

Intelligent Systems Reference Library 73

Margarita N. Favorskaya
Lakhmi C. Jain *Editors*

Computer Vision in Control Systems-1

Mathematical Theory

 Springer

Intelligent Systems Reference Library

Volume 73

Series editors

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

Lakhmi C. Jain, University of Canberra, Canberra, Australia, and
University of South Australia, Adelaide, Australia
e-mail: Lakhmi.Jain@unisa.edu.au

About this Series

The aim of this series is to publish a Reference Library, including novel advances and developments in all aspects of Intelligent Systems in an easily accessible and well structured form. The series includes reference works, handbooks, compendia, textbooks, well-structured monographs, dictionaries, and encyclopedias. It contains well integrated knowledge and current information in the field of Intelligent Systems. The series covers the theory, applications, and design methods of Intelligent Systems. Virtually all disciplines such as engineering, computer science, avionics, business, e-commerce, environment, healthcare, physics and life science are included.

More information about this series at <http://www.springer.com/series/8578>

Margarita N. Favorskaya · Lakhmi C. Jain
Editors

Computer Vision in Control Systems-1

Mathematical Theory

 Springer

Editors

Margarita N. Favorskaya
Department of Informatics and Computer
Techniques
Siberian State Aerospace University
Krasnoyarsk
Russia

Lakhmi C. Jain
Faculty of Education, Science, Technology
and Mathematics
University of Canberra
Canberra
Australia

ISSN 1868-4394

ISBN 978-3-319-10652-6

DOI 10.1007/978-3-319-10653-3

ISSN 1868-4408 (electronic)

ISBN 978-3-319-10653-3 (eBook)

Library of Congress Control Number: 2014951912

Springer Cham Heidelberg New York Dordrecht London

© Springer International Publishing Switzerland 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law. The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

The mathematics underlying computer vision can be traced back to the work of A. Rosenfeld on the digital topology,¹ G. Matheron² on decidability through regionals, and J. Serra³ on mathematical morphology. Digital topology begins with the separation of the pixels in a digital image into subsets called segments and studying the basic properties of the subsets such as adjacency and connectedness. It was G. Matheron who suggested regionals as a basis for decideability and called attention to the importance of a hit-or-miss topology in the study of closed compact subsets in a bounded region in \mathbb{R}^2 . And it was J. Serra who suggested using hit-or-miss topology⁴ as a basis for mathematical morphology. In practice, we start with a small set called a structuring element A_h , used to probe the parts of a binary image. A probe is used to check whether A_h hits a subset B (the intersection $A_h \cap B$ is not empty) in an image viewed as a hyperspace (collection of closed subsets of a topological space) or whether A_h misses B ($A_h \cap B$ is empty).

Digital topology has its roots in the work by Archimedes and Apollonius in defining the locations of points in a plane by their distances from two straight lines and later by R. Descartes in defining nonnegative coordinates in the plane. Negative coordinates were introduced by I. Newton. The set theoretic view of digital images can be traced back to the work by H. Poincaré on the similarities between point-sets in a physical continuum⁵ and by F. Hausdorff on topological spaces,⁶ focusing on open and closed sets and on metrizable spaces (spaces that are homeomorphic to metric spaces). It was Hausdorff who pointed out that every point has a least one neighborhood.⁷

¹ see [1].

² see, e.g., [2].

³ see, e.g., [3].

⁴ For a detailed view of hit-or-miss topology, see [4].

⁵ see, e.g., [5].

⁶ see, e.g., [6].

⁷ For a detailed view of various types of neighborhoods in digital images, see [7].

Appropriately, this book begins with an introduction to morphological image analysis for computer vision applications by Yu.V. Vizilter, Yu.P. Pyt'ev, A.I. Chulichkov, and L.M. Mestetskiy. These authors focus on skeleton-based continuous binary morphology, a morphological pattern spectrum, and what is known as Pyt'ev morphology. The notion of a skeleton (middle set of points) of a closed region in the Euclidean plane is a locus of centers of maximum empty circles. This leads to a very interesting presentation of what is known as a discrete morphological pattern spectrum.⁸ The Pyt'ev morphology is based on vector algebra and functional analysis, whereas the Serra morphology is based on nonlinear set-theoretic (complete lattice) models.

The central motifs in this book are threefold.

1. **Mathematical morphology**

Morphological spectral patterns and Pyt'ev morphology (Yu.V. Vizilter, Yu.P. Pyt'ev, A.I. Chulichkov and L.M. Mestetskiy, Chap. 2), fuzzy morphological contour basis for image segmentation (V.L. Fox, M. Milanova, S. Al-Ali, Chap. 8).

2. **Image correspondence**

Structural image similarity based on spectral criteria (Y.S. Radchenko, A.V. Bulygin, Chap. 3), digital image correlation (R. Kountchev, R. Kountcheva, Chap. 4), recognition of digital images with geometric transforms (V. Lutsiv, Chap. 5).

3. **Image-Based Signal analysis**

Energy and phase-energy spectra in analysing interframe differences in video signals (A. Bogoslovsky, I. Zhigulina, Chap. 6), cooperative measurement using multiple visual motion sensors (S. Gepshtein, I. Tyukin, Chap. 7), digital video stabilization via motion vector separation using fuzzy set theory (M. Favorskaya, L.C. Jain, V. Buryachenko, Chap. 9), Strip-method of image transformation that entails cutting a 1-dimensional signal into n strips, forming an n -dimensional vector, mixing and superimposing image fragments on each other using a Hadamard matrices and variations of such matrices (L. Mironovsky, V. Slaev, Chap. 10), and criteria useful in estimating the efficiency of telecommunication systems (A.A. Borisenko, V.V. Kalashnikov, A.E. Goryachev, N.I. Kalashnykova, Chap. 11).

This book ably demonstrates the utility of the basic mathematical framework provided by morphology (and its underlying attention to aspects of set theoretic topology and the basic geometric structures found in digital images) as well as the utility of a variety of approaches in image correspondence detection and image-based signal analysis. The interplay of these concepts is cogently demonstrated by the contributors to this volume.

⁸ In this book, see, e.g., Fig. 2.14, page 30.

I strongly recommend this book as a concise and very original introduction to the mathematical foundations of image analysis and the practical application of the mathematics across a broad spectrum in the study of digital images.

June 2014

James F. Peters
Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, MB, Canada

and

Faculty of Arts and Sciences
Department of Mathematics
Adıyaman University
Adıyaman, Turkey

References

1. Rosenfeld A, Pfaltz JL (1968) Distance functions on digital pictures. *Pattern Recognit* 1:33–61; Rosenfeld A (1979) Digital topology. *The Amer Math Monthly* 86(8):621–630
2. Matheron G (1975) *Random sets and integral geometry*. Wiley, New York; Matheron G (1989) *Estimating and choosing. An Essay on Probability in Practice* (trans: Hasofer AM). Springer-Verlag, Berlin, pp ix + 141
3. Serra J (1976) *Lectures on image analysis by mathematical morphology*, Tokyo Noko University; Serra J (1982) *Image analysis and mathematical morphology I*, Academic Press, London-New York, pp xiv + 610
4. Beer G (1993) *Topologies on closed and closed convex sets*, Kluwer, Dordrecht
5. Peters JF, Wasilewski P (2012) Tolerance spaces: origins, theoretical aspects and applications, *Inform Sci* 185:211–225
6. Hausdorff F (1957) *Set theory* (trans: Aumann JR) AMS Chelsea Pub, 1957
7. Peters JF (2014) *Topology of digital images. Visual Pattern Discovery in Proximity Spaces*, Springer, Ch. 1

Preface

The research book is focused on the recent advances in computer vision methodologies and technical solutions using conventional and intelligent paradigms. The contemporary solutions based on advanced mathematical achievements emphasize more information and visual monitoring in natural and human environment. The real challenge of designing such observation models are to make them close to realistic visualization and interpretation of events in our world.

The book presents some of the research results from some of the most respectable researchers in the field of computer vision stressing on mathematical theory. Of the 11 chapters, the first chapter presents a brief introduction of the chapters presented in the book. Chapter 2 is on the Morphological Image Analysis for Computer Vision Applications. Chapter 3 presents techniques for Detecting the Structural Changes in Computer Vision. Chapter 4 is on Hierarchical Adaptive KL-based Transform: Algorithms and Applications. Chapter 5 is on Automatic Estimation for Parameters of Image Projective Transforms Based on Object-invariant Cores. Chapter 6 is on the Analysis of Energy for Image and Video Sequence Processing. Chapter 7 is on Optimal Measurement of Visual Motion Across Spatial and Temporal Scales. Chapter 8 presents the Analysis of Scene Using Morphological Mathematics and Fuzzy Logic. Chapter 9 is on Digital Video Stabilization in Static and Dynamic Scenes. Chapter 10 presents the Implementation of Hadamard Matrices for Image Processing. The final chapter is on A Generalized Criterion of Efficiency for Telecommunication Systems.

The book is directed to the Ph.D. students, professors, researchers and software developers working in the areas of digital video processing and computer vision technologies.

We wish to express our gratitude to the authors and reviewers for their contribution. The assistance given by the Springer-Verlag and team is acknowledged.

Russia
Australia

Margarita N. Favorskaya
Lakhmi C. Jain

Contents

1	Development of Mathematical Theory in Computer Vision	1
	Margarita N. Favorskaya and Lakhmi C. Jain	
1.1	Introduction	1
1.2	Chapters Included in the Book	2
1.3	Conclusion	7
	References	8
2	Morphological Image Analysis for Computer Vision Applications	9
	Y.V. Vizilter, Y.P. Pyt'ev, A.I. Chulichkov and L.M. Mestetskiy	
2.1	Introduction	10
2.2	Basics of Mathematical Morphology	12
2.2.1	Mathematical Morphology as a Set-Theoretic Scheme	12
2.2.2	Binary Mathematical Morphology Based on Structuring Elements	13
2.2.3	Grayscale Mathematical Morphology Based on Structuring Elements	14
2.2.4	Mathematical Morphology as a Lattice-Theoretic Scheme	17
2.2.5	Morphologies Based on Connected Filters	17
2.2.6	Morphological Skeleton	18
2.3	Skeleton-Based Continuous Binary Morphology	20
2.3.1	Skeleton of Binary Image Versus Binary Image of Skeleton	21
2.3.2	Continuous Representation of Raster Image Boundary	22
2.3.3	Polygonal Figure Skeleton	23
2.3.4	Skeleton-Based Continuous Binary Morphologies	26

- 2.4 Morphological Spectrum: Concept and Computation 28
 - 2.4.1 Pattern Spectrum and Morphological Spectra 29
 - 2.4.2 Thickness Map and Morphological Spectrum with Disk Structuring Elements 31
 - 2.4.3 Calculation of Binary Morphological Spectra Based on Continuous Skeletal Representation 32
 - 2.4.4 Calculation of Grayscale Morphological Spectra 33
- 2.5 Morphological Image Analysis (Pyt'ev Morphology) 35
 - 2.5.1 Image Shape as an Invariant of Image Transforms. 37
 - 2.5.2 Scene Recognition Based on Image Shape 40
 - 2.5.3 Scene Change Detection Based on Image Shape 41
 - 2.5.4 Scene Recognition Based on the Shape of Noisy Image 42
 - 2.5.5 Morphological Shape Matching 43
- 2.6 Projective Morphologies, Morphological Segmentation and Complexity Analysis 47
 - 2.6.1 Projective Morphologies Based on Morphological Decompositions 47
 - 2.6.2 Image Segmentation in the Framework of Projective Morphology 48
 - 2.6.3 Shape Regularization and Morphological Filters by Regularization 50
 - 2.6.4 Morphological Complexity, Filters, and Spectra by Complexity 52
- 2.7 Conclusion 53
- References 54

- 3 Methods for Detecting of Structural Changes in Computer Vision Systems 59**
 - Yury S. Radchenko and Aleksey V. Bulygin
 - 3.1 Introduction 60
 - 3.2 Pixel Structural Similarity Criteria 61
 - 3.3 Spectral Criteria of Structural Image Similarity 64
 - 3.3.1 Polynomial Transforms 65
 - 3.3.2 Discrete Transforms 68
 - 3.4 Spectral Image Variation Detection 75
 - 3.4.1 Optimal Detection Algorithm 76
 - 3.4.2 Quasi-optimal Algorithms 76
 - 3.5 Experimental Research of Structural Similarity Algorithms 77
 - 3.5.1 Practical Using of Pixel and Spectral Algorithms in Image Analysis 77
 - 3.5.2 Experimental Research of Spectral Statistics D_0 and D_E 80

3.5.3 Experimental Research of MSSIM and MNSSIM1(2) Criteria 83

3.6 Conclusion 89

References 89

4 Hierarchical Adaptive KL-Based Transform: Algorithms and Applications. 91

Roumen Kountchev and Roumiana Kountcheva

4.1 Introduction 92

4.2 Analysis of the Image Transform Methods Based on the KLT 93

4.2.1 Karhunen-Loeve Transform for Inter-frame (3D) Processing of a Group of Correlated Images 95

4.2.2 Karhunen-Loeve Transform for Intra-frame (2D) Processing of Single Images 97

4.3 Hierarchical Adaptive Karhunen-Loeve Based Transform for a Group of Images 99

4.3.1 Determination of the Length of the Group of Images, Processed with the Hierarchical Adaptive KLT 99

4.3.2 Hierarchical Adaptive KLT Based on Sub-groups of Three Images 101

4.3.3 Hierarchical Adaptive KLT Based on Sub-groups of Two Images 103

4.4 Analytical Definition of the AKLT Matrix of Size 2×2 and 3×3 103

4.4.1 Algebraic Method for Calculation of the AKLT Matrix of Size 3×3 105

4.4.2 Evaluation of the Decorrelation of the Images Transformed Through HA-KLT 3×3 108

4.4.3 Algebraic Method for Determination of the AKLT Matrix of Size 2×2 110

4.5 HA-KLT Applications for Sequences of Motion-Compensated TV Frames 112

4.6 Evaluation of the Computational Complexity of the Two-Level HA-KLT 3×3 for a Group of Images 113

4.7 Experimental Results for Group of Consecutive Images 115

4.8 Principle of the Hierarchical Adaptive KL-Based Transform for Single Image 122

4.9 Algorithm for Recursive 2D Hierarchical Adaptive KLT 126

4.10 Evaluation of the Computational Complexity of 2D HA-KLT. 128

4.11 Experimental Results for Single Images 131

4.12 Conclusion 134

References 134

5 Automatic Estimation for Parameters of Image Projective Transforms Based on Object-Invariant Cores 137
 Vadim Lutsiv

5.1 Introduction 137

5.2 Projective Transform as a Sequence of Simpler Geometric Transformations. 139

5.3 Determining the Parameters of Image Translation 141

5.4 Determining the Direction and Value of Anisotropic Scaling of Image 143

5.5 Determining the Parameters of Isotropic Scaling of Image 147

5.6 Determining the Parameters of Rotation and Specular Reflection. 148

5.7 Iterative Compensation and Estimation of Full Projective Transform 150

5.8 Experimental Results 163

5.9 Similar Results Reached by Other Authors 167

5.10 Conclusion 168

References 180

6 A Way of Energy Analysis for Image and Video Sequence Processing 183
 Andrei Bogoslovsky and Irina Zhigulina

6.1 Introduction 184

6.2 Problem Statement 185

6.3 Frequency Response 186

6.4 Frequency Responses Difference Model 188

6.5 Special Cases 192

6.6 FR-IFD Application. 193

6.7 Image Energy at Discrete Spatial Filter Output 197

6.8 Input Signal Matrix 200

6.9 Average Squared Error Minimization. 202

6.10 Quasi-Optimal Spatial Filtering 205

6.11 Canonical Form of Output Energy Functional 207

6.12 Conclusion 209

References 209

7 Optimal Measurement of Visual Motion Across Spatial and Temporal Scales. 211
 Sergei Gepshtein and Ivan Tyukin

7.1 Introduction 212

7.2 Gabor’s Uncertainty Relation in One Dimension 213

7.2.1 Single Sensors 214

7.2.2 Sensor Populations 217

7.2.3 Cooperative Measurement. 219

7.3	Gabor’s Uncertainty in Space-Time	220
7.3.1	Uncertainty in Two Dimensions	220
7.3.2	Equivalence Classes of Uncertainty	222
7.3.3	Spatiotemporal Interaction: Speed	223
7.4	Optimal Conditions for Motion Measurement	224
7.4.1	Minima of Uncertainty	225
7.4.2	The Shape of Optimal Set	226
7.5	Sensor Allocation	228
7.5.1	Adaptive Allocation	228
7.5.2	Mechanism of Adaptive Allocation	230
7.6	Conclusions	231
	References	236
8	Scene Analysis Using Morphological Mathematics and Fuzzy Logic	239
	Victoria Lynn Fox, Mariofanna Milanova and Salim Al-Ali	
8.1	Introduction	240
8.2	Background Material	241
8.2.1	Segmentation Methods	241
8.2.2	Morphological Mathematics	246
8.2.3	Fuzzy Logic	247
8.2.4	Visual Attention	248
8.3	Proposed Method	249
8.3.1	Morphological Mean Curvature	250
8.3.2	A Hybrid Morphological Contour	251
8.3.3	A Fuzzy Morphological Contour with Visual Attention	254
8.4	Conclusion	257
	References	258
9	Digital Video Stabilization in Static and Dynamic Scenes	261
	Margarita N. Favorskaya, Lakhmi C. Jain and Vladimir Buryachenko	
9.1	Introduction	262
9.2	Problem Statement	263
9.3	Related Work	264
9.4	Video Stabilization in Complex Static Scenes	269
9.4.1	Motion Estimation in Static Scene	270
9.4.2	Unwanted Motion Compensation	277
9.4.3	Static Scene Alignment	278
9.5	Video Stabilization Method in Dynamic Scenes	280
9.5.1	Scene Separation	280
9.5.2	Deblurring for Visual Objects with Complex Motion	282
9.5.3	Motion Estimation of Background and Foreground	284

