{k+1}) \mid e(j_1, \dots, j_k) \in L_k, \\ e(j_1, \dots, j_{k-1}, j_{k+1}) \in L_k\}.$$

Thus $L_k \bowtie L_k$ is the set of all $k + 1$ itemsets for which 2 subsets with k items each are frequent. As this condition also holds for all elements in C_{k+1} one has $C_{k+1} \subset L_k \bowtie L_k$. The C_{k+1} is then obtained by removing elements which contain infrequent subsets.

For example, if $(1, 0, 1, 0, 0) \in L_2$ and $(0, 1, 1, 0, 0) \in L_2$ then $(1, 1, 1, 0, 0) \in L_2 \bowtie L_2$. If, in addition, $(1, 1, 0, 0, 0) \in L_2$ then $(1, 1, 1, 0, 0) \in C_3$.

Having developed a construction for C_k we can now determine the bounds for the size of the candidate itemsets based purely on combinatorial considerations. The main tool for our discussion is the Kruskal-Katona theorem. The proof of this theorem and much of the discussion follows closely the exposition in Chapter 5 of [4]. The bounds developed in this way have first been developed in [9].

We will denote the set of all possible k-itemsets or bitvectors with exactly k bits as $\mathcal{I}_k$. Subsets of this set are sometimes also called hypergraphs in the literature. The set of candidate itemsets $C_k \subset \mathcal{I}_k$.

Given a set of k-itemsets C_k the *lower shadow of* C_k is the set of all $k - 1$ subsets of the elements of C_k:

$$\partial(C_k) := \{y \in \mathcal{I}_{k-1} \mid y < z \text{ for some } z \in C_k\}.$$

Table 1. All bitvectors with two bits out of five set and their integer.

(0,0,0,1,1)	3
(0,0,1,0,1)	5
(0,0,1,1,0)	6
(0,1,0,0,1)	9
(0,1,0,1,0)	10
(0,1,1,0,0)	12
(1,0,0,0,1)	17
(1,0,0,1,0)	18
(1,0,1,0,0)	20
(1,1,0,0,0)	24

This is the set of bitvectors which have $k-1$ bits set at places where some $z \in C_k$ has them set. The shadow ∂C_k can be smaller or larger than the C_k. In general, one has for the size $|\partial C_k| \geq k$ independent of the size of C_k. So, for example, if $k = 20$ then $|\partial C_k| \geq 20$ even if $|C_k| = 1$. (In this case we actually have $|\partial C_k| = 20$.) For example, we have $\partial C_1 = \emptyset$, and $|\partial C_2| \leq d$.

It follows now that the sequence of sets of itemsets C_k satisfies the apriori condition iff

$$\partial(C_k) \subset C_{k-1}.$$

The Kruskal-Katona Theorem provides an estimate of the size of the shadow.

First, recall the mapping $\phi : \mathbb{X} \to \mathbb{N}$, defined by:

$$\phi(x) = \sum_{i=0}^{d-1} 2^{ix_i},$$

and the induced order

$$y \prec z :\Leftrightarrow \phi(y) < \phi(z)$$

which is the *colex (or colexicographic) order*. In this order the itemset $\{3,5,6,9\} \prec \{3,4,7,9\}$ as the largest items determine the order. (In the lexicographic ordering the order of these two sets would be reversed.)

Let $[m] := \{0,\ldots,m-1\}$ and $[m]^{(k)}$ be the set of all k-itemsets where k bits are set in the first m positions and all other bits can be either 0 or 1. In the colex order any z where bits m (and beyond) are set are larger than any of the elements in $[m]^{(k)}$. Thus $[m]^k$ is just the set of the first $\binom{m-1}{k}$ bitvectors with k bits set.

We will now construct the sequence of the first m bitvectors for any m. This corresponds to the first numbers, which, in the binary representation have m ones set. Consider, for example the case of $d = 5$ and $k = 2$. For this case all the bitvectors are in table 1. (Printed with the lowest significant bit on the right hand side for legibility.)

As before, we denote by e_j the $j-th$ atom and by $e(j_1, \ldots, j_k)$ the bitvector with bits $j_1, \ldots, j_k$ set to one. Furthermore, we introduce the element-wise join of a bitvector and a set C of bitvectors as:

$$C \vee y := \{z \vee y \mid z \in C\}.$$

For $0 \leq s \leq m_s < m_{s+1} < \cdots < m_k$ we introduce the following set of k-itemsets:

$$B^{(k)}(m_k, \ldots, m_s) := \bigcup_{j=s}^{k} \Big([m_j]^{(j)} \vee e(m_{j+1}, \ldots, m_k) \Big) \subset \mathcal{I}_k.$$

As only term j does not contain itemsets with item m_j (all the others do) the terms are pairwise disjoint and so the union contains

$$|B^{(k)}(m_k, \ldots, m_s)| = b^{(k)}(m_k, \ldots, m_s) := \sum_{j=s}^{k} \binom{m_j}{j}$$

k-itemsets. This set contains the first (in colex order) bitvectors with k bits set. By splitting off the last term in the union one then sees:

$$B^{(k)}(m_k, \ldots, m_s) = \Big(B^{(k-1)}(m_{k-1}, \ldots, m_s) \vee e_{m_k} \Big) \cup [m_k]^{(k)} \qquad (3)$$

and consequently

$$b^{(k)}(m_k, \ldots, m_s) = b^{(k-1)}(m_{k-1}, \ldots, m_s) + b^{(k)}(m_k).$$

Consider the example of table 1 of all bitvectors up to $(1, 0, 0, 1, 0)$. There are 8 bitvectors which come earlier in the colex order. The highest bit set for the largest element is bit 5. As we consider all smaller elements we need to have all two-itemsets where the 2 bits are distributed between positions 1 to 4 and there are $\binom{4}{2} = 6$ such bitvectors. The other cases have the top bit fixed at position 5 and the other bit is either on position 1 or two thus there are $\binom{2}{1} = 2$ bitvectors for which the top bit is fixed. Thus we get a total of

$$\binom{4}{2} + \binom{2}{1} = 8$$

bitvectors up to (and including) bitvector $(1, 0, 0, 1, 0)$ for which 2 bits are set. This construction is generalised in the following.

In the following we will show that $b^{(k)}(m_k,\ldots,m_s)$ provides a unique representation for the integers. We will make frequent use of the identity:

$$\binom{t+1}{r} - 1 = \sum_{l=1}^{r} \binom{t-r+l}{l}. \tag{4}$$

Lemma 10: *For every $m, k \in \mathbb{N}$ there are numbers $m_s < \cdots < m_k$ such that*

$$m = \sum_{j=s}^{k} \binom{m_j}{j} \tag{5}$$

and the m_j are uniquely determined by m.

Proof: The proof is by induction over m. In the case of $m = 1$ one sees immediately that there can only be one term in the sum of equation (5), thus $s = k$ and the only choice is $m_k = k$.

Now assume that equation (5) is true for some $m' = m - 1$. We show uniqueness for m. We only need to show that m_k is uniquely determined as the uniqueness of the other m_j follows from the induction hypothesis applied to $m' = m - m - \binom{m_k}{k}$.

Assume a decomposition of the form (5) is given. Using equation (4) one gets:

$$\begin{aligned} m &= \binom{m_k}{k} + \binom{m_{k-1}}{k-1} + \cdots + \binom{m_s}{s} \\ &\leq \binom{m_k}{k} + \binom{m_k - 1}{k-1} + \cdots + \binom{m_k - k + 1}{1} \\ &= \binom{m_k + 1}{k} - 1 \end{aligned}$$

as $m_{k-1} \leq m_k - 1$ etc. Thus we get

$$\binom{m_k}{k} \leq m \leq \binom{m_k+1}{k} - 1.$$

With other words, m_k is the largest integer such that $\binom{m_k}{k} \leq m$. This provides a unique characterisation of m_k which proves uniqueness.

Assume that the m_j be constructed according to the method outlined in the first part of this proof. One can check that equation (5) holds for these m_j using the characterisation.

What remains to be shown is $m_{j+1} > m_j$ and using inductions, it is enough to show that $m_{k-1} < m_k$. If, on the contrary, this does not hold and $m_{k-1} \geq m_k$, then one gets from (4):

$$\begin{aligned} m &\geq \binom{m_k}{k} + \binom{m_{k-1}}{k-1} \\ &\geq \binom{m_k}{k} + \binom{m_k}{k-1} \\ &= \binom{m_k}{k} + \binom{m_k - 1}{k-1} + \cdots + \binom{m_k - k + 1}{1} + 1 \\ &\geq \binom{m_k}{k} + \binom{m_{k-1}}{k-1} + \cdots + \binom{m_s}{s} + 1 \text{ by induction hyp.} \\ &= m + 1 \end{aligned}$$

which is not possible. □

Let $N^{(k)}$ be the set of all k-itemsets of integers. It turns out that the $B^{(k)}$ occur as natural subsets of $N^{(k)}$:

Theorem 11: The set $B^{(k)}(m_k, \ldots, m_s)$ consists of the first $m = \sum_{j=s}^{k} \binom{m_j}{j}$ itemsets of $N^{(k)}$ (in colex order).

Proof: The proof is by induction over $k - s$. If $k = s$ and thus $m = \binom{m_k}{k}$ then the first elements of $N^{(k)}$ are just $[m_k]^{(k)}$. If $k > s$ then the first $\binom{m_k}{k}$ elements are still $[m_k]^{(k)}$. The remaining $m - \binom{m_k}{k}$ elements all contain bit m_k. By the induction hypothesis the first $b^{k-1}(m_{k-1}, \ldots, m_s)$ elements containing bit m_k are $B^{k-1}(m_{k-1}, \ldots, m_s) \vee e_{m_k}$ and the rest follows from (3). □

The shadow of the first k-itemsets $B^{(k)}(m_k, \ldots, m_s)$ are the first $k-1$-itemsets, or more precisely:

Lemma 12:

$$\partial B^{(k)}(m_k, \ldots, m_s) = B^{(k-1)}(m_k, \ldots, m_s).$$

Proof: First we observe that in the case of $s = k$ the shadow is simply set of all $k - 1$ itemsets:

$$\partial [m_k]^k = [m_k]^{(k-1)}.$$

This can be used as anchor for the induction over $k - s$. As was shown earlier, one has in general:

$$B^{(k)}(m_k, \ldots, m_s) = [m_k]^k \cup \left(B^{(k-1)}(m_{k-1}, \ldots, m_s) \vee e_{m_k} \right)$$

and, as the shadow is additive, as

$$\begin{aligned}\partial B^{(k)}(m_k,\dots,m_s) &= [m_k]^{(k-1)} \cup \left(B^{(k-2)}(m_{k-1},\dots,m_s) \vee e_{m_k}\right)\\ &= B^{(k-1)}(m_k,\dots,m_s).\end{aligned}$$

Note that $B^{(k-1)}(m_{k-1},\dots,m_s) \subset [m_l]^{(k-1)}$. □

The shadow is important for the apriori property and we would thus like to determine the shadow, or at least its size for more arbitrary k-itemsets as they occur in the apriori algorithm. Getting bounds is feasible but one requires special technology to do this. This is going to be developed further in the sequel. We would like to reduce the case of general sets of k-itemsets to the case of the previous lemma, where we know the shadow. So we would like to find a mapping which maps the set of k-itemsets to the first k itemsets in colex order without changing the size of the shadow. We will see that this can almost be done in the following. The way to move the itemsets to earlier ones (or to "compress" them) is done by moving later bits to earlier positions.

So we try to get the k itemsets close to $B^{(k)}(m_k,\dots,m_s)$ in some sense, so that the size of the shadow can be estimated. In order to simplify notation we will introduce $z+e_j$ for $z \vee e_j$ when $e_j \not\leq z$ and the reverse operation (removing the j-th bit) by $z - e_j$ when $e_j \leq z$. Now we introduce *compression* of a bitvector as

$$R_{ij}(z) = \begin{cases} z - e_j + e_i & \text{if } e_i \not\leq z \text{ and } e_j \leq z\\ z & \text{else}\,. \end{cases}$$

Thus we simply move the bit in position j to position i if there is a bit in position j and position i is empty. If not, then we don't do anything. So we did not change the number of bits set. Also, if $i < j$ then we move the bit to an earlier position so that $R_{ij}(z) \leq z$. For our earlier example, when we number the bits from the right, starting with 0 we get $R_{1,3}((0,1,1,0,0)) = (0,0,1,1,0)$ and $R_{31}((0,0,0,1,1)) = (0,0,0,1,1)$. This is a "compression" as it moves a collection of k-itemsets closer together and closer to the vector $z = 0$ in terms of the colex order.