- 9.5.4 Unwanted Motion Compensation 290
- 9.5.5 Motion Inpainting in Dynamic Scenes 292
- 9.6 Discussion of Experimental Results 294
 - 9.6.1 Experimental Results for Motion Estimations 299
 - 9.6.2 Experimental Results for Stabilization Estimations. 300
- 9.7 Conclusion and Future Development 304
- References. 305

- 10 Implementation of Hadamard Matrices for Image Processing 311**
 Leonid Mironovsky and Valery Slaev
 - 10.1 Introduction 312
 - 10.2 Related Work 312
 - 10.3 Strip-Method of Image Transformation. 313
 - 10.3.1 Two-Dimensional Strip-Transformation 313
 - 10.3.2 Choice of Optimal Transformation Matrices 319
 - 10.4 Hadamard Matrices and Matrices Closed to Them. 321
 - 10.4.1 Hadamard Matrices 322
 - 10.4.2 Shortened Hadamard Matrices. 324
 - 10.4.3 Conference Matrices 325
 - 10.4.4 Optimal Orthogonal Matrices of the Odd Order
 (M-Matrices). 328
 - 10.4.5 Two-, Three-, and Many-Levels M-Matrices. 333
 - 10.5 Conclusion 347
 - References. 347

- 11 A Generalized Criterion of Efficiency for Telecommunication
 Systems 351**
 Alexey A. Borisenko, Vyacheslav V. Kalashnikov,
 Alexey E. Goryachev and Nataliya I. Kalashnykova
 - 11.1 Introduction 352
 - 11.2 Economic Evaluation of Telecommunication Systems 353
 - 11.2.1 Technical-Economic Evaluation
 of Telecommunication Systems 355
 - 11.2.2 Technical Evaluation of Telecommunication
 Systems 355
 - 11.3 Problem Specification 358
 - 11.3.1 A Survey of Recent Results of Research 359
 - 11.3.2 The Unsolved Part of the Problem. 359
 - 11.3.3 The Results of This Chapter 360
 - 11.4 Replacement of a Telecommunication System
 with Its Ideal Model 360
 - 11.5 Main Results 361
 - 11.5.1 Evaluation of Efficiency of Telecommunication
 Systems 364

11.5.2	An Algorithm to Assess the Efficiency of a Telecommunication System	366
11.6	An Example of Evaluation of the Efficiency of a Telecommunication System	367
11.7	Conclusion	370
	References	371

About the Editors



Margarita N. Favorskaya received her engineering diploma from Rybinsk State Aviation Technological University, Russia, in 1980 and was awarded a Ph.D. by S.-Petersburg State University of Aerospace Instrumentation, S.-Petersburg, in 1985. Since 1986 she worked as Associate Professor of Siberian State Aerospace University, Krasnoyarsk. Margarita Favorskaya defended her doctoral dissertation in Siberian Federal University in 2011. Since 2011 she is a Professor and a Head of Department of Informatics and Computer Techniques at Siberian State Aerospace University.

Her main research interests are digital image and videos processing, pattern recognition, fractal image processing, artificial intelligence, information technologies, remote sensing. She is the author or the co-author of nearly 130 scientific publications and 20 educational manuals in these fields. Margarita Favorskaya is a member of KES organization, IPC member of International Conferences, and Co-Chair of Invited Sessions. She serves as the Reviewer, Guest Editor, and Associate Editor in International Journals.



Lakhmi C. Jain is with the Faculty of Education, Science, Technology and Mathematics at the University of Canberra, Australia and University of South Australia, Australia. He is a Fellow of the Institution of Engineers Australia.

Dr. Jain founded the KES International for providing a professional community the opportunities for publications, knowledge exchange, cooperation and teaming. Involving around 5,000 researchers drawn from universities and companies world-wide, KES facilitates international cooperation and generate synergy in teaching and research. KES regularly provides networking opportunities for professional community through one of the largest conferences of its kind in the area of KES. www.kesinternational.org.

His interests focus on the artificial intelligence paradigms and their applications in complex systems, security, e-education, e-healthcare, unmanned air vehicles and intelligent agents.

Chapter 1

Development of Mathematical Theory in Computer Vision

Margarita N. Favorskaya and Lakhmi C. Jain

Abstract This chapter presents a brief description of chapters devoted to the theoretical development of computer vision. Original investigations in mathematical morphology, estimations of structural changes, the hierarchical adaptive Karhunen-Loeve and projective transforms, among others, provide the great contribution in mathematical foundations of computer vision. Each theoretical chapter involves practical implementations, which demonstrate the merit of proposed methods in practice.

Keywords Computer vision · Image processing · Videos processing

1.1 Introduction

In the past decades, computer vision techniques have progressed significantly and are widely used in many implementations of control systems. Great advances have been made in image filtering, segmentation, pattern recognition, and events understanding. However, the excellent mathematical models and methods cannot be directly applied in many practical situations. The majority of efforts focus on designing the efficient and real-time methods to analyze images and video data on various levels of processing. The contemporary solutions based on advanced mathematical achievements emphasize on more information and visual monitoring

M.N. Favorskaya (✉)

Institute of Informatics and Telecommunications, Siberian State Aerospace University,
31 Krasnoyarsky Rabochy, Krasnoyarsk 660014, Russian Federation
e-mail: favorskaya@sibsau.ru

L.C. Jain

Faculty of Education, Science, Technology and Mathematics, University of Canberra,
Canberra, ACT 2601, Australia
e-mail: lakhmi.jain@unisa.edu.au

in natural and human environment. The goal of current investigations is designing such observation models, which are close to realistic visualization and interpretation of events in our world.

1.2 Chapters Included in the Book

The main purpose of this research book is to present a sample of research results on recent advances in computer vision. This book includes eleven chapters on the “Mathematical Theory” aspect of the computer vision.

Chapter 2 introduces the morphological framework as a very wide theoretical platform for creation of mid-level image analysis tools for specialized computer vision applications. It utilizes the structural image modeling and decides some image filtering, segmentation, and comparison problems. Mathematical Morphology (MM) by Serra [1] and Matheron [2] is still the most well-known version of MM until these days. Another morphological approach proposed by Pyt’ev is based on geometrical and algebraic reasoning. In the framework of Pyt’ev morphology, images are considered as piecewise-constant 2D functions [3]. The tessellation of image frame by a set of non-intersected connected regions with constant intensities determines the “shape” of the image. The main idea of this approach is the projection of one image onto the shape of other image. The detection of morphological changing is performed by comparison of image and its projection to the reference image. Such morphological tools are invariant relative to image intensity transforms and stable relative to noise. The idempotent operators such as morphological filters or projectors are introduced using a concept of figure filling by structured elements. In other version, morphological filters are based on merging of grayscale image connected regions (“flat” zones). The continuous binary morphology is based on computational geometry and provides very fast tool for computation of continuous figure skeletons using approximation of 2D binary image by region border polygons and calculation of Voronoi diagram for segments these polygons [4]. A skeletal representation of the figure is formed as its skeleton and the radial function determined in skeleton points. The projective morphology is a generalized framework based on Serra’s MM and Pyt’ev’s Morphological Analysis. It combines the ideas of both morphological approaches and allows construction of some new morphological systems and operators based on different image decompositions and transforms and/or criteria (energy functions). Criterion-based projective morphological filters are implemented using numeric optimization techniques (linear programming, dynamic programming, graph cutting, and so on) [5]. The morphological spectrum as a multi-scale morphological shape analysis tool based on “granulometry” also contains in this chapter.

Chapter 3 discusses the criteria of Mean Structural Similarity Index Measure (MSSIM) and the developed Mean Nonparametric Structural Similarity Index Measure (MNSSIM), as well as the spectral algorithm for detecting structural changes in a frame, which have been used to good effect in video codec analysis [6].

These criteria provide the estimations for structural (texture) variations of images. The growing popularity of these criteria is proved by their quite appropriate compliance with the human vision system [7]. The detection of variations in the image segment structure is based on spectral and correlation analysis of space-time fields. At present, the quasi-optimum heuristic algorithms applying variations of field correlation features, non-invariance of spectrum in various bases (in relation to a segment movement and change of their texture features) exist. The different estimation methods and algorithms for images presented by numbered blocks as well as the criteria and metrics being a basis to detect these differences are investigated. In this chapter, the reader can find practical examples using of pixel and spectral algorithms in image analysis.

Chapter 4 investigates a novel approach to process a single image or sequences of frames through the Hierarchical Adaptive Karhunen-Loeve (KL)-based Transform (HA-KLT). This approach is suitable for image block coding and for inter-frame processing of correlated frames in groups [8]. The basic aim of a new transform is to achieve a decorrelation of the image blocks, respectively of all frames in the processed group. This is realized by a multiple applying of the HA-KLT. After each level of the hierarchical transform, all sub-blocks (respectively groups) are rearranged so, that the components with highest correlation, which are obtained in the preceding level, would be placed in a new sub-blocks of the current level. The kernel of the multi-level transform is the Adaptive KL Transform (AKLT). The AKLT with a transform matrix of size 2×2 and 3×3 is used for the processing of the image sub-blocks and the pixels with same position in the sub-groups of frames respectively. The algebraic method for the calculation of the elements of the AKLT transform matrix of size 2×2 and 3×3 is presented in this chapter. The 2D and 3D HA-KLT algorithms for the blocks of a single image and for inter-frame processing of sequences (groups) of frames are also developed [9]. The computational complexity of these algorithms is compared with the “classic” KLT. On the one hand, the proposed approach ensures a higher accuracy of color segmentation in all cases, when a distribution of color vectors is not Gaussian. This is achieved by using a polynomial kernel for the color space expansion, after which the HA-KLT is applied to the expanded color vectors. In result, a decorrelation of the transformed vectors and an information concentration in their first components are achieved. On the other hand, this permits to reduce a number of components of the transformed vectors, retaining the first two only. In a new 2D space, the color vectors clusterization in respect to RGB space is enhanced, and they can be classified with high accuracy by using the support vector machine algorithm or other similar methods [10]. The HA-KLT method is a basis for the creation of novel efficient algorithms for a fusion of 3D images in face recognition task, an objects tracing in videos, a compression with movement compensation and without visual quality loss of TV and multi-view visual information, medical and multispectral images, etc [11].

Chapter 5 provides the design of object-invariant cores, which correspond to all types of spatially compact object images (previously segmented from a background), under the affine and projective transformations caused by an image

projection through the spherical (or almost spherical) lenses being the traditional parts of photo- and video-cameras [12]. The object-invariant core is synthesized by means of truncating the high-frequency harmonics in a spatial image spectrum. These rejected high-frequency harmonics present the object peculiarities, while the rest (extremely low-frequency) harmonics contain the information about spatial image transformations. It is shown that such object-independent core is mathematically described by elliptic paraboloid (quadratic parabola in 1D image projection). All parameters of affine geometric transformation (except a rotation and a mirror-like reflection) are measured analytically from this object invariant core. The parameters of rotation and mirror-like reflection are calculated from the cyclic narrow-band harmonic cores of image projection on the angular coordinate in a polar system. While the 6-parametric affine transformation is entirely linear, the full projective transformation contains additionally a nonlinear part described by two additional parameters. Due to this nonlinearity, the specific parameters of projective transformation cannot be measured analytically. A novel iterative optimization procedure is proposed to measure all parameters of projective transformation [13]. It is proposed to measure the missing parameters of projective transformation by a displacement of object-invariant core under the test transformations. The convergence of iterative measurement procedure is rigorously proven. At the end of the chapter, the examples of practical applications for automatic measurement of all projective transformation parameters are presented.

Chapter 6 presents a way of energy analysis for image and video sequence processing as a preliminary processing in vision systems [14]. Usually the object movements are determined by the analysis of an Inter-Frame Difference (IFD) in video signals. It is the simplest universal method. However, it doesn't exhaust opportunities for intelligent processing, especially in extremely low luminance. The IFD of energy spectrums and phase-energy spectrums are considered as an alternative analysis. The phase-energy spectrum is a product of partial derivatives in spatial phase-frequency spectrum over their spatial frequencies. It provides the detailed information about motion in finite frames [15]. The modeling of the IFD of frequency responses shows the necessity of analysis for pixels located near the moving boundaries. A processing of such pixels intensities increases a probability of movement's detection. Also distortions of moving object's shape, movement's characteristics, and a quantity of moving objects are possible to define based on the analysis of the IFD types. The phase-energy spectrums are used for edges analysis, if any movement is detected in a scene. The analysis of the energy spectrums is applied to design the effective 2D filters. The changes of the energetic indexes in static images determine the efficiency function on a whole set of impulse responses of the filter. The function of efficiency has a positively certain quadratic form with the coefficients of energy spectrum decomposition into the 2D Fourier series over the cosines. The analysis of stationary points by using this function of efficiency allows to synthesize the optimum and the quasi-optimum 2D filters. The proposed way of energy analysis provides some novel possibilities, for example, the detection of objects with extremely small contrast image.

Chapter 7 studies on how visual motion can be estimated at the lowest overall uncertainty of measurement across the entire range of useful sensor sizes (in artificial systems) [16] or the entire range of receptive fields (in biological systems). In other words, the following is an attempt to develop an economic normative theory of motion-sensitive systems. Such norms are derived for efficient design of systems, and then the norms are compared with facts of biological vision. This approach from the first principles of measurement and parsimony helps to understand the forces that shape the characteristics of biological vision. These characteristics include the spatiotemporal contrast sensitivity function, the adaptive transformations of this function caused by stimulus change, and also some characteristics of the higher-level perceptual processes such as perceptual organization [17]. In the following, the minimax strategy is implemented by assuming the maximal (worst-case) uncertainty of measurement on the sensors that span the entire range of the useful spatial and temporal scales. This strategy is used in two ways. First, the consequences of Gabor's uncertainty relation are investigated by assuming that the uncertainty of measurement is as high as possible. Second, the outcomes of measurement on different sensors are anticipated by adding their component uncertainties.

Chapter 8 presents the segmentation of natural images as a challenging task in image processing. Many methods have been proposed in the literature regarding algorithms for segmentation of such images [18]. Many of algorithms are complex in nature and inefficient in practice with unaltered images. In order to efficiently use the algorithms it is beneficial to pre-process the natural images. However, natural images often involve subjects and background that are not easily quantified with crisp pre-processing parameters. A partial solution to the problem of segmenting complex images is to use features that discriminate in the active contour algorithms [19]. These feature descriptions range from curvature to the orientation of level sets and usually result in better segmentation. An unfortunate side effect of using feature discriminates is that the complexity of the algorithm greatly increases resulting in even higher computational cost and difficulty in implementing the method. The goal is to develop a morphological level set active contour segmentation method that can robustly and efficiently segment multiphase textural images of high complexity [20]. To do this the usage of region statistics inside and outside the contour, membership functions from fuzzy logic methodology, and a Gaussian kernel function are required. In this chapter, a number of existing methods for shape feature extraction and representation are presented. At the end, application examples for using object shape representation in application for object recognition and human activity recognition are show.