The mapping R_{ij} is not injective as

$$R_{ij}(z) = R_{ij}(y)$$

when $y = R_{ij}(z)$ and this is the only case. Now consider for any set C of bitvectors the set $R_{ij}^{-1}(C) \cap C$. These are those elements of C which stay in C when compressed by R_{ij}. The compression operator for bitsets is now defined as

$$\tilde{R}_{i,j}(C) = R_{ij}(C) \cup (C \cap R_{ij}^{-1}(C)).$$

Thus the points which stay in C under R_{ij} are retained and the points which are mapped outside C are added. Note that by this we have avoided the problem with the non-injectivity as only points which stay in C can be mapped onto each other. The size of the compressed set is thus the same. However, the elements in the first part have been mapped to earlier elements in the colex order. In our earlier example, for $i, j = 1, 3$ we get

$$C = \{(0,0,0,1,1), (0,1,1,0,0), (1,1,0,0,0), (0,1,0,1,0)\}$$

we get

$$\tilde{R}_{i,j}(C) = \{(0,0,0,1,1), (0,0,1,1,0), (1,1,0,0,0), (0,1,0,1,0)\}.$$

Corresponding to this compression of sets we introduce a mapping $\tilde{R}_{i,j}$ (which depends on C) by $\tilde{R}_{i,j}(y) = y$ if $R_{i,j}(y) \in C$ and $\tilde{R}_{i,j}(y) = R_{i,j}(y)$ else. In our example this maps corresponding elements in the sets onto each other. In preparation, for the next lemma we need the simple little result:

Lemma 13: *Let C be any set of k itemsets and $z \in C, e_j \leq z, e_i \not\leq z$. Then*

$$z - e_j + e_i \in \tilde{R}_{i,j}(C).$$

Proof: There are two cases to consider:

(1) Either $z - e_j + e_i \notin C$ in which case $z - e_j + e_i = \tilde{R}_{i,j}(z)$.
(2) Or $z - e_j + e_i \in C$ and as $z - e_j + e_i = \tilde{R}_{i,j}(z - e_j + e_i)$ one gets again $z - e_j + e_i \in \tilde{R}_{i,j}(C)$. □

The next result shows that in terms of the shadow, the "compression" $\tilde{R}_{i,j}$ really is a compression as the shadow of a compressed set can never be larger than the shadow of the original set. We suggest therefor to call it compression lemma.

Lemma 14: *Let C be a set of k itemsets. Then one has*

$$\partial\tilde{R}_{i,j}(C) \subset \tilde{R}_{i,j}(\partial C).$$

Proof: Let $x \in \partial\tilde{R}_{i,j}(C)$. We need to show that

$$x \in \tilde{R}_{i,j}(\partial C)$$

and we will enumerate all possible cases.

First notice that there exists a $e_k \not\leq x$ such that

$$x + e_k \in \tilde{R}_{i,j}(C)$$

so there is an $y \in C$ such that

$$x + e_k = \tilde{R}_{i,j}(y).$$

(1) In the first two cases $\tilde{R}_{i,j}(y) \neq y$ and so one has (by the previous lemma)

$$x + e_k = y - e_j + e_i, \quad \text{for some } y \in C, e_j \leq y, e_i \not\leq y.$$

(a) First consider $i \neq k$. Then there is a bitvector z such that $y = z + e_k$ and $z \in \partial C$. Thus we get

$$x = z - e_j + e_i \in \tilde{R}_{i,j}(\partial C)$$

as $z \in \partial C$ and with lemma 13.

(b) Now consider $i = k$. In this case $x + e_i = y - e_j + e_i$ and so $x = y - e_j \in \partial C$. As $e_j \not\leq x$ one gets

$$x = \tilde{R}_{i,j}(x) \in \tilde{R}_{i,j}(\partial C).$$

(2) In the remaining cases $\tilde{R}_{i,j}(y) = y$, i.e., $x + e_k = \tilde{R}_{i,j}(x + e_k)$. Thus $x + e_k = y \in C$ and so $x \in \partial C$. Note that $\tilde{R}_{i,j}$ actually depends on ∂C!

(a) In the case where $e_j \not\leq x$ one has $x = \tilde{R}_{i,j}(x) \in \tilde{R}_{i,j}(\partial C)$.

(b) In the other case $e_j \leq x$. We will show that $x = \tilde{R}_{i,j}(x)$ and, as $x \in \partial C$ one gets $x \in \tilde{R}_{i,j}(\partial C)$.

i. If $k \neq i$ then one can only have $x + e_k = \tilde{R}_{i,j}(x + e_k)$ if either $e_i \leq x$, in which case $x = \tilde{R}_{i,j}(x)$, or $x - e_j + e_i + e_k \in C$ in which case $x - e_j + e_i \in \partial C$ and so $x = \tilde{R}_{i,j}(x)$.

ii. Finally, if $k = i$, then $x + e_i \in C$ and so $x - e_j + e_i \in \partial C$ thus $x = \tilde{R}_{i,j}(x)$. □

The operator $\tilde{R}_{i,j}$ maps sets of k itemsets onto sets of k itemsets and does not change the number of elements in a set of k itemsets. One now says that a set of k itemsets C is *compressed* if $\tilde{R}_{i,j}(C) = C$ for all $i < j$. This means that for any $z \in C$ one has again $R_{ij}(z) \in C$. Now we can move to prove the key theorem:

Theorem 15: Let $k \geq 1, A \subset N^{(k)}$, $s \leq m_s < \cdots < m_k$ and

$$|A| \geq b^{(k)}(m_k, \ldots, m_s)$$

then

$$|\partial A| \geq b^{(k-1)}(m_k, \dots, m_s).$$

Proof: First we note that the shadow is a monotone function of the underlying set, i.e., if $A_1 \subset A_2$ then $\partial A_1 \subset \partial A_2$. From this it follows that it is enough to show that the bound holds for $|A| = b^{(k)}(m_k, \dots, m_s)$.

Furthermore, it is sufficient to show this bound for compressed A as compression at most reduces the size of the shadow and we are looking for a lower bound. Thus we will assume A to be compressed in the following.

The proof uses double induction over k and $m = |A|$. First we show that the theorem holds for the cases of $k = 1$ for any m and $m = 1$ for any k. In the induction step we show that if the theorem holds for $1, \dots, k-1$ and any m and for $1, \dots, m-1$ and k then it also holds for k and m, see figure 9.

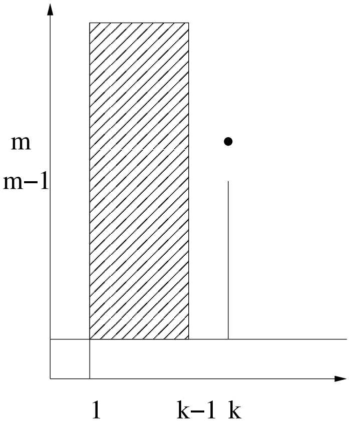


Fig. 9. Double induction.

In the case of $k = 1$ (as A is compressed) one has:

$$A = B^{(1)}(m) = \{e_0, \dots, e_{m-1}\}$$

and so

$$\partial A = \partial B^{(1)}(m) = \{0\}$$

hence $|\partial A| = 1 = b^{(0)}(m)$.

In the case of $m = |A| = 1$ one has:

$$A = B^{(k)}(k) = \{e(0, \dots, k-1)\}$$

and so:

$$\partial A = \partial B^{(k)}(k) = [k]^{(k-1)}$$

hence $|\partial A| = k = b^{(k-1)}(k)$.

The key step of the proof is a partition of A into bitvectors with bit 0 set and such for which bit 0 is not set: $A = A_0 \cup A_1$ where $A_0 = \{x \in A | x_0 = 0\}$ and $A_1 = \{x \in A | x_0 = 1\}$.