Chapter 9 is devoted to digital video stabilization oriented on removal of intentional motions from video sequences caused by camera vibrations under strong wind in static scenes, by motion of robots unstable platforms in dynamic scenes, or jitters during a human hand-held shooting [21]. The analysis of dynamic scenes is required in advanced intelligent methods and directly depends from a problem statement. Several sequential stages connect with the choice of anchor frame, local and global motion estimations, and the jitters compensation algorithm. The choice of anchor frame into static scenes may be random with duration 1 s or 24 frames. In

the case of dynamic scene, the additional problem of scenes' separation should be solved for receiving a 'good' anchor frame. Most existing methods and algorithms do not work in real time. For investigation purposes, a non-real time approach is developed, however practical applications need in fast and reliable solutions. Several strategies are used for Local Motion Vectors (LMVs) building based on the keypoint detectors and block-matching algorithm [22]. The application of fuzzy logic operators improves the separation results between the unwanted motion and the real motion of rigid objects. For dynamic scenes, the kurtosis estimations are calculated and tracking curves are built in the case of small vibrations, and frame interpolation is applied, if vibrations have large values. The fuzzy model based on triangular, trapezoidal, and *S*-shape memberships partitions the LMVs concerning them to an unwanted camera motion and objects motion into a scene. The output of fuzzy logic model indicates a final reliability of matching quality by using the Takagi-Sugeno-Kang model. Such zero-order fuzzy model generates the quality index (a value in the range [0, 1]). The quality of the points matching is classified into four categories: excellent, good, medium, and bad. Therefore, fuzzy logic is used for improvement of local and global motion estimations and determines the novelty of approach. The similar procedure is applied for estimation of Global Motion Vectors (GMVs). The corrective algorithm compensates the unwanted motion into frames. Thereby, the scene is aligned. For restoration of current frame, pixels are shifted on a value of Accumulated Motion Vector (AMV) of unwanted motion. However, the sizes of stabilized frames became less relatively the original video sequence and the restoration of "missing" frame edges is required.

Chapter 10 examines the problems of transforming information and studying data connected with processing and transmitting images. The strip-method for storage and noise-immune transmission of images is studied [23]. Before transmission, the matrix transformations of an original image are executed, during which the image fragments are mixed and superimposed on each other. The transformed image is transmitted over a communication channel, where it is distorted with a pulse noise, the latter being for example a possible reason for a complete loss of separate image fragments. In the process of receiving a signal at the receiving end, an inverse transformation is performed. At the end of this transformation, the reconstruction of the image takes place. If it is possible to provide a uniform distribution of the pulse noise over the whole area the image occupies (without any changes of its energy), then a noticeable decrease of noise amplitude will take place and an acceptable quality of all fragments of the image reconstructed. In this chapter, many tasks are considered such as versions of the two-sided strip-transformation of images, choice of optimal transformation matrices, investigation of root images of the strip-transformation, and illustration of capabilities of the method suggested using particular examples. In order to get the maximum decrease of the pulse noise amplitude, it is necessary to achieve a uniform distribution of the noise over the image by applying the inverse transformation at the receiving end of the communication channel. This will allow information about distorted or "lost" fragments to be reconstructed. Now a problem of determining the type of the transformation matrices A and B arises. The solution of this problem will provide

the possibility to minimize the noise amplitude in the reconstructed image. A well-known solution of this problem is related to the cases of n , which can be divided by four, i.e. the so-called normalized Hadamard matrices [24]. The less-known solution for even n , not divisible by four, consists in so-called C -matrices (Conference-matrices). Such matrices have a zero diagonal and their remaining elements are equal to ± 1 .

Chapter 11 provides a discussion about the generalized criterion of efficiency for telecommunication systems [25]. Besides the partial criteria, there exists also a need in developing generalized ones allowing to compare various telecommunication systems and to choose the most efficient ones among them. To this end, the generalized criteria should consider and incorporate the partial ones, establish certain relationships between them, and hence possess the highest possible objectivity. Such criteria should be rather simple, easily computable, and provide the way to compare the telecommunication systems within a definite numerical scale, that is, they should be normalized [26]. The chapter develops a generalized criterion to estimate the efficiency of telecommunication systems that can be applied to economics information systems, too. The criterion combines evaluation of such special properties as the information quantity, noise immunity, the data transmission speed, and the transmission cost. In contrast to other criteria, the proposed one is non-dimensional and normalized, thus estimating a telecommunication system by means of real number between 0 and 1. The design of the developed criterion based upon the concept of conditional entropy is rather simple. It allows one to calculate the system's characteristic value with sufficient accuracy for practice, thus comparing various telecommunication systems to transfer the economic information. The generalized criterion is composed as a product of some partial criteria, which permits one to estimate the telecommunication systems not only as a whole, but also with respect to their partial characteristics, such as their productivity, reliability, and transmission cost.

1.3 Conclusion

The chapter has provided a briefly description of ten chapters with original mathematical investigations in computer vision techniques applied in advanced control systems. All included chapters involve the recent achievements in mathematical morphological theory, advanced criteria for structural similarity and the efficiency for telecommunication systems, the analysis of energy spectrums, complicated image transforms such as hierarchical adaptive Karhunen-Loeve transform and projective transform, optimal measurement of visual motion based on perception theory, intelligent methods for digital video stabilization, approaches for transmitting images based on Hadamard matrices Each chapter of the book explores experimental results, illustrating its use and applicability.

References

1. Serra J (1988) Image analysis and mathematical morphology. Theoretical advances. Academic Press, London
2. Matheron G (1975) Random sets and integral geometry. Wiley, New York
3. Pyt'ev Y (1998) Methods for morphological analysis of color images. *Pattern Recognit Image Anal* 8(4):517–531
4. Mestetskiy L (2010) Skeleton representation based on compound Bezier curves. In: 5th international conference on computer vision theory and applications (VISAPP'2010), vol 1. INSTICC Press, pp 44–51
5. Vizilter YV (2009) Design of data segmentation and data compression operators based on projective morphological decompositions. *J Comput Syst Sci Int* 48(3):415–429
6. Sheikh HR, Bovik FC (2011) Image information and visual quality. *IEEE Trans Image Process* 20(1):88–98
7. Pratt WK (2001) Digital image processing. PIKS inside, 3rd edn. Wiley, New York
8. Dony R (2001) Karhunen-Loeve transform. In: Rao K, Yip P (eds) *The transform and data compression handbook*. CRC Press, Boca Raton, London, New York, Washington, DC
9. Kountchev R, Kountcheva R (2013) Decorrelation of multispectral images, based on hierarchical adaptive PCA. *Int J WSEAS Trans Sign Proc* 3(9):120–137
10. Ivanov P, Kountchev R (2013) Hierarchical principal component analysis-based transformation of multispectral images. *Int J Reasoning-Based Intell Syst (IJRIS)* 5(4):260–273
11. Rao K, Kim D, Hwang J (2010) *Fast Fourier transforms: algorithms and applications*. Springer, Dordrecht, Heidelberg, London, New York
12. Shapiro LG, Stockman GC (2001) *Computer vision*. Prentice Hall, Upper Saddle River, New Jersey
13. Lutsiv V (2009) Method of iteratively compensating projective image distortions. *J Opt Technol* 76(7):417–422
14. Jähne B (2005) *Digital image processing*, 6th edn. Springer, Berlin
15. Xiao Z, Hou Z (2004) Phase based feature detector consistent with human visual system characteristics. *Pattern Recogn Lett* 25(10):1115–1121
16. Marr D (1982) *Vision: a computational investigation into the human representation and processing of visual information*. W. H. Freeman, San Francisco
17. Gepshtein S, Tyukin I, Kubovy M (2011) A failure of the proximity principle in the perception of motion. *Humana Mente* 17:21–34
18. Ilea DE, Whelan PF (2011) Image segmentation based on the integration of colour-texture descriptors—a review. *Pattern Recogn* 44(10):2479–2501
19. Marquez-Neila P, Baumela L, Alvarez L (2014) A morphological approach to curvature-based evolution of curves and surfaces. *IEEE Trans Pattern Anal Mach Intell* 36(1):2–17
20. Fox V, Milanova M, Al-Ali S (2013) A morphological multiphase active contour for vascular segmentation. *Int J Bioinform Biosci* 3(3):1–12
21. Battiato S, Lukac R (2008) *Video stabilization techniques*. Encyclopedia of multimedia. Springer, New York, pp 941–945
22. Rawat P, Singhai J (2011) Review of motion estimation and video stabilization techniques for hand held mobile video. *Int J Signal Image Process* 2(2):159–168
23. Mironowsky LA, Slaev VA (2011) Strip-method for image and signal transformation. De Gruyter, Berlin
24. Hadamard J (1893) Resolution d'une Question Relative aux Determinants. *Bull Sci Math ser 2* 17(1):240–246
25. Yüksel S, Başar T (2013) *Stochastic networked control systems*. Springer, New York, Heidelberg
26. Govindan M, Tang CM (2010) Information system evaluation: an ongoing measure. *Int J Business Inform Syst* 6(3):336–353

Chapter 2

Morphological Image Analysis for Computer Vision Applications

Y.V. Vizilter, Y.P. Pyt'ev, A.I. Chulichkov and L.M. Mestetskiy

Abstract Some original and novel morphological concepts and tools are presented in this chapter as well as required amount of mathematical morphological basics. The continuous binary morphology based on a computational geometry is presented as a very fast approach to shape representation via real-time computation of figures' skeletons. A skeletal representation of the figure is formed as a skeleton graph, and the radial function is determined in skeleton points. The proposed morphological spectrum is the multi-scale morphological shape description and analysis tools based on granulometry. It is shown how the tasks of change detection and shape matching in images can be solved using a morphological image analysis. The projective morphology as a generalized framework based on the mathematical morphology and the morphological image analysis provides fast and efficient solutions of morphological segmentation problem in complex images.

Keywords Mathematical morphology · Shape representation · Continuous skeleton · Morphological spectrum · Image analysis · Change detection · Image matching · Image segmentation

Y.V. Vizilter (✉)

State Research Institute of Aviation Systems, 7, Viktorenko str., Moscow 125319,
Russian Federation
e-mail: viz@gosniias.ru

Y.P. Pyt'ev · A.I. Chulichkov

M.V. Lomonosov Moscow State University, 1, Leninskie Gory, Moscow 119991,
Russian Federation
e-mail: achulichkov@gmail.ru

L.M. Mestetskiy

Moscow International University, 17, Leningradsky av., Moscow 125040,
Russian Federation
e-mail: mestlm@mail.ru

2.1 Introduction

The morphological framework is very wide theoretical platform for creation of mid-level image analysis tools for specialized computer vision applications. It utilizes the structural image modeling and is useful for some image filtering, segmentation and comparison problems. The foundation of Mathematical Morphology (MM) by Serra and Matheron was in 1960 [1], and this original version of MM with structuring elements, erosion/dilation operators, and monotonous opening/closing filters is still the most well-known version of MM until nowadays. However, the current morphological framework contains more ideas and tools than the initial MM. Some of them are just unknown for computer vision developers and engineers. The purpose of this chapter is to provide a brief sketch of some novel morphological techniques useful for different practical applications. The chapter contains the following issues:

- Basics of mathematical morphology.
- Skeleton-based continuous binary morphology.
- Morphological pattern spectrum: concepts and computation.
- Morphological image analysis (Pyt'ev morphology).
- Projective morphologies, morphological segmentation, and complexity analysis.

The MM is the most well-known morphological technique based on a set theory and (later) a lattice theory. The idempotent operators (morphological filters or projectors) are introduced using concept of figure filling by structuring elements. In other version, the morphological filters are based on merging of grayscale image connected regions (flat zones). A brief description of basic MM notions and concepts is required for understanding of following techniques.

The continuous binary morphology is a skeleton-based approach for description and analysis of figure shapes proposed and developed by Mestetskiy. It is based on a computational geometry and provides very fast tool for real-time computation of continuous figure skeletons using approximation of 2D binary image by region border polygons and calculation of Voronoi diagram to segment these polygons. A skeletal representation of the figure is formed as a skeleton graph, and the radial function is determined in skeleton points. The computational efficiency of such approach is based on the fact that skeleton-based continuous binary morphology uses the finite and relatively small number of analytical structuring elements for representation of binary image shape. Each analytical structuring element is connected with one edge of continuous skeleton.

A morphological spectrum is a multi-scale morphological shape description and analysis tool based on granulometry—a set of filters with different grades. Each of filters provides the details of certain size and shape to pass. The original “Pattern spectrum” proposed by Maragos is based on the Serra MM filters and describes the distribution of local figure thickness. Many modifications and generalizations of this idea are known and utilized now. In this chapter, the fast algorithm for pattern

spectrum calculation using the continuous binary skeletons is described. Such implementation allows to apply the morphological spectra in the real-time machine vision systems.

The Morphological Image Analysis (MIA) proposed by Pyt'ev is well-known in Russia since 1970. It is based on geometrical and algebraic reasoning. In the framework of Pyt'ev morphology, images are considered as piecewise-constant 2D functions. The tessellation of image frame by a set of the non-intersected connected regions with constant intensities determines the "shape" of the image. From mathematical point of view, any shape is a hyperplane in a linear space of images. The crucial idea of this approach is the projection of one image onto the shape of other image. Here a morphological image comparison is performed using the normalized morphological correlation coefficients. The morphological change detection is performed by a comparison of an image and its projection to the reference image. Such morphological tools are invariant relative to transforms of image intensity and stable relative to noise. In this chapter, a morphological shape matching technique is described that generalizes a morphological approach to shape-to-shape comparison.

A projective morphology is a generalized framework based on the Serra mathematical morphology, the Pavel shape theory, and the Pyt'ev morphological analysis. It combines ideas of these morphological approaches and allows to construct some new morphological systems and operators based on different image decompositions and transforms and/or criteria (energy functions). The criterion-based projective morphological filters are implemented using numeric optimization techniques (linear programming, dynamic programming, graph cutting, and so on). The use of morphological shape complexity as a criterion for shape regularization provides tools for shape complexity analysis those are more general than tools based on the MM granulometry concept. In particular, the definitions of the morphological filters and the morphological spectra by complexity are given.

Thus, some original and modern morphological concepts and tools are presented in this chapter as well as required amount of morphological basics. From one hand, this material allows to learn of modern morphology techniques without any previous background in the MM. From the other hand, some tools and techniques those are applicable for real-time technical vision systems, especially for vision systems of moving vehicles and other controlled real-time technical devices with video cameras or several imaging sensors, are selected and presented.

The chapter is organized as follows. The basics of mathematical morphology are discussed in Sect. 2.2. The skeleton-based continuous binary morphology is described in Sect. 2.3. The concept and computation of morphological spectrum are represented in Sect. 2.4. The morphological image analysis (Pyt'ev morphology) is given in Sect. 2.5. Section 2.6 describes the projective morphologies, a morphological segmentation and a complexity analysis. Conclusion is situated in Sect. 2.7.

2.2 Basics of Mathematical Morphology

The MM is a well-known theoretical framework for image processing and shape analysis. It was originally developed for binary image processing, and classically stated in the set-theoretical terms. Then the MM was extended to grayscale images, color images, graphs, among others. At present, the description of the MM in terms of complete lattices is the widest MM theoretical formalism.

The MM was originally developed by Serra [1] in 1964 and Matheron [2] in 1975. In 1960–1970, the set of popular MM operators was proposed including Hit-or-miss transform, dilation, erosion, opening, closing, granulometry, thinning, skeletonization, ultimate erosion, etc. In 1970–1980, some novel MM operators like morphological gradients, top-hat transform, and the watershed were proposed. In 1986, the MM generalization based on complete lattices was proposed by Serra. In 1990–2000, some further theoretical advancement was developed including the concepts of connection and leveling.