(1) If $x \in \partial A_0$ then $x + e_j \in A_0$ for some $j > 0$. As A is compressed it must also contain $x + e_0 = R_{0j}(x + e_j) \in A_1$ and so $x \in A_1 - e_0$ thus

$$|\partial A_0| \leq |A_1 - e_0| = |A_1|.$$

(2) A special case is $A = B^{(k)}(m_k, \ldots, m_s)$ where one has $|A_0| = b^{(k)}(m_k - 1, \ldots, m_s - 1)$ and $|A_1| = b^{(k-1)}(m_k - 1, \ldots, m_s - 1)$ and thus

$$\begin{aligned} m = b^{(k)}(m_k, \ldots, m_s) = b^{(k)}&(m_k - 1, \ldots, m_s - 1) \\ &+ b^{(k-1)}(m_k - 1, \ldots, m_s - 1) \end{aligned}$$

(3) Now partition ∂A_1 into 2 parts:

$$\partial A_1 = (A_1 - e_0) \cup (\partial(A_1 - e_0) + e_0).$$

It follows from previous inequalities and the induction hypothesis that $|\partial A_1| = |A_1 - e_0| + |\partial(A_1 - e_0) + e_0| = |A_1| + |\partial(A_1 - e_0)| \geq b^{(k-1)}(m_k - 1, \ldots, m_s - 1) + b^{(k-2)}(m_k - 1, \ldots, m_s - 1) = b^{(k-1)}(m_k, \ldots, m_s)$ and hence

$$|\partial A| \geq |\partial A_1| \geq b^{(k-1)}(m_k, \ldots, m_s) \qquad \square$$

This theorem is the tool to derive the bounds for the size of future candidate itemsets based on a current itemset and the apriori principle.

Theorem 16: Let the sequence C_k satisfy the apriori property and let

$$|C_k| = b^{(k)}(m_k, \ldots, m_r).$$

Then

$$|C_{k+p}| \leq b^{(k+p)}(m_k, \ldots, m_r)$$

for all $p \leq r$.

Proof: The reason for the condition on p is that the shadows are well defined.

First, we choose r such that $m_r \le r + p - 1$, $m_{r+1} \le r + 1 + p - 1$, $\ldots$, $m_{s-1} \le s - 1 + p - 1$ and $m_s \ge s + p - 1$. Note that $s = r$ and $s = k + 1$ may be possible.

Now we get an upper bound for the size $|C_k|$:

$$\begin{aligned} |C_k| &= b^{(k)}(m_k, \ldots, m_r) \\ &\le b^{(k)}(m_k, \ldots, m_s) + \sum_{j=1}^{s-1} \binom{j + p - 1}{j} \\ &= b^{(k)}(m_k, \ldots, m_s) + \binom{s + p - 1}{s - 1} - 1 \end{aligned}$$

according to a previous lemma.

If the theorem does not hold then $|C_{k+p}| > b^{(k+p)}(m_j, \ldots, m_r)$ and thus

$$\begin{aligned} |C_{k+p}| &\ge b^{(k+p)}(m_j, \ldots, m_r) + 1 \\ &\ge b^{(k+p)}(m_k, \ldots, m_s) + \binom{s + p - 1}{s + p - 1} \\ &= b^{(k+p)}(m_k, \ldots, m_s, s + p - 1). \end{aligned}$$

Here we can apply the previous theorem to get a lower bound for C_k:

$$|C_k| \ge b^{(k)}(m_k, \ldots, m_s, s + p - 1).$$

This, however is contradicting the higher upper bound we got previously and so we have to have $|C_{k+p}| \le b^{(k+p)}(m_j, \ldots, m_r)$. □

As a simple consequence one also gets tightness:

Corollary 17: *For any m and k there exists a C_k with $|C_k| = m = b^{(k+p)}(m_k, \ldots, m_{s+1})$. such that*

$$|C_{k+p}| = b^{(k+p)}(m_k, \ldots, m_{s+1}).$$

Proof: The C_k consists of the first m k-itemsets in the colexicographic ordering. □

In practice one would know not only the size but also the contents of any C_k and from that one can get a much better bound than the one provided by the theory. A consequence of the theorem is that for L_k with $|L_k| \le \binom{m_k}{k}$ one has $|C_{k+p}| \le \binom{m_k}{k+p}$. In particular, one has $C_{k+p} = \emptyset$ for $k > m_p - p$.

4. Extensions

4.1. *Apriori TID*

One variant of the apriori algorithm discussed above computes supports of itemsets by doing intersections of columns. Some of these intersections are repeated over time and, in particular, entries of the Boolean matrix are revisited which have no impact on the support. The Apriori TID [3] algorithm provides a solution to some of these problems. For computing the supports for larger itemsets it does not revisit the original table but transforms the table as it goes along. The new columns correspond to the candidate itemsets. In this way each new candidate itemset only requires the intersection of two old ones.

The following demonstrates with an example how this works. The example is adapted from [3]. In the first row the itemsets from C_k are depicted. The minimal support is 50 percent or 2 rows. The initial matrix of the tid algorithm is equal to

$$\left[\begin{array}{ccccc}1&2&3&4&5\\\hline 1&0&1&1&0\\0&1&1&0&1\\1&1&1&0&1\\0&1&0&0&1\end{array}\right]$$

Note that the column (or item) four is not frequent and is not considered for C_k. After one step of the Apriori tid one gets the matrix:

$$\left[\begin{array}{cccccc}(1,2)&(1,3)&(1,5)&(2,3)&(2,5)&(3,5)\\\hline 0&1&0&0&0&0\\0&0&0&1&1&1\\1&1&1&1&1&1\\0&0&0&0&1&0\end{array}\right]$$

Here one can see directly that the itemsets $(1,2)$ and $(1,5)$ are not frequent. It follows that there remains only one candidate itemset with three items, namely $(2,3,5)$ and the matrix is

$$\left[\begin{array}{c}(2,3,5)\\\hline 0\\1\\1\\0\end{array}\right]$$

Let $z(j_1,\ldots,j_k)$ denote the elements of C_k. Then the elements in the transformed Boolean matrix are $a_{z(j_1,\ldots,j_k)}(x_i)$.

We will again use an auxiliary array $v \in \{0,1\}^n$. The apriori tid algorithm uses the join considered earlier in order to construct a matrix for the frequent itemsets L_{k+1} from L_k. (As in the previous algorithms it is assumed that all matrices are stored in memory. The case of very large data sets which do not fit into memory will be discussed later.) The key part of the algorithm, i.e., the step from k to $k+1$ is then:

(1) Select a pair of frequent k-itemsets (y,z), mark as read
(2) expand $x^y = \bigwedge_i x_i^{y_i}$, i.e., $v \leftarrow x^y$
(3) extract elements using x^z, i.e., $w \leftarrow v[x^z]$
(4) compress result and reset v to zero, $v \leftarrow 0$

There are three major steps where the auxiliary vector v is accessed. The time complexity for this is

$$T = \sum_{i=1}^{n} (2|(x^{(i)})^y| + |(x^{(i)})^z|)\tau.$$

This has to be done for all elements $y \vee z$ where $y, z \in L_k$. Thus the average complexity is

$$E(T) = \sum_k 3nm_k E(x^y)\tau$$

for some "average" y and $x^y = \bigwedge_i x_i^{y_i}$. Now for all elements in L_k the support is larger than σ, thus $E(x^y) \geq \sigma$. So we get a lower bound for the complexity:

$$E(T) \geq \sum_k 3nm_k\sigma\tau.$$

We can also obtain a simple upper bound if we observe that $E(x^y) \leq E(|x|)/d$ which is true "on average". From this we get

$$E(T) \leq \sum_k 3nm_k \frac{E(|x|)}{d}\tau.$$

Another approximation (typically a lower bound) is obtained if we assume that the components of x are independent. In this case $E(x^y) \approx (E(|x|)/d)^k$ and thus

$$E(T) \geq \sum_k 3nm_k (E(|x|)/d)^k\tau.$$

From this we would expect that for some $r_k \in [1, k]$ we get the approximation

$$E(T) \approx \sum_k 3nm_k (E(|x|)/d)^{r_k} \tau.$$

Now recall that the original column-wise apriori implementation required

$$E(T) \approx \sum_k 3nm_k k (E(|x|)/d) \tau$$

and so the "speedup" we can achieve by using this new algorithm is around

$$S \approx \frac{\sum_k km_k}{\sum_k (E(|x|)/d)^{r_k - 1} m_k}.$$

which can be substantial as both $k \geq 1$ and $E(|x|)/d)^{r_k-1} < 1$. We can see that there are two reasons for the decrease in work: First we have reused earlier computations of $x_{j_1} \wedge \cdots \wedge x_{j_k}$ and second we are able to make use of the lower support of the k-itemsets for larger k. While this second effect does strongly depend on r_k and thus the data, the first effect always holds, so we get a speedup of at least

$$S \geq \frac{\sum_k km_k}{\sum_k m_k},$$

i.e., the average size of the k-itemsets. Note that the role of the number m_k of candidate itemsets maybe slightly diminished but this is still the core parameter which determines the complexity of the algorithm and the need to reduce the size of the frequent itemsets is not diminished.

4.2. *Constrained association rules*

The number of frequent itemsets found by the apriori algorithm will often be too large or too small. While the prime mechanism of controlling the discovered itemsets is the minimal support σ, this may often not be enough. Small collections of frequent itemsets may often contain mostly well known associations whereas large collections may reflect mostly random fluctuations. There are effective other ways to control the amount of itemsets obtained. First, in the case of too many itemsets one can use constraints to filter out trivial or otherwise uninteresting itemsets. In the case of too few frequent itemsets one can also change the attributes or features which define the vector x. In particular, one can introduce new "more general" attributes. For example, one might find that rules including the item "ginger beer" are not frequent. However, rules including "soft drinks" will have much higher support and may lead to interesting new rules. Thus one introduces new more general items. However, including more general items while maintaining the original special items leads to duplications in the itemsets, in our example the itemset containing ginger beer and soft drinks is identical to the set which only contains ginger beer. In order to avoid this one can again introduce constraints, which, in our example would identify the itemset containing ginger beer only with the one containing softdrink and ginger beer.

Constraints are conditions for the frequent itemsets of interest. These conditions take the form "predicate = true" with some predicates

$$b_1(z), \ldots, b_s(z).$$

Thus one is looking for frequent k-itemsets L_k^* for which the b_j are true, i.e.,

$$L_k^* := \{z \in L_k \mid b_j(z) = 1\}.$$

These constraints will reduce the amount of frequent itemsets which need to be further processed, but can they also assist in making the algorithms more efficient? This will be discussed next after we have considered some examples. Note that the constraints are not necessarily simple conjunctions! Examples:

- We have mentioned the rule that any frequent itemset should not contain an item and its generalisation, e.g., it should not contain both soft drinks and ginger beer as this is identical to ginger beer. The constraint is of the form $b(x) = \neg a_y(x)$ where y is the itemset where the "softdrink and ginger beer bits" are set.
- In some cases, frequent itemsets have been well established earlier. An example are crisps and soft drinks. There is no need to rediscover this association. Here the constraint is of the form $b(x) = \neg \delta_y(x)$ where y denotes the itemset "softdrinks and chips".
- In some cases, the domain knowledge tells us that some itemsets are prescribed, like in the case of a medical schedule which prescribes certain procedures to be done jointly but others should not be jointly. Finding these rules is not interesting. Here the constraint would exclude certain z, i.e., $b(z) = \neg \delta_y(z)$ where y is the element to exclude.
- In some cases, the itemsets are related by definition. For example the predicates defined by $|z| > 2$ is a consequence of $|z| > 4$. Having discovered the second one relieves us of the need to discover the first one. This,

however, is a different type of constraint which needs to be considered when defining the search space.

A general algorithm for the determination of the L_k^* determines at every step the L_k (which are required for the continuation) and from those outputs the elements of L_k^*. The algorithm is exactly the same as apriori or apriori tid except that not all frequent itemsets are output. See Algorithm 2. The work is almost exactly the same as for the original apriori algorithm.

Algorithm 2 Apriori with general constraints

```
$C_1 = \mathcal{A}(\mathbb{X})$ is the set of all one-itemsets, $k = 1$
while $C_k \neq \emptyset$ do
    scan database to determine support of all $z \in C_k$
    extract frequent itemsets from $C_k$ into $L_k$
    use the constraints to extract the constrained frequent itemsets in $L_k^*$
    generate $C_{k+1}$
    $k := k + 1$.
end while
```

Now we would like to understand how the constraints can impact the computational performance, after all, one will require less rules in the end and the discovery of less rules should be faster. This, however, is not straight-forward as the constrained frequent itemsets L_k^* do not necessarily satisfy the apriori property. There is, however an important class of constraints for which the apriori property holds:

Theorem 18: If the constraints $b_j,\ j = 1, \ldots, m$ are anti-monotone then the set of constrained frequent itemsets $\{L_k^*\}$ satisfies the apriori condition.

Proof: Let $y \in L_k^*$ and $z \leq y$. As $L_k^* \subset L_k$ and the (unconstrained frequent itemsets) L_k satisfy the apriori condition one has $z \in L_{\text{size}(z)}$.