The basic MM concepts and operations in order to explore the interconnections between well-known morphological tools and some novel morphological concepts and ideas are briefly introduced in Sect. 2.2.1. A binary morphology and a grayscale mathematical morphology based on structuring elements are introduced in Sects. 2.2.2 and 2.2.3, respectively. The mathematical morphology as a lattice-theoretic scheme is discussed in Sect. 2.2.4. The novel morphological concept based on connected filters is given in Sect. 2.2.5. A building of morphological skeleton is presented in Sect. 2.2.6.

2.2.1 Mathematical Morphology as a Set-Theoretic Scheme

In a set-theoretic terms [1] the MM operations are defined for any Euclidean space E^N equipped by the set-theoretic inclusion (\subset), union (\cup), and intersection (\cap). Any operator (transform) of this space $\Psi: E^N \rightarrow E^N$ is called:

- The increasing, if it preserves the inclusion $(X \subset Y) \Rightarrow (\Psi(X) \subset \Psi(Y))$, $X, Y \subset E^N$.
- The dilation, if it preserves the union $\Psi(\cup X_i) = \cup \Psi(X_i)$, $\forall X_i \subset E^N$.
- The erosion, if it preserves the intersection $\Psi(\cap X_i) = \cap (\Psi(X_i))$, $\forall X_i \subset E^N$.
- The extensive, if $\Psi(X) \supseteq X$, and anti-extensive, if $\Psi(X) \subseteq X$.
- The idempotent (or algebraic projector), if $(\Psi(\Psi(X))) = \Psi(X)$.

All inclusion-preserving operators are called the morphological operators. Well-known Matheron theorem states that any morphological operator can be represented as a union of erosions or as an intersection of dilations. Due to this theorem, the erosion and the dilation are called the basic morphological operators.

The idempotent morphological operators are called the morphological filters:

- The anti-extensive morphological filter is called the opening.
- The extensive morphological filter is called the closing.

Let us consider the original implementation of these terms and notions.

2.2.2 Binary Mathematical Morphology Based on Structuring Elements

The classic implementation of the formal scheme applying to binary image processing is given in [1]. It is called the binary mathematical morphology. In the framework of this MM, a binary image is considered as a set of non-zero points of the plane $P = R^2$.

Let us define the translation of the set $X \subset P$ by the vector $z \in P$ as the transformation $X_z = \{y|y = x + z, x \in X\}$, where the points of the plane are summed as vectors (coordinates are added component-wise). Let $X, B \subset P$, where $b \in B$ is a Structuring Element (SE). The operation represented by Eq. 2.1 is called the Minkowski addition.

$$X \oplus B = \{x + b|x \in X, b \in B\} = \cup\{B_x|x \in X\} = \cup\{X_b|b \in B\} \quad (2.1)$$

The operation provided by Eq. 2.2 is called the Minkowski subtraction.

$$X \ominus B = \{z \in P|B_z \subseteq X\} \quad (2.2)$$

In the framework of MM these operations (Eqs. 2.1–2.2) are called the dilatation and the erosion of an image X with structuring element B . The dilatation and the erosion are the basic MM operations (see Fig. 2.1). If a structuring element B has a central symmetry relative to the origin of P , then the erosion of binary image corresponds to dilation of figure background (set-theoretic completion) and vice versa (Eq. 2.3).

$$X \ominus B = (X^C \oplus B)^C, X \oplus B = (X^C \ominus B)^C \quad (2.3)$$

The combined operation “dilatation after erosion” is called the opening of X by B (Eq. 2.4).

$$X \circ B = (X \ominus B) \oplus B \quad (2.4)$$

It has a clear geometrical meaning—the union of all structuring elements of shape B completely included in figure X : be the following expression:

$$X \circ B = \{B_z|B_z \subseteq X\}.$$

The combined operation “erosion after dilatation” is called closing of X by B (Eq. 2.5).

$$X \bullet B = (X \oplus B) \ominus B \quad (2.5)$$

If a structuring element B has a central symmetry relative to the origin of P , then the opening of binary image corresponds to closing of figure background and vice versa as it shown in Eq. 2.6.

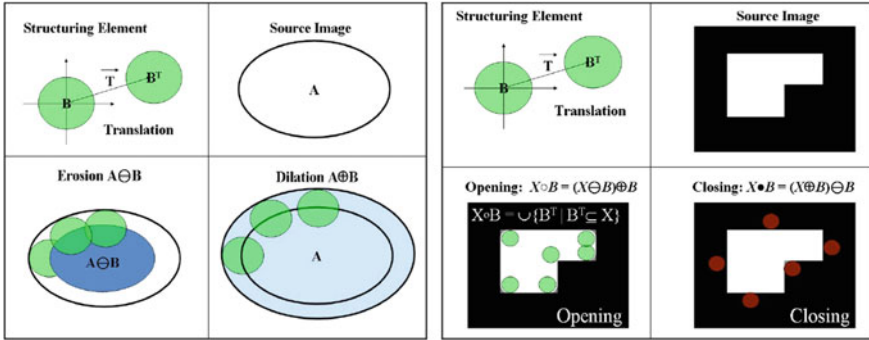


Fig. 2.1 Basic operators and filters of binary MM based on SE

$$X \circ B = (X^C \bullet B)^C, X \bullet B = (X^C \circ B)^C \quad (2.6)$$

If a structuring element B is fixed, then such opening and closing operators are idempotent (projectors) and inclusion-preserving. These operators are called the morphological filters of binary images based on the SEs.

These basic operations and filters of MM were proposed for providing the mathematically founded tools for solution of different practical shape analysis problems. For example, the task for detail extraction of figure based on the expected size and shape can be solved in a way demonstrated in Fig. 2.2.

2.2.3 Grayscale Mathematical Morphology Based on Structuring Elements

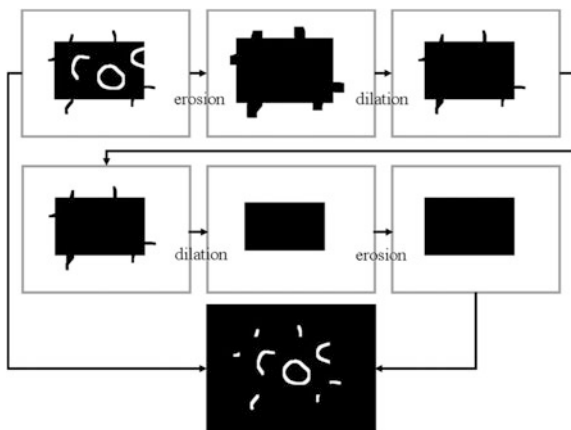
In the framework of grayscale morphology [3–6] the image $f(x, y)$ and a grayscale structuring element $k(u, v)$ are usually represented as the nonnegative two-dimensional functions determined on the plane $P = R^2$ or some square frame $F \subseteq P$. In the simplest way, the grayscale erosion and the grayscale dilation can be determined by Eqs. 2.7–2.8, respectively.

$$f(x, y) \oplus k(u, v) = \max_{(u,v)} \{f(x - u, y - v) + k(u, v)\} \quad (2.7)$$

$$f(x, y) \ominus k(u, v) = \min_{(u,v)} \{f(x + u, y + v) - k(u, v)\} \quad (2.8)$$

The grayscale morphological filters—grayscale opening and grayscale closing are defined as the combinations of grayscale erosion and grayscale dilation provided by Eqs. 2.9–2.10.

Fig. 2.2 The detail extraction using basic morphological operations



$$f \circ k = (f \ominus k) \oplus k, \quad (2.9)$$

$$f \bullet k = (f \oplus k) \ominus k \quad (2.10)$$

The following systematic formulation of grayscale morphology can be given in terms of “image umbra” [3]. Let the set of image f values in frame F be denoted as E . Then umbra of f is a set of 3D points $U(f) \subset F \times E$ lying “under the f ”:

$$U(f) = \{(x, y) \in F \times E | y \leq f(x)\}.$$

The top surface of the set $A \subseteq F \times E$ is a set $T(A): F \rightarrow E$ defined as follows:

$$T[A](x) = \max\{y | (x, y) \in A\}.$$

Based on these notions, one can define the grayscale morphological operations in usual set-theoretic way as described above binary MM operators, but for 3D ($F \times E$) space point sets. Let $F, K \subseteq E^2$, $f: F \rightarrow E$, $k: K \rightarrow E$. Then

- The dilation of f by k SE is $f \oplus k = T[U(f) \oplus U(k)]$.
- The erosion of f by k SE is $f \ominus k = T[U(f) \ominus U(k)]$.

These definitions preserve the set-theoretic background, but in computational sense they are equivalent to previous ones given in terms of min and max operations on pixels of grayscale images.

Another useful way for definition and computation of grayscale morphological operators is based on the notion of grayscale image level sets. This approach was proposed in [7]. For simplicity, let us consider the case of “flat” structuring element with two levels of intensity $k(u, v) \in \{0, -\infty\}$ that corresponds to binary SE $b(u, v) = \{(u, v) | k(u, v) = 0\}$, but for general case of grayscale SE such construction can be defined too.

Let the level image or a slice of grayscale image $f(x, y)$ at intensity level l be a binary image $f_l(x, y) = \{1: f(x, y) \geq l; 0: f(x, y) < l\}$. As it was proved in [7], in the case of “flat” SE all operators of Serra grayscale morphology can be represented as a combination of corresponding binary morphological operators applied to each level of level set as it shown in Eqs. 2.11–2.14.

$$f(x, y) \oplus k(u, v) = \max_{l \in E} \{l \times (f_l(x, y) \oplus b(u, v))\} \quad (2.11)$$

$$f(x, y) \ominus k(u, v) = \max_{l \in E} \{l \times (f_l(x, y) \ominus b(u, v))\} \quad (2.12)$$

$$f(x, y) \circ k(u, v) = \max_{l \in E} \{l \times (f_l(x, y) \circ b(u, v))\} \quad (2.13)$$

$$f(x, y) \bullet k(u, v) = \max_{l \in E} \{l \times (f_l(x, y) \bullet b(u, v))\} \quad (2.14)$$

If a digital image has a fixed and relatively small number of discrete gray levels ($E = 0, \dots, N - 1$), then this approach based on level sets provides the computationally efficient implementation of grayscale morphological operators. Figure 2.3 demonstrates the grayscale morphological operations and morphological detail extraction via corresponding background normalization (difference of source image and morphological filter).

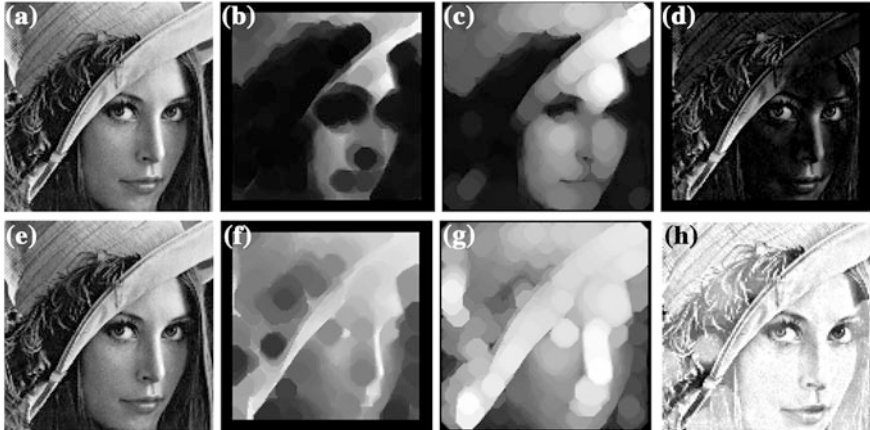


Fig. 2.3 An example of grayscale morphological operators: **a** the source image, **b** the erosion, **c** the opening, **d** the background normalization by the opening (bright details are extracted), **e** the dilatation, **f** the connected closing, **g** the background normalization by the closing (dark details are extracted)

2.2.4 Mathematical Morphology as a Lattice-Theoretic Scheme

A complete lattice [8] (L, \leq) is a set L with partial order relation (\leq) such that any subset of this set have the least element (infimum symbolized by \wedge) and the greatest element (supremum symbolized by \vee) with respect to this relation. The least element of lattice is denoted as \emptyset . The supremum of lattice U is called the “universe” of this lattice. Let us note that any pair of elements should have the infimum and supremum. For example, if lattice L is a set of subsets of some set S and ordering relation (\leq) is a set-theoretic inclusion (\subseteq) , then for any $A, B \in S$ (Eq. 2.15).

$$A \wedge B = A \cap B \quad A \vee B = A \cup B \quad (2.15)$$

Let $L = (\{X_i\}, \leq)$. The dilation is an operator $\delta: L \rightarrow L$ and the erosion is an operator $\varepsilon: L \rightarrow L$ [3] provided by Eqs. 2.16–2.17, respectively [9].

$$\bigvee_i \delta(X_i) = \delta(\bigvee_i X_i) \quad \delta(\emptyset) = \emptyset \quad (2.16)$$

$$\bigwedge_i \varepsilon(X_i) = \varepsilon(\bigwedge_i X_i) \quad \varepsilon(U) = U \quad (2.17)$$

For every dilation δ , there is one erosion ε such that

$$X \leq \varepsilon(Y) \Leftrightarrow \delta(X) \leq Y \quad \text{for all } X, Y \in L$$

and vice versa: every erosion have the dilation satisfying the above condition. Moreover, if δ and ε satisfy this condition, then they must be the dilation and erosion. Such pairs of connected operations are called the adjunctions. For every adjunction (ε, δ) , the morphological opening and the morphological closing are defined by Eqs. 2.18–2.19, respectively.

$$\gamma: L \rightarrow L: \gamma = \delta \varepsilon \quad (2.18)$$

$$\phi: L \rightarrow L: \phi = \varepsilon \delta \quad (2.19)$$

2.2.5 Morphologies Based on Connected Filters

The connected filters are the connectivity preserving morphological filters [8, 10, 11]. The image domain (frame) can be partitioned into the disjoint sets based on connected components (in the binary case) or the connected zones of constant grey/color levels (in the greyscale/color case). A connected filter works by merging the disjoint sets in the partition and assigning new grey levels or colors to them. This means that no new edges are introduced by the connected filters. A connected

filtering operates with image regions rather than pixels. Obviously, such filtering depends on a definition of connectivity (8 or 4 neighbors for pixel).

The grayscale connected operators act by merging of flat zones (regions of constant intensity). A partition is a set of nonoverlapping, nonvoid regions that fills the entire space. Let \mathbf{P} is the partition of the frame, $P(n)$ is a region of \mathbf{P} that contains pixel n . A partial order relationship of complexity among partitions can be defined as follows: \mathbf{P}_1 is finer than \mathbf{P}_2 ($\mathbf{P}_1 \subseteq \mathbf{P}_2$), if $\forall n: P_1(n) \subseteq P_2(n)$.

The set of flat zones of image f is a partition of space P_f . A grayscale operator ψ is connected, if the partition of flat zones of its input f is always finer than the partition of flat zones of its output $\psi(f)$, that is $P_f \subseteq P_{\psi(f)}$, $\forall f$.

The connected filters bridge the gap between filtering and segmentation. Figure 2.4 demonstrates an image and its filtered version with corresponding partition. This example illustrates the simplification of the image shape by the connected filter.

There are three types of connected filters:

- The filters by reconstruction perform the reconstruction of connected regions (flat zones) after their processing by some ant-extensive operator, for example, some SE erosion.
- The area filters delete regions (flat zones) with area lower than some threshold.
- The attribute filters delete regions (flat zones) with some region attributes (area, perimeter, diameter, inertia moment, etc.) non-satisfying the predefined rules (for example, lower or higher than some threshold). The area filters and filters by reconstruction are the particular cases of attribute filters.

The connected and the SE based filters complement each other in the shape analysis applications providing different tools for morphological image transformation and object selection. Such transformation means the modification of image connectivity and shape. The selection presumes an elimination or an extraction (using the background normalization scheme) of image elements with given shape, size, and contrast sign. These properties of filters are contradictive, and should be separated. Connected filters provide the solution of this problem. An example of morphological corner detection via combination of opening and connected opening is shown in Fig. 2.5. Figure 2.6 provides an example of grayscale connected morphological filters—opening and closing by reconstruction [12].