As the b_j are antimonotone and $y \in L_k^*$ one has

$$b_j(z) \geq b_j(y) = 1$$

and so $b_j(z) = 1$ from which it follows that $z \in L_{\text{size}(z)}^*$. □

When the apriori condition holds one can generate the candidate itemsets C_k in the (constrained) apriori algorithm from the sets L_k^* instead of from the larger L_k. However, the constraints need to be anti-monotone. We know that constraints of the form $a_{z^{(j)}}$ are monotone and thus constraints of the form $b_j = \neg a_{z^{(j)}}$ are antimonotone. Such constraints say that a certain combination of items should not occur in the itemset. An example of this is the case of ginger beer and soft drinks. Thus we will have simpler frequent itemsets in general if we apply such a rule. Note that itemsets have played three different roles so far:

(1) as data points $x^{(i)}$
(2) as potentially frequent itemsets z and
(3) to define constraints $\neg a_{z^{(j)}}$.

The constraints of the kind $b_j = \neg a_{z^{(j)}}$ are now used to reduce the candidate itemsets C_k *prior* to the data scan (this is how we save most). Even better, it turns out that the conditions only need to be checked for level $k = |z^{(j)}|$ where k is the size of the itemset defining the constraint. (This gives a minor saving.) This is summarised in the next theorem:

Theorem 19: Let the constraints be $b_j = \neg a_{z^{(j)}}$ for $j = 1, \ldots, s$. Furthermore let the candidate k-itemsets for L_k^* be sets of k-itemsets such that

$$C_k^* = \{y \in \mathcal{I}_k \mid \text{ if } z < y \text{ then } z \in L_{|z|} \text{ and } b_j(y) = 1\}$$

and a further set defined by

$$\tilde{C}_k = \{y \in \mathcal{I}_k \mid \text{if } z < y \text{ then } z \in L_{|z|} \text{ and if } |z^{(j)}| = k \text{ then } b_j(y) = 1 \ \}.$$

Then $\tilde{C}_k = C_k^*$.

Proof: We need to show that every element $y \in \tilde{C}_k$ satisfies the constraints $b_j(y) = 1$. Remember that $|y| = k$. There are three cases:

- If $|z^{(j)}| = |y|$ then the constraint is satisfied by definition
- If $|z^{(j)}| > |y|$ then $z^{(j)} \not\leq y$ and so $b_j(y) = 1$
- Consider the case $|z^{(j)}| < |y|$. If $b_j(y) = 0$ then $a_{z^{(j)}}(y) = 1$ and so $z^{(j)} \leq y$. As $|z^{(j)}| < |y|$ it follows $z^{(j)} < y$. Thus it follows that $z^{(j)} \in L^*_{z^{(j)}}$ and, consequently, $b_j(z^{(j)}) = 1$ or $z^{(j)} \not\leq z^{(j)}$ which is not true. It follows that in this case we have $b_j(y) = 1$.

From this it follows that $\tilde{C}_k \subset C_k^*$. The converse is a direct consequence of the definition of the sets. □

Thus we get a variant of the apriori algorithm which checks the constraints only for one level, and moreover, this is done to reduce the number of candidate itemsets. This is Algorithm 3.

Algorithm 3 Apriori with antimonotone constraints

```
$C_1 = \mathcal{A}(\mathbb{X})$ is the set of all one-itemsets, $k = 1$
while $C_k \neq \emptyset$ do
    extract elements of $C_k$ which satisfy the constraints $a_{z^{(j)}}(x) = 0$ for $|z^{(j)}| = k$ and put into $C_k^*$
    scan database to determine support of all $y \in C_k^*$
    extract frequent itemsets from $C_k^*$ into $L_k^*$
    generate $C_{k+1}$ (as per ordinary apriori)
    $k := k + 1$.
end while
```

4.3. *Partitioned algorithms*

The previous algorithms assumed that all the data was able to fit into main memory and was resident in one place. Also, the algorithm was for one processor. We will look here into partitioned algorithms which lead to parallel, distributed and out-of-core algorithms with few synchronisation points and little disk access. The algorithms have been suggested in [15].

We assume that the data is partitioned into equal parts as

$$D = [D_1, D_2, \dots, D_p]$$

where $D_1 = (x^{(1)}, \dots, x^{(n/p)})$, $D_2 = (x^{(n/p+1)}, \dots, x^{(2n/p)})$, etc. While we assume equal distribution it is simple to generalise the discussions below to non-equal distributions.

In each partition D_j an estimate for the support $s(a)$ of a predicate can be determined and we will call this $\hat{s}_j(a)$. If $\hat{s}(a)$ is the estimate of the support in D then one has

$$\hat{s}(a) = \frac{1}{p} \sum_{j=1}^{p} \hat{s}_j(a).$$

This leads to a straight-forward parallel implementation of the apriori algorithm: The extraction of the L_k can either be done on all the processors redundantly or on one master processor and the result can then be communicated. The parallel algorithm also leads to an out-of-core algorithm which does the counting of the supports in blocks. One can equally develop an apriori-tid variant as well.

There is a disadvantage of this straight-forward approach, however. It does require many synchronisation points, respectively, many scans of the disk, one for each level. As the disks are slow and synchronisation expensive

Algorithm 4 Parallel Apriori

```
$C_1 = \mathcal{A}(\mathbb{X})$ is the set of all one-itemsets, $k = 1$
while $C_k \neq \emptyset$ do
    scan database to determine support of all $z \in C_k$ on each $D_j$ and sum up the results
    extract frequent itemsets from $C_k$ into $L_k$
    generate $C_{k+1}$
    $k := k + 1$.
end while
```

this will cost some time. We will not discuss and algorithm suggested by [15] which substantially reduces disk scans or synchronisation points at the cost of some redundant computations. First we observe that

$$\min_k \hat{s}_k(a) \leq \hat{s}(a) \leq \max_k \hat{s}_k(a)$$

which follows from the summation formula above. A consequence of this is

Theorem 20: Each a which is frequent in D is at least frequent in one D_j.

Proof: If for some frequent a this would not hold then one would get

$$\max \hat{s}_j(a) < \sigma_0$$

if σ_0 is the threshold for frequent a. By the observation above $\hat{s}(a) < \sigma_0$ which contradicts the assumption that a is frequent. □

Using this one gets an algorithm which generates in a first step frequent k-itemsets $L_{k,j}$ for each D_j and each k. This requires one scan of the data, or can be done on one processor, respectively. The union of all these frequent itemset is then used as a set of candidate itemsets and the supports of all these candidates is found in a second scan of the data. The parallel variant of the algorithm is then Algorithm 5. Note that the supports for all the levels k are collected simultaneously thus they require only two synchronisation points. Also, the apriori property holds for the C_k^p:

Proposition 21: *The sequence C_k^p satisfies the apriori property, i.e.,*

$$z \in C_k^p \quad \& \quad y \leq z \quad \Rightarrow y \in C_{|y|}^p.$$

Algorithm 5 Parallel Association Rules

```
determine the frequent $k$-itemsets $L_{k,j}$ for all $D_j$ in parallel
$C_k^p := \bigcup_{j=1}^p L_{k,j}$ and broadcast
determine supports $\hat{s}_k$ for all candidates and all partitions in parallel
collect all the supports, sum up and extract the frequent elements from $C_k^p$.
```