2.2.6 Morphological Skeleton

A morphological skeleton is a compact description of 2D figure shape that can be obtained by some sequence of morphological operations and provides the possibility for reconstruction of described shape using some other sequence of morphological operations. The morphological formula for the skeleton of a continuous binary image was proposed in [1, 13]. For discrete case this formula is implemented as follows. Let $\{nB\}$, $n = 0, 1, \dots$, be a sequence of shapes based on some structuring element B , $nB = B \oplus \dots \oplus B$ (n times), and $OB = \{o\}$, where o is the



Fig. 2.4 The simplification of image by the connected filter: **a** the source grayscale image, **b** the image filtered by connected version, **c** corresponding frame partition

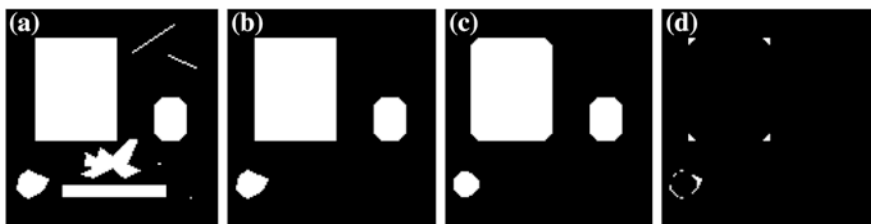


Fig. 2.5 An example of morphological corner detection: **a** the source image, **b** the result of connected opening by SE and reconstruction, **c** the result of SE opening, **d** the result of the corner extraction operator (background normalization—difference between connected opening and SE opening)

origin of 2D image plane, n is a size of the structuring element nB . A discrete skeleton $S(X)$ of a discrete binary image $X \subset Z^2$ (Fig. 2.7a, b) is the union of the skeleton subsets $\{S_n(X)\}$, $n = 0, 1, \dots, N$ provided by Eq. 2.20.

$$S_n(X) = (X \oplus nB) - (X \oplus nB) \circ B \quad (2.20)$$

The reconstruction of original shape X from the skeleton is performed by Eq. 2.21 while the partial reconstruction corresponds to the opening of original shape is represented by Eq. 2.22.

$$X = \bigcup_n (S_n(X) \oplus nB) \quad (2.21)$$

$$\bigcup_{n \geq m} (S_n(X) \oplus nB) = X \circ mB \quad (2.22)$$

In the next Sect. 2.3, another approach for definition and computation of continuous skeletons based on concept of medial axes as a set of centers of maximal inscribed balls (discs) will be considered.

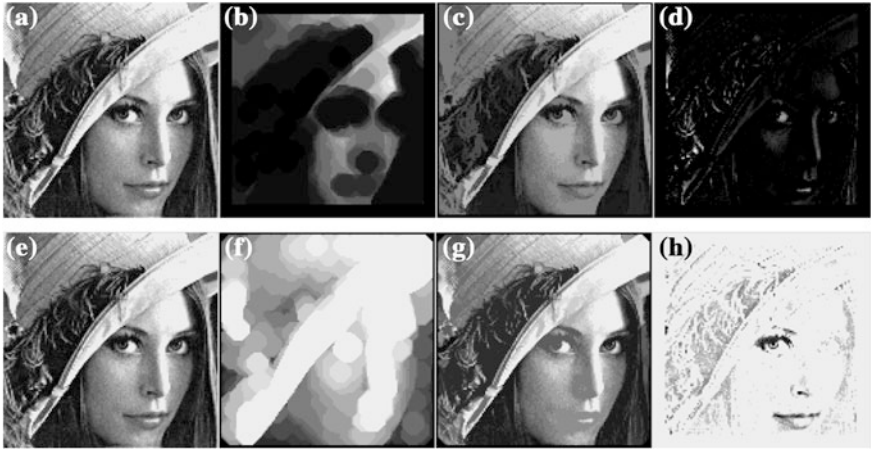


Fig. 2.6 An example of grayscale connected morphology operators: **a** the source image, **b** the erosion, **c** the connected opening, **d** the background normalization by the connected opening, **e** the dilation, **f** the connected closing, **g** the background normalization by the connected closing

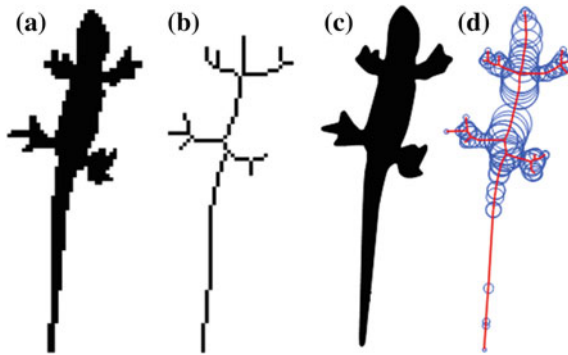


Fig. 2.7 The discrete approach of skeleton building: **a** the discrete binary image, **b** the discrete skeleton, **c** the continuous binary image, **d** the continuous skeleton and inscribed circles

2.3 Skeleton-Based Continuous Binary Morphology

This section is devoted to skeleton-based continuous binary morphology. The main concepts for the skeleton of binary images are presented in Sect. 2.3.1. The continuous representation of raster image boundary is discussed in Sect. 2.3.2. A polygonal figure skeleton based on the Delaunay graph is described in Sect. 2.3.3. Section 2.3.4 provides the novel skeleton-based continuous binary morphologies.

2.3.1 *Skeleton of Binary Image Versus Binary Image of Skeleton*

The skeleton (or medial axis representation) is a powerful and widely used tool for image shape analysis [14]. The concept of the skeleton (the middle set of points) was introduced and investigated by Blum [15]. The skeleton of a closed region in Euclidean plane is a locus of centers of maximum empty circles in this region. The circle is considered to be empty, if all its internal points are internal points of the region. One can formulate two approaches to extend the concept of the skeleton to discrete images.

The first approach, which is the most popular, will be called discrete. It consists in a morphological transformation of the original image (Fig. 2.7a) and a construction of a new image (Fig. 2.7b), which can be regarded as a skeleton. In this new bitmap, a medial axis is represented by discrete lines one pixel width. One can say that the resulting image is a digital image of the skeleton. The discrete approach is implemented in different ways: based on distance maps, thinning, Voronoi diagrams of boundary points [16, 17]. The main advantage of the discrete approach is the simplicity of the algorithm and a graphic visualization of the skeleton in the source raster format (Fig. 2.7a, b).

Another approach, which is called continuous, is based on the approximation of a discrete object by the geometrical figure in terms of a continuous geometry (Fig. 2.7c) and the construction of the skeleton for this figure (Fig. 2.7d). The resulting skeleton is considered as a continuous skeleton of discrete objects.

A continuous approach has its advantages. The main advantage of the approach is the continuous medial representation of the object's shape [17] as a geometrical graph with a radial function, which determines the width of the object. The radial function sets at each point of the geometrical graph radius of the inscribed circle are centered at this point. A continuous medial representation allows the use of graph theory and computational geometry algorithms for image shape analysis and recognition.

A comparative analysis shows the advantages of continuous skeleton compared to discrete one. These advantages are mathematical rigor, information content, and computational efficiency. This section is based on the studies described in papers and books [18–23] and provides the full implementation of the continuous approach to the skeleton construction for binary raster images of any complexity. Here an original method for continuous skeleton representation as a planar graph, whose edges are segments of straight lines and quadratic parabolas, is represented. The concept of continuous skeleton for raster binary image is developed to use the correct and elegant model of Voronoi diagram of line segments and polygonal figures to obtain the skeleton of a discrete object.

The proposed idea consists of three parts (Fig. 2.8):

- The approximation of binary image (Fig. 2.8a) by a polygonal figure (Fig. 2.8b).
- The computation of Voronoi diagram of obtained set of figures by methods of computational geometry (Fig. 2.8c).

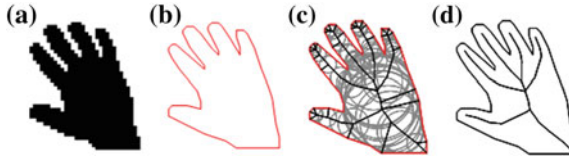


Fig. 2.8 The continuous approach: **a** the source binary image, **b** the polygonal approximation, **c** the medial representation of the polygon, **d** the resultant skeleton after pruning

- The obtaining of the skeleton from the Voronoi diagram in a convenient format for further analysis (Fig. 2.8d).

The choice of a polygonal figure for the approximation is explained by the fact that the skeleton of a polygonal figure is a fairly simple structure and can be obtained from the Voronoi diagram of this figure. The implementation of this approach required solving of several computational problems addressed below.

2.3.2 Continuous Representation of Raster Image Boundary

Let the object points of binary image be black, and the background points be white. Let us define an adjacency structure on a set of pixels as follows. For a black pair of pixels, the neighborhood of 8-adjacency and, for a white pair and black-white pair, the neighborhood of 4-adjacency are defined. A set of one-colored pixels is called connected, if, for each pair of pixels into it, there is a path from one pixel to another consisting of sequentially neighboring pixels of the same color. The maximal connected set of pixels of one color is called a connected component. The discrete figures are the connected black-colored components.

Let us call a pair of 4-adjacent black-white points as a boundary pair, and a segment connecting these points as a boundary segment. Two components, to which points of a boundary pair belong, are called adjacent, and the boundary pair is called a dividing for these components. A set of all dividing boundary pairs for two adjacent components let us call a boundary corridor. Each discrete figure defines one or more boundary corridors.

There is a minimal length path among all closed paths lying in a boundary corridor. This path is a closed polyline called a separating Minimal Perimeter Polygon (MPP). The set of all MPP of a discrete figure defines a polygonal figure (polygon with polygonal holes). Thus, we have defined the polygonal figures with minimal perimeters, which approximate the discrete figures in a binary image. The set of approximating polygonal figures exists and unique for any binary image.

The line scanning of image rows is used for tracing of all contours. Such tracing consists of detection the first boundary pair and sequential finding the next boundary pairs until the returning to start position. A process ends, when a line scanning of image rows is completed, and all contours are extracted.

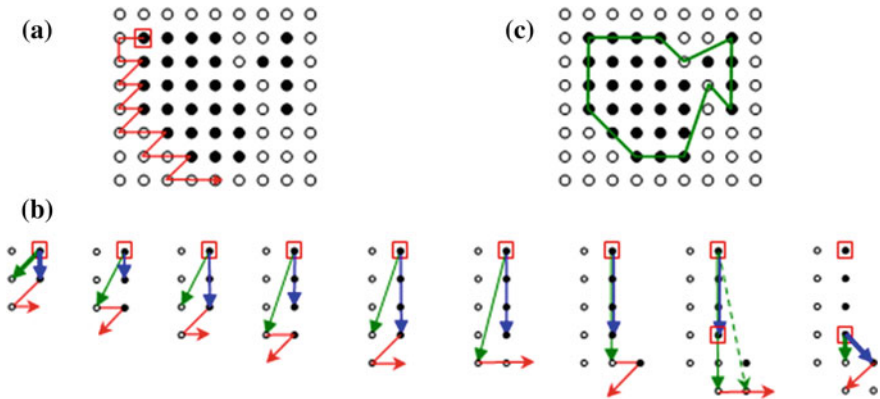


Fig. 2.9 Detection of next corner point for the minimal perimeter polygon: **a** the initial corner point, **b** the obtaining of next corner point by sequential steps for the correction of the coverage sector, **c** the minimal perimeter polygon

The sequence of contour points forms an ordered list called a tracing track. All vertices of MPP are points of a tracing track. Let us call such point a corner. Figure 2.9 illustrates the process of the MPP constructing. Figure 2.9a shows the first corner point and a part of the tracing track. Figure 2.9b presents the sequential steps of coverage sector correction from starting position (left image) until the next corner is obtained (right image). Figure 2.9c shows the final obtained MPP.

2.3.3 Polygonal Figure Skeleton

The Voronoi diagram and the Delaunay graph of polygonal figure. Let $P \in R^2$ be a multiple-connected polygonal domain and $S(P)$ be a set of all sites of P ($S(P)$ consists of all vertices and sides of P called vertex-sites and segment-sites, respectively). The medial axis $M(P)$ is a subset of Voronoi diagram $VD(P)$ of the site set of P . This is the feature most algorithms computing medial axis of a polygonal figure rely on.

The concept of the Voronoi diagram for line segments is commonly used for a skeleton construction of a polygonal figure [24, 25]. The polygonal figure boundary is a union of linear segments and vertices, which are considered as the Voronoi sites. The Voronoi diagram of these sites is generated, and the skeleton is extracted as a subset of the diagram. The skeleton of a polygonal figure with n sides can be obtained from the Voronoi diagram taking $O(n)$ time. By-turn, there are known the effective $O(n \log n)$ algorithms to construct the Voronoi diagram for the general set of linear segments [26, 27] as well as for the sides of a simple polygon [28] or multiple-connected polygonal figures [22].

A polygonal figure skeleton looks like a planar graph with edges consisting of line segments and parabolas [29, 30]. Skeleton vertices are convex vertices of a polygonal figure (one degree vertices) and also points—centers of inscribed circles tangent to figure boundary in three or more points (three and more degree vertices). The radial function is defined in each skeleton point as the radius of inscribed circle centered in this point. It is especially necessary to note that the polygonal figure skeleton always exists and is unique.

The construction of skeleton based on the Delaunay graph. The main idea of the algorithm is based on constructing the Delaunay graph of figure sites and an adjacency tree of figure boundary contours (Fig. 2.10).

Let P be a multiple-connected polygonal domain. Two sites of P are adjacent, if they are adjoining to each other (vertex-site and segment-site of a common side) or there exists a disk inscribed in P and touching both sites (in other words, if their Voronoi cells have non-empty intersection). The Delaunay graph $DG(P)$ of P is a graph (S, E_S) , where S is a set of sites of P and $E_S \subseteq S \times S$ contains all pairs of adjacent sites from S . Similarly, the Delaunay graph of a subset S' of S is defined as a graph $(S', E_{S'})$, where $S' \subseteq S$ and $E_{S'} \subseteq S' \times S'$ contains all pairs of adjacent sites from S' .

The Delaunay graph of P is a dual structure for the Voronoi diagram of P . The duality of $DG(P)$ and $VD(P)$ is that there a bijection between their structure elements exists: every $VD(P)$ vertex with its incident edges and incident faces is assigned to $DG(P)$ face with its incident edges and incident vertices. Given $DG(P)$ constructed, the $VD(P)$ can be obtained in $O(n)$ time.

The algorithm to transform the Delaunay graph to the corresponding Voronoi diagram is described in [31]. Actually, under the non-degeneracy assumption that no point in the plane is equidistant to more than three sites of P , the Delaunay graph is the triangulation of a set of the sites of P (possibly, with non-straight edges).

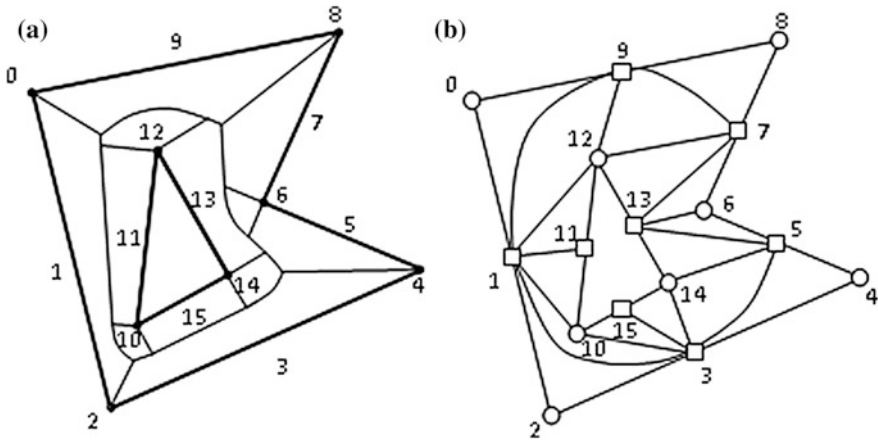


Fig. 2.10 A polygonal domain site set: **a** the Voronoi diagram, **b** the Delaunay graph (circles denote vertex-sites and squares denote segment-sites)

The algorithm to compute the Delaunay graph of a simple polygon involves the steps mentioned below:

1. The generating initial elementary chains of sites and constructing their Delaunay graphs.
2. The iterative pairwise merge of Delaunay graphs of chains. At each iteration the following operations are performed as follows:
 - The clearing Delaunay graphs in every pair so that the Delaunay condition is satisfied for every edge relative to the union of sites from both graphs in the pair. At this stage some edges can be eliminated in every pair of the Delaunay graphs.
 - The “sewing” of the Delaunay graphs in every pair so that the Delaunay condition is satisfied for every new inserted edge relative to the union of sites from both graphs in the pair. At this stage, every pair of the Delaunay graphs is “sewed” and new Delaunay graphs are formed.

The computational complexity of the algorithm is $O(n \log n)$. This algorithm is similar to one proposed in [29] for merging Delaunay triangulations.

A skeleton of multiple-connected polygonal figure. Fast algorithms for constructing skeleton of simple polygon with n vertices through Voronoi diagram have a computational complexity $O(n \log n)$ in the worst case [28]. The known generalizations to the case of a polygonal figure with holes [32–34] have computational complexity $O(kn + n \log n)$, where k is a quantity of polygonal holes, n is a general number of vertices. Such computational complexity leads to high costs in time. For example, in the problem of construction of an external skeleton for segmentation of the text document image [18] values k and n have an order 10^3 and 10^5 , respectively. At the same time, the efficient algorithms for Voronoi diagram construction of linear segment set [26, 27] do not use specific features of segment set of polygonal figure boundary owing to their universality. In particular, these algorithms build Voronoi partitioning not only inside, but also outside of a polygonal figure, and this is a superfluous work.

This solution is based on concept definition for adjacency of polygonal figure boundary contours and on construction of so-called adjacency tree of these contours. Two boundary polygons are adjacent, if the circle inscribed into a figure and contacted both of these polygons exists. The given relation of contour adjacency defines a graph of contour adjacency. It is obvious that this graph is connected. Each spanning set of it (the minimal connected spanning subgraph) is a tree. Such tree is called a boundary adjacency tree of a polygonal figure. The boundary adjacency tree gives the chance to reduce a problem of a polygonal figure skeletonization to a problem of a simple polygon skeletonization. Let us transform chains of side of polygons by “cutting-in” them into one another. As a result the polygonal figure conditionally transforms to “polygon”. In details, this process was described in [18]. The $O(n \log n)$ sweepline algorithm for finding a boundary adjacency tree and a figure skeleton construction on its basis is described *ibidem*.

A skeletal base of polygonal figure. Small irregularities in figure boundary lead to occurrence of skeleton branches unessential for analysis of image form. A problem of skeleton regularization consists in removing these branches and leaving only fundamental part of a skeleton, which characterizes properties of the form. This fundamental part looks like a skeleton subgraph. Let us call it a skeletal base. A transformation of a skeleton to a skeletal base consists in removing of unessential vertices and edges. This process is called a “pruning”.

Let C be a polygonal figure with the boundary ∂C , the skeleton S , and the skeleton radial function $\rho(s)$, $s \in S$. The skeleton is a planar graph $S = (P, E)$ with the set of vertices P and edges E . A skeleton vertex with one incident edge is a terminal, and with two or more edges is an internal. An edge incident to terminal vertex is also called terminal. An edge incident to two internal vertices is called linking. The linking edges can be included in one or more cycles, and in this case they are called cyclic.

A pruning is an iterative removal of “unessential” terminal vertices and skeleton edges. Essential edges remain in a skeletal base. A pruning preserves a skeleton connectivity and all cycles. Let $S' = (P', E')$ be some connected subgraph of skeleton $S = (P, E)$ such that $P' \subseteq P$, $E' \subseteq E$, and there are no cyclic edges of skeleton S among edges from the set $E \setminus E'$ (Fig. 2.11). Such graph S' is called a truncated subgraph of S . Consider the set of points formed by union of all inscribed circles centered in points of truncated subgraph S' , which radiuses are equal $\rho(s)$, $s \in S'$. This set of points forms the closed region, which is called a silhouette of subgraph S' . The important property of a truncated subgraph silhouette is the topological equivalence to figure C . In particular, it is a connected set.

A skeletal base of figure C is the minimal truncated subgraph S' of its skeleton S with ε -silhouette $V_{S'}$ satisfying a condition $H(C, V_{S'}) \leq \varepsilon$, where $\varepsilon > 0$ is regularizing parameter and $H(C, V_{S'})$ is the Hausdorff distance between a figure C and a silhouette $V_{S'}$. For each value of parameter ε the skeletal base always exists and is unique as well as its ε -silhouette. One can call the skeletal base as a continuous skeleton of a discrete figure (Fig. 2.12). A computational complexity of algorithm depends linearly on a number of skeleton vertices, i.e. it is equal $O(n)$ at the worst case, where n is a quantity of polygonal figure vertices.

2.3.4 Skeleton-Based Continuous Binary Morphologies

Let us consider the MM with disk structuring element $D(r)$ of size r . Continuous binary opening requires the infinite number of disks for shape reconstruction

$$P \circ D(r) = \bigcup_{D(r) \subseteq P} D(r).$$

That is why the continuous MM filters are usually out of consideration.

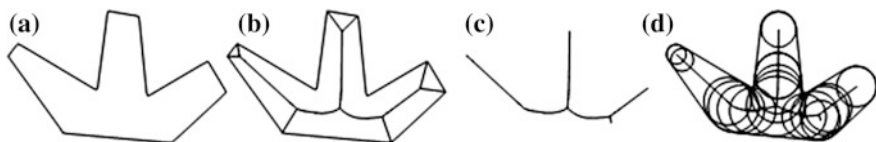


Fig. 2.11 A construction of skeletal subgraph silhouette: **a** the initial image, **b** the skeleton, **c** the truncated subgraph of skeleton, **d** the silhouette of the subgraph

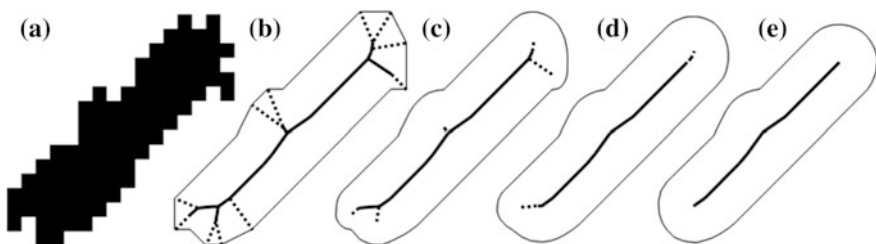


Fig. 2.12 The skeletal base construction: **a** the initial image, **b** the polygonal figure and its skeleton, **c-e** the skeleton subgraphs and their silhouettes

On the other hand, considering the notion of continuous skeleton, Eq. 2.23 for continuous MM opening can be obtained, where $D(p, t)$ is an empty disk of size t with center p , $S(P)$ is a skeleton of figure P .

$$P \circ D(r) = \cup_{p \in S(P)} \{D(p, t) : t \geq r, D(p, t) \subseteq P\} \quad (2.23)$$

If a figure P is polygonal, then its skeleton consists of finite number of line and parabolic segments. This allows representing the figure as a union of finite number of ANalytical struCTure ELements (anxels) (Fig. 2.13). Let each i th edge of skeleton $S(P)$ is denoted as S_i . The silhouette of S_i (a union of all empty disks centered on S_i) let us call the i th anxel of figure P and denote it as P_i . Thus, an anxel representation of figure P takes the form of Eq. 2.24, where n is a number of skeleton edges.

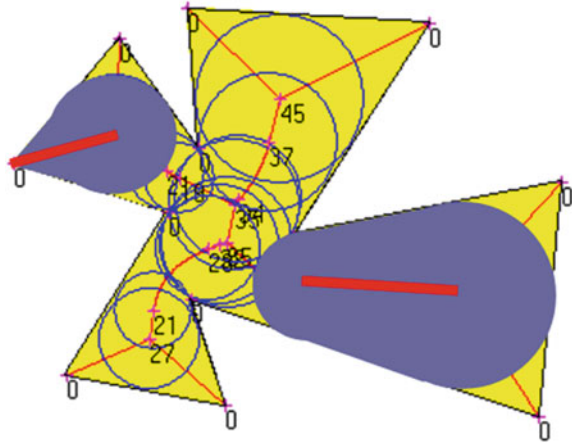
$$P = \cup_{i=1, \dots, n} P_i \quad (2.24)$$

Therefore, the opening of P can be represented by Eq. 2.25.

$$P \circ D(r) = \cup_{i=1, \dots, n} \cup_{p \in S_i} \{D(p, t) : t \geq r, D(p, t) \subseteq P_i\} = \cup_{i=1, \dots, n} P_i \circ D(r) \quad (2.25)$$

The opening of each anxel can be calculated analytically: a silhouette of r -opened anxel P_i will be empty (if all radial function values on S_i are less than r) or bordered by parts of its initial border and possibly parts of circles of size r .

Fig. 2.13 A skeleton of polygonal figure and analytical structure elements (anxels)



Let us note that a mapping of figure P to its ε -silhouette described above also satisfies the properties of morphological filter (opening), and such ε -opening can be described in analogous way using the anxel figure representation as a process of ε -exclusion and ε -cutting of terminal anxels. In both cases of anxel-based morphologies, a corresponding continuous closing is implemented via continuous opening of a figure background. Thus, the anxel representation of polygonal figures allows to define different continuous binary morphologies with different continuous filters based on selection and/or transformation of figure or background anxels.

In the next Sect. 2.4 the applicability of this approach for efficient calculation of morphological pattern spectrum will be demonstrated.

2.4 Morphological Spectrum: Concept and Computation

A morphological spectrum is one of special tools from the reach morphological toolbox. The original Pattern Spectrum (PS) was proposed by Maragos [35] based on the MM filters with SE [1]. It describes the distribution of local figure thickness. Later some modifications and implementations were proposed. All modifications and generalizations of the PS one can refer as the morphological spectra. The morphological spectra are the sensitive and stable descriptors of image shape especially useful for texture analysis, object selection, tuning of morphological filtering and segmentation parameters [11, 36, 37]. However, for many years this morphological tool was not so popular because of its very expensive computational implementation. In this section, let us discuss a computationally efficient approach to morphological spectra calculation based on thickness map concept, continuous skeletal representation [38], and level decomposition of morphological filters [39]. This approach provides a possibility to use the morphological spectra in real-time vision applications.

The pattern spectrum and morphological spectra are discussed in Sect. 2.4.1. The thickness map and morphological spectrum are located in Sect. 2.4.2. Section 2.4.3 contains a calculation of binary morphological spectra based on continuous skeletal representation while as well as a calculation of grayscale morphological spectra is given in Sect. 2.4.4.

2.4.1 Pattern Spectrum and Morphological Spectra

Let B is a figure (compact and convex point set) of the plane P including the origin. Then the figure rB of shape B and size r is defined as follows

$$rB = \{rb = (rx, ry) | b = (x, y) \in B\}, r \geq 0.$$

The morphological PS (PS_X) of figure $X \subseteq P$ with structuring element B [35] is defined by Eqs. 2.26–2.27, where $S(X)$ is an area of figure X . Equations 2.26–2.27 specify the spectrum for positive and negative parts of the r axis, respectively.

$$PS_X(r, B) = -\partial S(X \circ rB) / \partial r \quad r \geq 0 \quad (2.26)$$

$$PS_X(-r, B) = \partial S(X \bullet rB) / \partial r \quad r > 0 \quad (2.27)$$

The Discrete Morphological Pattern Spectrum (DMPS) of X [35] is defined by Eqs. 2.28–2.29, where $r_i = i\Delta r, i \in Z, \Delta r$ is a sampling step of the scale r .

$$PS_{X,B,\Delta r}(r_i) = -\frac{S(X \circ r_i B) - S(X \circ r_{i+1} B)}{r_i - r_{i+1}} \quad r \geq 0 \quad (2.28)$$

$$PS_{X,B,\Delta r}(r_i) = \frac{S(X \bullet (-r_i B)) - S(X \bullet (-r_{i+1} B))}{r_{i+1} - r_i} \quad r < 0 \quad (2.29)$$

The special test image filled by disks of different size is shown in Fig. 2.14. If $B = D$ is a disk SE, then $S(X \circ rB)$ will be the step function of r with steps at each size of disks presented in the image, and the amplitude of the step is equal to the sum of areas of disks with this size. Hence, positive part of discrete pattern spectrum contains peaks corresponding to these steps with the same amplitude.

As noted in [35], the PS conveys four useful types of information about X . A boundary roughness of X relative B determines peaks in the lower size part of the PS. The long capes or bulky protruding parts in X consisting of patterns rB produce isolated impulses in the positive part of the PS at r scale value. The B-shapeness of X is a maximal degree of rB in X measured by ratio $PS_X(r, B)/S(X)$. The left part of the spectrum (r is negative) demonstrates the significant isolated impulses, if corresponding gulfs or holes are in X .

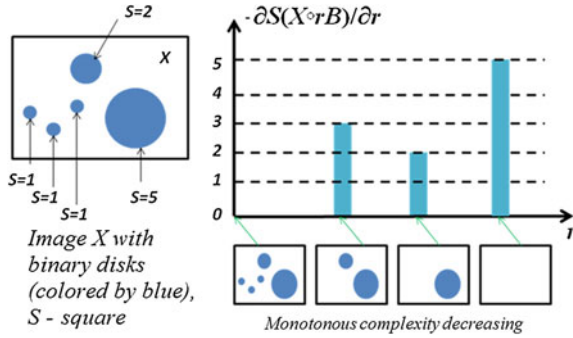


Fig. 2.14 A positive part of morphological spectrum with disk SE and corresponding steps of opening with increasing disk size providing changes in opened figure area

From theoretical point of view, the PS with all its modifications and generalizations (morphological spectra) are based on granulometries—the sets of morphological filters of different grades, each allowing details (grains) of certain size classes to pass. Let $\{\gamma_k\}$, $k = 0, 1, \dots$ be a set opening operators $\forall X \gamma_0(X) = X$, $\gamma_{k+1}(X) \subseteq \gamma_k(X)$. For example, $\gamma_k(X) = X \circ r_k B$. A granulometry function $G_k(X)$ is a cardinality (area, volume, etc.) of the pattern $\gamma_k(X)$: $G_k(X) = |\gamma_k(X)|$. The PS or size distribution of X will be a sequence of differences, produced by sequential subtraction of granulometry function values provided by Eq. 2.30.

$$PS_X(k, \gamma) = G_k(X) - G_{k+1}(X) \quad (2.30)$$

Another generalized scheme was proposed by Matheron in terms of sieving by series of sieves with decreasing the sizes of holes. The filtering of X through k th sieve $\Psi_k(X)$ returns the subset of X sub-elements with sizes not greater than k .

The sieving operators $\{\Psi_k(X)\}$, $k = 0, 1, \dots$ have the following properties:

- The anti-extensive: each sieve reduces the amount of grains, i.e. $\Psi_k(X) \subseteq X$.
- The increasing: a sieving preserves the inclusion, i.e. $X \subseteq Y \Rightarrow \Psi_k(X) \subseteq \Psi_k(Y)$.
- The stable: a passing through two sieves is determined by the smallest hole size, i.e. $\Psi_k \Psi_m(X) = \Psi_m \Psi_k(X) = \Psi_{\max(k,m)}(X)$.

Such sequences of operators are the base of granulometry.

From practical point of view, there are two main classes of morphological spectra: based on filters with SEs and based on the connected filters. The first class contains the original PS [35], the spatial morphological shape-size PS proposed by Wilkinson [40], and some other. The second class includes the size pattern spectra, the shape pattern spectra, the binned 2D shape-size PS [41], and so on.

The morphological spectra based on the SEs are robust and have invariant shape-size descriptors but they are very computationally expensive because they require one opening operation per bin of the spectrum. Therefore, the efforts for improvement of such technique were put to the creation of fast SE filters

computation algorithms [42–50]. The best result was obtained by Urbach and Wilkinson [50]. They have proposed the efficient 2D grayscale morphological transformation with arbitrary “flat” SEs. This algorithm is based on decomposition of SE into “chords”. However, even the use of this fast algorithm provides the SE spectrum computation time about s per image in the best case.