Proof: If $z \in C_k^p \quad \& \quad y \leq z$ then there exists a j such that $z \in L_{k,j}$. By the apriori property on D_j one has $y \in L_{|y|,j}$ and so $y \in C_{|y|}^p$. □

In order to understand the efficiency of the algorithm one needs to estimate the size of the C_k^p. In the (computationally best case, all the frequent itemsets are identified on the partitions and thus

$$C_k^p = L_{k,j} = L_k.$$

We can use any algorithm to determine the frequent itemsets on one partition, and, if we assume that the algorithm is scalable in the data size the time to determine the frequent itemsets on all processors is equal to $1/p$ of the time required to determine the frequent itemsets on one processor as the data is $1/p$ on each processor. In addition we require to reads of the data base which has an expectation of $n\lambda\tau_{Disk}/p$ where λ is the average size of the market baskets and τ_{Disk} is the time for one disk access. There is also some time required for the communication which is proportional to the size of the frequent itemsets. We will leave the further analysis which follows the same lines as our earlier analysis to the reader at this stage.

As the partition is random, one can actually get away with the determination of the supports for a small subset of C_k^p, as we only need to determine the support for a_z for which the supports have not been determined in the first scan. One may also wish to choose the minimal support σ for the first scan slightly lower in order to further reduce the amount of second scans required.

4.4. *Mining sequences*

The following is an example of how one may construct more complex structures from the market baskets. We consider here a special case of sequences, see [10,1]. Let the data be of the form

$$(x_1, \ldots, x_m)$$

where each x_i is an itemset (not a component as in our earlier notation). Examples of sequences correspond to the shopping behaviour of customers of retailers over time, or the sequence of services a patient receives over time. The focus is thus not on individual market-baskets but on the customers. We do not discuss the temporal aspects, just the sequential ones.

In defining our space of features we include the empty sequence () but not components of the sequences are 0, i.e.,

$$x_i \neq 0.$$

The rationale for this is that sequences correspond to actions which occur in some order and 0 would correspond to a non-action. We are not interested in the times when a shopper went to the store and didn't buy anything at all. Any empty component itemsets in the data will also be removed.

The sequences also have an intrinsic partial ordering

$$\mathbf{x} \leq \mathbf{y}$$

which holds for $(x_1, \ldots, x_m)$ and $(y_1, \ldots, y_k)$ when ever there is a sequence $1 \leq i_1 < i_2 < \cdots < i_m \leq k$ such that

$$x_i \leq y_{i_s}, \quad s = 1, \ldots, m.$$

One can now verify that this defines a partial order on the set of sequences introduced above. However, the set of sequences does not form a lattice as there are not necessarily unique lowest upper or greatest lower bounds. For example, the two sequences $((0,1),(1,1),(1,0))$ and $((0,1),(0,1))$ have the two (joint) upper bounds $((0,1),(1,1),(1,0),(0,1))$ and $((0,1),(0,1),(1,1),(1,0)$ which have now common lower bound which is still an upper bound for both original sequences. This makes the search for frequent itemsets somewhat harder.

Another difference is that the complexity of the mining tasks has grown considerably, with $|\mathcal{I}|$ items one has $2^{|\mathcal{I}|}$ market-baskets and thus $2^{|\mathcal{I}|m}$ different sequences of length $\leq m$. Thus it is essential to be able to deal with the computational complexity of this problem. Note in particular, that the probability of any particular sequence is going to be extremely small. However, one will be able to make statements about the support of small subsequences which correspond to shopping or treatment patterns.

Based on the ordering, the *support* of a sequence $\mathbf{x}$ is the set of all sequences larger than $\mathbf{x}$ is

$$s(\mathbf{x}) = P\left(\{\mathbf{x} | \mathbf{x} \leq y\}\right).$$

This is estimated by the number of sequences in the data base which are in the support. Note that the itemsets now occur as length 1 sequences and thus the support of the itemsets can be identified with the support of the corresponding 1 sequence. As our focus is now on sequences this is different from the support we get if we look just at the distribution of the itemsets.

The length of a sequence is the number of non-empty components. Thus we can now define an apriori algorithm as before. This would start with the determination of all the frequent 1 sequences which correspond to all the frequent itemsets. Thus the first step of the sequence mining algorithm is just the ordinary apriori algorithm. Then the apriori algorithm continues as before, where the candidate generation step is similar but now we join any two sequences which have all components identical except for the last (non-empty) one. Then one gets a sequence of length $m+1$ from two such sequences of length m by concatenating the last component of the second sequence on to the first one. After that one still needs to check if all subsequences are frequent to do some pruning.

There has been some arbitrariness in some of the choices. Alternatives choose the size of a sequence as the sum of the sizes of the itemsets. In this case the candidate generation procedure becomes slightly more complex, see [1].

4.5. *The FP tree algorithm*

The Apriori algorithm is very effective for discovering a reasonable number of small frequent itemsets. However it does show severe performance problems for the discovery of large numbers of frequent itemsets. If, for example, there are 10^6 frequent items then the set of candidate 2-itemsets contains $5 \cdot 10^{11}$ itemsets which all require testing. In addition, the Apriori algorithm has problems with the discovery of very long frequent itemsets. For the discovery of an itemset with 100 items the algorithm requires scanning the data for all the 2^{100} subsets in 100 scans. The bottleneck in the algorithm is the creation of the candidate itemsets, more precisely, the number of candidate itemsets which need to be created during the mining. The reason for this large number is that the candidate itemsets are visited in a breadth-first way.

The FP tree algorithm addresses these issues and scans the data in a depth-first way. The data is only scanned twice. In a first scan, the frequent items (or 1-itemsets) are determined. The data items are then ordered based on their frequency and the infrequent items are removed. In the second scan, the data base is mapped onto a tree structure. Except for the root all the nodes are labelled with items, each item can correspond to multiple nodes. We will explain the algorithm with the help of an example, see table 2 for the original data and the records with the frequent itemsets only (here we look for support > 0.5).