The morphological spectra based on the connected filters may use the attribute openings taking in account both width (thickness) and any other attributes of flat zones. Some of these spectra (with simplest filtering of flat zones by area) even can be implemented for real-time processing. Unfortunately, such spectra do not contain the information about local thickness that characterizes the original PS. Therefore, for real-time implementation the computational approach of the SE morphological spectra calculation, which does not require one opening for each bin, ought to be designed.

2.4.2 Thickness Map and Morphological Spectrum with Disk Structuring Elements

The PS describes the distribution of local figure thickness. This fact points the alternative way for calculation based on the notion of thickness map. Let a rectangular frame K completely contains figure X : $X \subseteq K$. A binary image consists of figure X and its background $X^{C(K)} = K \setminus X$ and described by Eq. 2.31.

$$f_X(x, y) = \begin{cases} 1 & \text{if } p = (x, y) \in X \\ 0 & \text{if } p = (x, y) \in X^{C(K)} \end{cases} \quad (2.31)$$

The thickness map $t_{X,B}(x, y)$ of continuous binary image f_X with structuring element B is a real-valued 2D function defined on the frame K . In this map, all points of the figure have positive values equal to the maximal size of inscribed disk centered in this point. The background values are negative with magnitude equal to the maximal size of disk centered in this point, and they are inscribed to a figure completion. A relation between thickness map and Serra MM filters is provided by Eq. 2.32.

$$X \circ rB = \{(x, y): t_{X,B}(x, y) \geq r\} \quad X \bullet rB = \{(x, y): t_{X,B}(x, y) \geq -r\} \quad (2.32)$$

Let us define the function

$$\chi_{X,B}(x, y, r) = \begin{cases} 1: & t_{X,B}(x, y) \geq r \\ 0 & \text{otherwise} \end{cases}$$

and introduce a following measure

$$\mu_{X,B}(r) = \|\chi_{X,B}(x,y,r)\|_{L^1} = \int \int \chi_{X,B}(x,y,r) dx dy$$

$$\mu_{X,B}(r) = \begin{cases} \|X \circ rB\|_{L^1} : r \geq 0 \\ \|X \bullet (-r)B\|_{L^1} : r < 0 \end{cases}$$

Therefore, the PS according to Eqs. 2.26–2.27 is a distribution density function of thickness map determined by Eq. 2.33.

$$PS_{X,B}(r) = -\partial\mu_{X,B}(r)/\partial r \quad (2.33)$$

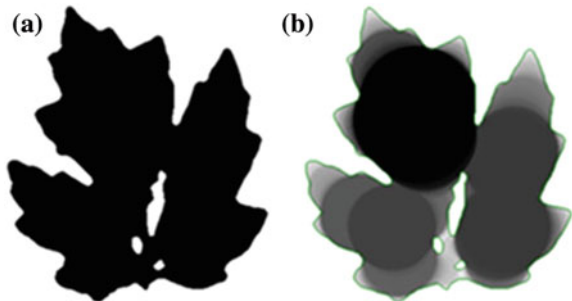
In similar way, one can show that a discrete morphological PS (Eqs. 2.28–2.29) is equal to histogram of discrete thickness map. Let us additionally note that the positive part of the thickness map (Fig. 2.15) and the positive part of the morphological PS remain unchanged, when a figure is shifted or rotated in frame.

Thus, if a thickness map is built, then one opening or closing operation for each bin of the spectrum is not required: just to collect the histogram of a thickness map is needed. However, some computationally efficient algorithm to calculate the thickness map ought to be designed.

2.4.3 Calculation of Binary Morphological Spectra Based on Continuous Skeletal Representation

The crucial idea of fast thickness map computation is derived from the fact that the centers of all empty discs (inscribed in figure and background) belong to the figure's skeleton. Thus, the information required for thickness map computation is already stored in a skeletal representation—skeleton and radial function. In this section, the continuous binary morphology given by Mestetskiy [38] and briefly represented in previous sections will be described. This approach provides the

Fig. 2.15 Example of thickness map: **a** a binary figure X , **b** a positive part of thickness map for a figure X



real-time algorithm for computation of skeletal representation based on the use of generalized Voronoi diagrams [51].

The thickness map forming algorithm is based on the voting of skeleton points into the two-dimensional discrete accumulator array with the same size as binary image to be processed [52–55]. Each continuous skeleton edge is rasterized using the Bresenham algorithm [56], and then each discrete edge point votes for all discrete accumulator cells (pixels) covered by empty disk centered in this point. The purpose of voting is to determine the maximal size (radius) of covering empty disk for each pixel of binary image. The histogram of accumulated maximal thickness values is called the Discrete-Continuous Morphological Pattern Spectrum (DCMPS) [55] due to discrete accumulation based on continuous skeleton. Figure 2.16 demonstrates the binary image of a figure, the positive part of its thickness map, the positive part of DCMPS, and the selected parts of figure with minimal width (local thickness) corresponding to peaks in a spectrum.

The experiments with software implementation of this algorithm have demonstrated the computational time for binary pattern spectrum calculation less than 10 ms for 640×480 binary image on PC configuration CPU Core i5-2320, 3.0 GHz, RAM 2 GB. Therefore, this algorithm can be applied for real-time implementations.

2.4.4 Calculation of Grayscale Morphological Spectra

In the grayscale morphology with image $f(x, y)$ and structuring element $k(u, v)$ the grayscale spectrum PS_f is defined by Eqs. 2.34–2.35 [35], where r is the size of SE $rk(x, y)$ and norm $\|f\|$ is a volume of umbra $U(f)$.

$$PS_f(r, k) = -\partial\|f \circ rk\|/\partial r \quad r \geq 0 \quad (2.34)$$

$$PS_f(-r, k) = \partial\|f \bullet rk\|/\partial r \quad r > 0 \quad (2.35)$$

The use of a level decomposition for the morphological operators allows fast calculation of morphological spectrum for grayscale images and filters with “flat” SEs. Let us consider the discrete N -level two-dimensional function $f(x, y) \in \{0, 1, \dots, N-1\}$ reconstructable from N binary level sets as it is shown in Eq. 2.36.

$$f(x, y) = \sum_{l=0, \dots, N-1} \{f_l(x, y)\} = \max_{l=0, \dots, N-1} \{l \times f_l(x, y)\} \quad (2.36)$$

The “flat” SE has two levels of intensity $k(u, v) \in \{0, -\infty\}$ and corresponds to binary SE $b(u, v) = \{1 \text{ if } k(u, v) = 0, 0 \text{ if } k(u, v) < 0\}$. As shown in [39], the filters of Serra grayscale morphology with “flat” SE may be presented as a combination of corresponding binary MM filters applying to image levels (Eqs. 2.37–2.38)

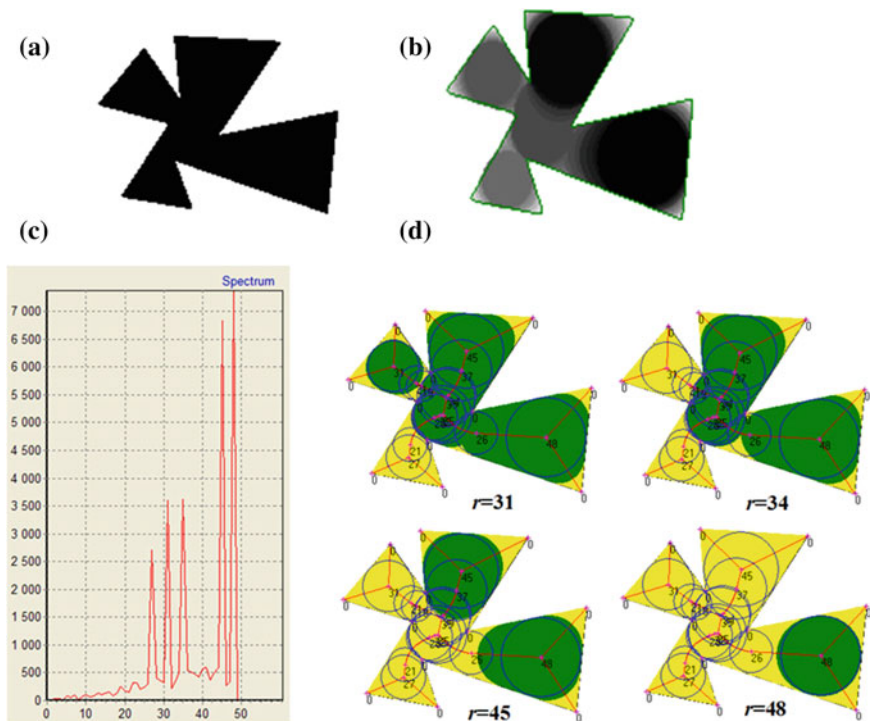


Fig. 2.16 Illustration of DCMPS: **a** the binary image of polygonal figure, **b** the positive part of thickness map, **c** the positive part of DCMPS, **d** the selected parts of figure with minimal width (local thickness) corresponding to essential peaks in a spectrum (results of opening with relevant disk size)

$$f(x, y) \circ b(u, v) = \sum_{l=0, \dots, N} \{f_l(x, y) \circ b(u, v)\} = \max_{l=0, \dots, N} \{l \times f_l(x, y) \circ b(u, v)\} \tag{2.37}$$

$$f(x, y) \bullet b(u, v) = \sum_{l=0, \dots, N} \{f_l(x, y) \bullet b(u, v)\} = \max_{l=0, \dots, N} \{l \times f_l(x, y) \bullet b(u, v)\} \tag{2.38}$$

Therefore, a grayscale spectrum can be calculated as a sum of level spectra by Eqs. 2.39–2.40.

$$PS_f(r, k) = \sum_{l=0, \dots, N-1} PS_{fl}(r, k) \tag{2.39}$$

$$PS_f(-r, k) = \sum_{l=0, \dots, N-1} PS_{fl}(-r, k) \tag{2.40}$$

Thus, the calculation of PS with “flat” disk SE for N -level grayscale image will be N times longer than the computation of the DCMPS for each its binary level image. However, if an approximate spectrum calculation is admissible, the speed of computations can be significantly increased through the use of lower number of binary levels for approximation of the grayscale image. A number n of required approximation levels is determined by a number of significant histogram modes.

Our level selection technique is based on the multimodal generalization of the Otsu bimodal separability criterion. Let us consider the $(n + 1)$ -dimensional vector $\mathbf{t} = (t_0, \dots, t_n)$, where $t_0 = 0$, $t_n = 255$, t_1, \dots, t_{n-1} are free variables corresponding to the thresholds between the histogram modes. If a number of the histogram modes is unknown, then the task of histogram segmentation is, generally speaking, incorrect and requires the regularization [55]. The optimal segmentation corresponds to solution of optimization problem defined by Eq. 2.41, where $\text{DISP}(t_i, t_{i+1})$ is a value of the dispersion of the image histogram fragment on the interval (t_i, t_{i+1}) , α is a regularization parameter.

$$\sum_{i=0, \dots, n} \text{DISP}(t_i, t_{i+1}) + \alpha n \rightarrow \min(n, t_1, \dots, t_{n-1}) \quad (2.41)$$

This problem is solved by the Dynamic Programming (DP) [53]. After that, the approximate morphological spectrum of grayscale image is formed via fast computing and summing of n binary level spectra [53–57]. The negative part of the spectrum (Eq. 2.40) can be calculated using the same algorithm applied to the inverted image (background).

2.5 Morphological Image Analysis (Pyt’ev Morphology)

The basic ideas of Morphological Image Analysis (MIA) are proposed and developed by Pyt’ev since 1960–1970 [58–64]. Further development and generalization of this morphological technique was performed by Pyt’ev, Chulichkov, Kalinin, Loginov, Smolovik, Falomkin, Zhivotnikov, Antonjuk, Zheltov, Vizilter, Rubis, and other researchers of Russian morphological school [65–81].

Let us note that the MIA was proposed approximately at the same time independently and had the proper evolution in parallel way relative to well-known MM proposed by Serra and Matheron [1, 2]. Both terms “morphology” and “morphological analysis” historically belong to both approaches, and they will make some confusion in this chapter. Therefore, for separating MIA from MM, let us refer the MM as the Serra morphology and the MIA as the Pyt’ev morphology (with great respect to all other authors of MM and MIA concepts and results). Unfortunately, basic and some further papers on MIA were published in Russian only, and due to this the Pyt’ev morphology is not so known and popular in the world, but its difference and similarity to the MM are worthy to become a subject of the most fixed consideration.

From practical point of view, the Pyt'ev morphology is basically a technique for invariant and robust image comparison and model-based matching (a template matching in the simplest case). In contrary, the Serra MM is basically a technique for image transforming (a filtering). However, the difference is not only in basic tasks of image analysis. Different classes of objects inspired these morphological approaches in their early days. The Serra morphology was initially developed for binary images of planar figures and then generalized for grayscale and color images. The Pyt'ev morphology was initially developed for comparison of grayscale and color images of 3D scenes. Different target objects produce different concepts of image shape.

From the theoretical point of view, the MIA is another algebraic approach to image shape description and analysis. The Serra morphology is based on non-linear set-theoretic (complete lattice) models. The Pyt'ev morphology is based on vector algebra and functional analysis. Therefore, the monotonous properties (extensive or ant-extensive) of Serra morphological filters (projectors) are not satisfied for Pyt'ev morphological filters (projectors). The Pyt'ev shapes are the rigorous mathematical objects with clear geometrical sense—the hyperplanes in an image space. The Serra shape models have no such interpretation. Due to this, the Pyt'ev morphological correlation coefficient or geometrical correlation coefficients (proposed by Vizilter, Rubis, and Zheltov in the framework of the MIA) have no analogies in the Serra MM. On the other hand, in the Serra MM based on connected filters there are many coincident terms to Pyt'ev notions of shape, shape complexity, and so on. It is important that formally vector algebra and set-theory (Boolean algebra) are the particular cases of the lattice theory. It is important that idempotent operators (projectors) play the central role in both mentioned morphologies. Thus, one can presume some deep unity of these morphological approaches. It was expressed in the formalism of projective morphology in section below.

Now let us start the brief consideration of Pyt'ev morphological ideas. The main purpose of all MIA techniques is the independence of image analysis results on the conditions of image registration. The invariant image properties determine the concept of the image shape. More generally, one can speak about the shape of any data or signal registered by some sensor.

In morphological analysis, it is supposed that any registered signal contains both “important information” about the source signal and the “secondary information” forced by conditions of registration. For example, let us try to explore some 3D scene by means of analysis of its 2D image. An image content depends both on the objects in the scene and the conditions and technique of image acquisition (lighting, weather, season, camera presets, etc.). For the task of 3D scene analysis the information about scene illumination or image acquisition parameters will be “secondary” information. The part of visible information presented in all images of this scene and determined basically by the content of the scene is called a shape of images in this scene. This information should not change, if conditions of scene registration are modified. Therefore, one can speak about the shape of one image, if this image contains the complete “important information” relevant to all images of this fixed scene view.

In order to define the rigorous mathematical notion of image shape, it is required to determine the transformation of the image corresponding to the changes of image acquisition conditions. The invariant of this transformation will be the shape of image. Based on this shape formalization, one can propose the invariant morphological methods for solution of different practical problems of data comparison, matching, recognition, classification, and estimation of sensor parameters for each proper type of signals or data [65, 68, 69].

A model of image shape invariant for transforms is described in Sect. 2.5.1. Scene recognition based on image shape is discussed in Sect. 2.5.2. Section 2.5.3 provides a detection of scene change based on image shape. A scene recognition based on the shape of noisy image is situated in Sect. 2.5.4. A morphological shape matching is given in Sect. 2.5.5.

2.5.1 Image Shape as an Invariant of Image Transforms

Let image f be a function of two variables (x, y) . A domain for definition of this function is called a field of view X , and the value $f(x, y)$ is called the intensity or color of the image at the point $(x, y) \in X$.