Table 2. Simple data base and data base after removal of the items with less than 50% support.

items	$s > 0.5$
f, a, c, d, g, i, m, p	f, c, a, m, p
a, b, c, f, l, m, o	f, c, a, b, m
b, f, h, j, o, w	f, b
b, c, k, s, p	c, b, p
a, f, c, e, l, p, m, n	f, c, a, m, p

Initially the tree consists only of the root. Then the first record is read and a path is attached to the root such that the node labelled with the first item of the record (items are ordered by their frequency) is adjacent to the root, the second item labels the next neighbour and so on. In addition to the item, the label also contains the number 1, see Step 1 in figure 10. Then the second record is included such that any common prefix (in the example the items f,c,a is shared with the previous record and the remaining items are added in a splitted path. The numeric parts of the labels of the shared prefix nodes are increased by one, see Step 2 in the figure. This is then done with all the other records until the whole data base is stored

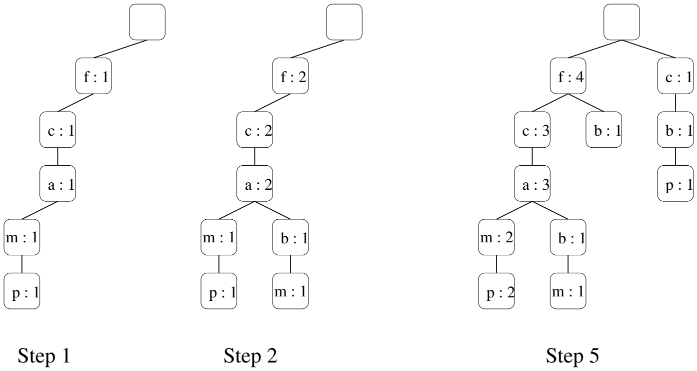


Fig. 10. Construction of the FP-Tree.

in the tree. As the most common items were ordered first, there is a big likelihood that many prefixes will be shared which results in substantial saving or compression of the data base. Note that no information is lost with respect to the supports. The FP tree structure is completed by adding a header table which contains all items together with pointers to their first occurrence in the tree. The other occurrences are then linked together so that all occurrences of an item can easily be retrieved, see figure 11.

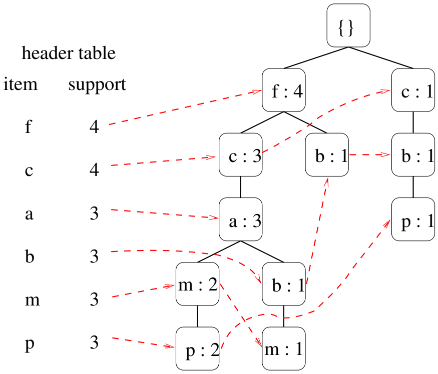


Fig. 11. Final FP-Tree.

The FP tree does never break a long pattern into smaller patterns the way the Apriori algorithm does. Long patterns can be directly retrieved from the FP tree. The FP tree also contains the full relevant information about the data base. It is compact, as all infrequent items are removed and the highly frequent items share nodes in the tree. The number of nodes is never less than the size of the data base measured in the sum of the sizes of the records but there is anecdotal evidence that compression rates can be over 100.

The FP tree is used to find all association rules containing particular items. Starting with the least frequent items, all rules containing those items can be found simply by generating for each item the conditional data base which consists for each path which contains the item of those items which are between that item and the root. (The lower items don't need to be considered, as they are considered together with other items.) These *conditional pattern bases* can then again be put into FP-trees, the *conditional FP-trees* and for those trees all the rules containing the previously selected and any other item will be extracted. If the conditional pattern base contains only one item, that item has to be the itemset. The frequencies of these itemsets can be obtained from the number labels.

An additional speed-up is obtained by mining long prefix paths separately and combine the results at the end. Of course any chain does not need to be broken into parts necessarily as all the frequent subsets, together with their frequencies are easily obtained directly.

5. Conclusion

Data mining deals with the processing of large, complex and noisy data. Robust tools are required to recover weak signals. These tools require highly efficient algorithms which scale with data size and complexity. Association rule discovery is one of the most popular and successful tools in data mining. Efficient algorithms are available. The developments in association rule discovery combine concepts and insights from probability and combinatorics. The original algorithm "Apriori" was developed in the early years of data mining and is still widely used. Numerous variants and extensions exist of which a small selection was covered in this tutorial.

The most recent work in association rules uses concepts from graph theory, formal concept analysis and statistics and links association rules with graphical models and with hidden Markov models.

In this tutorial some of the mathematical basis of association rules was covered but no attempt has been made to cover the vast literature discussing numerous algorithms.

Acknowledgements

I would like to thank Zuowei Shen, for his patience and support during the preparation of this manuscript. Much of the work has arisen in discussions and collaboration with John Maindonald, Peter Christen, Ole Nielsen, Steve Roberts and Graham Williams.

References

1. J.-M. Adamo. *Data Mining for Association Rules and Sequential Patterns.* Springer, New York, 2001.
2. R. Agrawal, T. Imielinski, and T. Swami. Mining association rules between sets of items in large databases. In *Proc., ACM SIGMOD Conf. on Manag. of Data*, pages 207–216, Washington, D.C., 1993.
3. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. Bocca, M. Jarke, and C. Zaniolo, editors, *Proc. Int.*

Conf. on Very Large Data Bases, pages 478–499, Santiago, Chile, 1994. Morgan Kaufman, San Francisco.

4. B. Bollobaś. *Combinatorics.* Cambridge University Press, 1986.
5. C. Borgelt and R. Kruse. *Graphical Models — Methods for Data Analysis and Mining.* J. Wiley & Sons, Chichester, United Kingdom, 2002.
6. B. A. Davey and H. A. Priestley. *Introduction to lattices and order.* Cambridge University Press, New York, second edition, 2002.
7. A. J. Dobson. *An introduction to generalized linear models.* Chapman & Hall/CRC Texts in Statistical Science Series. Chapman & Hall/CRC, Boca Raton, FL, second edition, 2002.
8. B. Ganter and R. Wille. *Formal Concept Analysis.* Springer, 1999.
9. F. Geerts, B. Goethals, and J. Van den Bussche. A tight upper bound on the number of candidate patterns. In *ICDM*, pages 155–162, 2001.
10. J. Han and M. Kamber. *Data Mining, Concepts and Techniques.* Morgan Kaufmann Publishers, 2001.
11. D. Heckerman. A tutorial on learning with bayesian networks. In M. Jordan, editor, *Learning in graphical models*, pages 301–354. MIT Press, 1999.
12. D. J. Schlimmer. *Congressional Quarterly Almanac, 98th Congress, 2nd session 1984*, volume XL. Congressional Quarterly Inc., 1985.
13. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences. In *Proc. 1st Int. Conf. Knowledge Discovery Databases and Data Mining*, pages 210–215, Menlo Park, Calif., 1995. AAAI Press.
14. G. McLachlan and D. Peel. *Finite mixture models.* Wiley Series in Probability and Statistics: Applied Probability and Statistics. Wiley-Interscience, New York, 2000.
15. A. Savasere, E. Omieski, and S. Navanthe. An efficient algorithm for mining association rules in large databases. In *Proc. 21st Int. Conf. Very Large Data Bases*, pages 432–444. Morgan Kaufmann, San Francisco, 1995.
16. G. Webb. Efficient search for association rules. In *Proc. of the Sixth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 99–107. ACM Press, 2000.