For defining the shape of the image, it is need to determine a model of image changing in variations conditions of registration. Let us suppose, for example, that changes in these conditions lead to intensity changes only and are described by Eq. 2.42, where $F(\cdot)$ is the unknown function of pixel-wise intensity transform.

$$g(x, y) = F(f(x, y)) \quad (2.42)$$

Let \mathbf{F} be the class of admissible transformations of the image f , and one can believe that if $F(\cdot) \in \mathbf{F}$, then image $g = F(f)$ will be the other realizable image of this scene corresponding to some certain conditions of registration.

For example, let us suppose that scene objects are the polyhedra with optically homogeneous flat sides. If they are illuminated by the uniform beam of light, then the image of this scene will be the piecewise-constant function in Eq. 2.43.

$$f(x, y) = \sum_{i=1}^N f_i \chi_{F_i}(x, y) \quad (2.43)$$

Here

$$\chi_{F_i}(x, y) = \begin{cases} 1 & \text{if } (x, y) \in A_{F_i} \\ 0 & \text{if } (x, y) \notin A_{F_i} \end{cases}$$

is the indicator function of A_{F_i} , $i = 1, \dots, N$, and the regions A_{F_i} , $i = 1, \dots, N$, with constant intensity form the tessellation of the field of view X :

$$A_{F_i} \cap A_{F_j} = \emptyset, \quad i \neq j, i, j = 1, \dots, N, \quad \bigcup_{i=1}^N A_{F_i} = X.$$

The differences in images of cube captured with different lighting conditions are shown in Fig. 2.17. Figure 2.18 illustrates a mathematical model of cube image and its shape represented as a tessellation of a field of view.

In this case, the class of transforms \mathbf{F} is specified as a class of all (Borel) functions $\{F(\cdot): R_1 \rightarrow R_1\}$. In result of transformation from Eq. 2.42, all sets (regions) of image points with equal intensity on the image f will have the equal intensity on the image g . Therefore, all f image regions of constant intensity (“flat zones” in the Serra MM terms) will preserve their (geometric) shape. However, in some certain special cases, some regions of the field of view X tessellation, which have different intensity on the image f , will merge into one region with constant intensity on the image g . Then the shape of the image g will be simpler than the shape of the image f .

In the described case (polyhedra world with uniform lighting), the invariant of transformations of class \mathbf{F} is a tessellation $\{A_{F_i}\}_{i=1, \dots, N}$ itself. It can be called the shape of the image f . Let us denote the set of all possible images of scene obtained at various registration conditions as follows

$$V_f = \{g = F(f), F \in \mathbf{F}\}.$$

This set can be equivalently considered as a shape of image f .

If V_f is a convex and closed set in the Euclidean space of all images, then there is the one-to-one correspondence between the set V_f and the operator of projection onto this set P_f . This operator can be easily calculated and also called the shape of image f . Obviously, the set of images V_f is invariant with respect to this Pyt’ev morphological projector and defined by Eq. 2.44.

$$P_f: P_f V_f = V_f \quad (2.44)$$

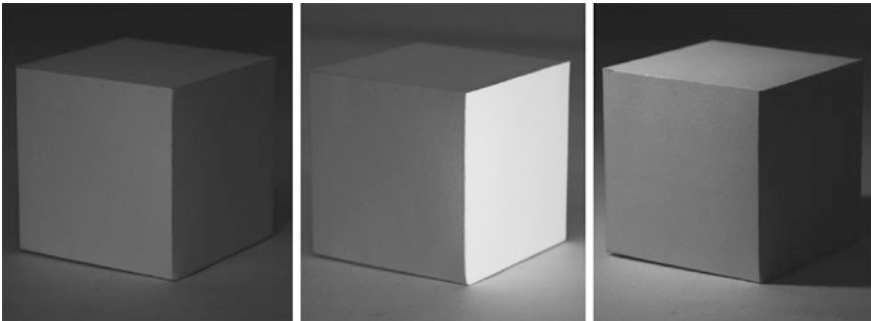


Fig. 2.17 Images of polyhedral world (cube) captured with different lighting conditions

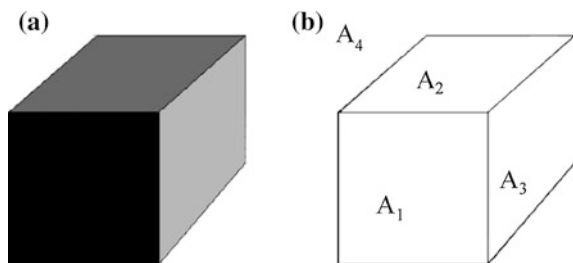


Fig. 2.18 Mathematical model of cube image: **a** a visual model, **b** a shape of cube as a tessellation

Corresponding to Eq. 2.43, an image shape (in this mathematical world) is the linear cover of indicators of regions with equal intensity. From a geometrical point of view, this set of images V_f is a linear subspace (hyperplane) of the Euclidean space of all images. Therefore, a projection of any image g onto the V_f is defined as an orthogonal projection of an image g on this linear subspace.

Constructively, the image projection of g on the set V_f is determined as a solution of the problem of finding best (closest) image in the set V_f to the given image g

$$\|g - P_f g\|^2 = \inf \left\{ \|g - q\|^2 \mid q \in V_f \right\}.$$

For piecewise-constant image f , this problem can be solved in explicit form

$$P_f g = \sum_{i=1}^N \frac{(g, \chi_{F_i})}{\|\chi_{F_i}\|^2} \chi_{F_i}.$$

Let us consider another example. If \mathbf{F} is the class of monotonously increasing functions, than the shape of the image f is a convex cone in Euclidean space of all images. The ordering of values of the intensity of piecewise-constant images is preserved in such monotonous intensity transform model:

$$f(x, y) = \sum_{i=1}^N f_i \chi_{F_i}(x, y), f_1 < f_2 < \dots < f_N \Rightarrow$$

$$F(f) = \sum_{i=1}^N F(f_i) \chi_{F_i}, F(f_1) < F(f_2) < \dots < F(f_N).$$

Local maxima and minima of image intensity are preserved in this shape model. Let us briefly consider some image analysis problems, which can be solved using the Pyt'ev morphology.

2.5.2 Scene Recognition Based on Image Shape

Let image f correspond to scene S registered from some fixed viewpoint by some fixed camera geometry. The inclusion $g \in V_f$ implies that an image g can be an image of the same scene S registered from the same viewpoint with the same camera geometry. The inclusion of $g \in V_f$ is equivalent to the equality $g = P_f g$. This simple condition can be easily and quickly verified.

For any $\varepsilon \geq 0$, the image v is “ ε -similar” to shape of image f , if condition in Eq. 2.45 is executed.

$$\|v - P_f(v)\| \leq \varepsilon \|v\| \quad (2.45)$$

Therefore, the Morphological Correlation Coefficient (MCC) proposed by Pyt'ev has a view of Eq. 2.46.

$$K_m = \frac{\|P_f v\|}{\|v\|} \quad (2.46)$$

The MCC is normalized ($0 \leq K_m \leq 1$) and $K_m = 1$ corresponds to shape equivalence by Eq. 2.47, while $K_m = 0$ corresponds to shape independence by Eq. 2.48, where Πv and Πf are the orthogonal projections of v and f , respectively, onto the set of “flat” images with constant intensity in all points in the field of view X .

$$K_m(v, f) = 1 \Leftrightarrow v \in V_f \Leftrightarrow v = P_f v \quad (2.47)$$

$$\|\Pi v - P_f v\| = 0, \quad \|\Pi f - P_f f\| = 0 \quad (2.48)$$

The coefficient of morphological correlation does not depend on the brightness transform $F(f(x, y))$: $K_m(F(f), f) = 1$. Moreover, if shape of the image v is simpler than shape of the image f , then $K_m(v, f) = 1$.

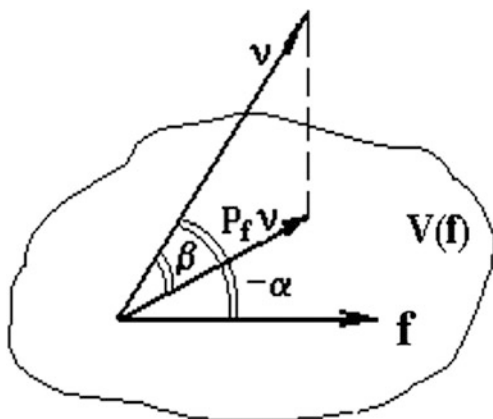
Let us compare this morphological correlation with usual Normalized Correlation (NC) between two images presented in Eq. 2.49.

$$K_u = \frac{(f, v)}{\|f\| \|v\|} \quad (2.49)$$

This NC coefficient is a similarity measure between images taking its values in interval $[-1, 1]$ and invariant relative to class of linear transforms of image intensity: if $v(x, y) = a v(x, y) + b$, then $|K_u(v, f)| = 1$.

A linear transform L is a particular case of arbitrary intensity transform F . Hence, the MCC has greater “invariance power” than the NC: if $K_u = 1$, then $K_m = 1$, but if $K_m = 1$, then K_u should not be equal to 1. Moreover, as it was demonstrated in [74], the MCC is always not less than absolute value of the NC: $K_m \geq |K_u|$. Figure 2.19

Fig. 2.19 Geometrical relation between normalized correlation and morphological correlation, $\alpha = \arccos K_n$, $\beta = \arccos K_m$



illustrates this relation geometrically. The NC is a cosine of angle between v and f in the vector image space. The MCC is a cosine of angle between v and V_f , i.e. between v and $P_f v$. The images are pictured as vectors, and a shape is pictured as a plane (Fig. 2.19). The second angle is always not smaller than the first one, so the MCC is always not less than the NC, and they are equal, if and only if $\|f\| = \|P_f v\|$.

Another difference between the MCC and the NC is that the MCC is not symmetrical relative to v and f : in general case $K_m(v, f) \neq K_m(f, v)$. In particular, if a shape of the image v is simpler than a shape of the image f , than $K_m(v, f) = 1$, but $K_m(f, v) < 1$. The reason of this asymmetry is that the MCC estimates not the similarity between images v and f or between shapes V_v and V_f , but the similarity between image v and shape V_f . So, $\{K_m(v, f) = K_m(v, V_f)\} \neq \{K_m(f, v) = K_m(f, V_v)\}$.

2.5.3 Scene Change Detection Based on Image Shape

Let g and f be the images of the same scene S with some small changes in this scene (some small elements in the scene are added or deleted). If P_f is a projector onto the set of images V_f , then by definition, $P_f g$ is the best approximation of the image g by images from V_f , and hence, $g - P_f g$ is a “morphological difference” of the image g from the shape of image f . This difference is invariant to the conditions of image registration. Thus, the image $g - P_f g$ points to scene changes corresponding to the changes in image g relative to the shape of image f .

The MIA technique is close to the concept of “background normalization” in the MM: the detail extraction uses the difference between initial image and image filtered by morphological filter (projector). Figure 2.20 illustrates this technique for scene change detection in different lighting conditions.

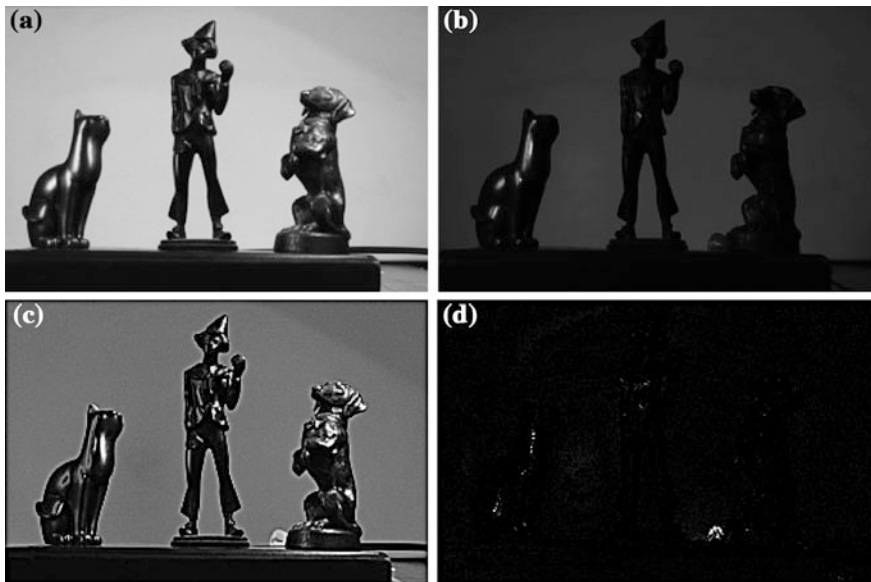


Fig. 2.20 Morphological change detection: **a** the image f determining the shape of scene, **b** the image g with small change in scene content (small bead at the legs of the dog) and strong changes in lighting conditions, **c** a simple difference of f and g (all pixels have changes, so scene changes and intensity changes can not be separated), **d** a morphological difference $g - P_f g$ demonstrates the position of new object (small bead) as an only one bright area

2.5.4 Scene Recognition Based on the Shape of Noisy Image

Let f be the image of scene S_f . Let us consider the registration of corrupted image ζ of some unknown scene: $\zeta = g + v$. Here v is the “noise image”—a model of image corruption by additive noise, and uncorrupted hidden image g is unobservable.

It is required to determine, whether it is possible to consider this image as an image of the scene S_f . For solving this problem in morphological way, let us define the morphological noise-to-signal ratio for image ζ and shape V_f by Eq. 2.50, where $\Pi\zeta$ is the orthogonal projection of ζ onto the set of “flat” images with constant intensity in all points of the field of view X .

$$t(\zeta) = \frac{\|\zeta - P_f \zeta\|^2}{\|\Pi\zeta - P_f \zeta\|^2} \quad (2.50)$$

Consequently, $\Pi\zeta$ is a constant image filled by the average value of intensity of the image ζ . The shorter the distance from ζ to V_f , and more difference of P_f from the constant, the smaller the ratio $t(\zeta)$. The numerator of this ratio contains the squared norm of difference from image ζ to closest image from the set V_f . If $\zeta \notin V_f$, then this difference can be explained only by the presence of noise. The denominator

is the squared norm of the component of image ζ , which is comparable in shape with an image f , and differs from the constant. For decision making about the scene recognition based on images distorted by noise, one needs to specify the threshold value for noise-to-signal ratio.

This brief overview presents the MIA as it was basically proposed. Next paragraphs will be devoted to description of modern generalizations of the MIA, but one needs to note that the MIA approach itself is still under development and able to generate some essentially new ideas. For example, in [82] the new MIA concepts of morphological oblique projectors and relative shapes are proposed by Pyt'ev for the morphological analysis of classes of images and for comparative analysis of their shapes as invariants (under image acquisition conditions). These concepts are used to characterize morphological dependences. More specifically, the relative shapes are characterized by the morphological independence index, and the absolute shapes, by the morphological connection index. New methods based on the constructs of relative shapes of image classes and on the oblique projection technique are described in [82] applying to the comparative analysis of absolute and relative shapes of image classes, the morphological filtration of images, the identification of images, the determination of unknown objects in scene images, and other problems.

2.5.5 Morphological Shape Matching

This subsection addresses the problem of image matching “just by shape” with no dependence on the concrete pixel values. For example, one can compare images of one scene captured at different seasons, time of day, weather and lighting conditions, spectral ranges, etc.

The most popular technique for such image shape comparison utilizes the Mutual Information (MI) measure based on probabilistic reasoning and information theory [83]. A mutual information $I(A, B)$ estimates the dependence of two random variables A and B by measuring the distance between the joint distribution $p_{AB}(a, b)$ and the distribution of complete independence $p_A(a)p_B(b)$ provided by Eqs. 2.51–2.53, where $H(A)$ is an entropy of A , $H(B)$ is an entropy of B , $H(A, B)$ is their joint entropy.

$$I(A, B) = H(A) + H(B) - H(A, B) \quad (2.51)$$

$$H(A) = - \sum_a p_A(a) \log p_A(a) \quad H(B) = - \sum_b p_B(b) \log p_B(b) \quad (2.52)$$

$$H(A, B) = - \sum_a \sum_b p_{AB}(a, b) \log p_{AB}(a, b) \quad (2.53)$$

For two image intensity values a and b of a pair of corresponding pixels in two images, required empirical estimations for the joint and marginal distributions can

be obtained by normalization of the joint (2D) and marginal (1D) histograms of compared image fragments. Maximal $I(A, B)$ value corresponds to the best geometrical matching of image fragments.

Such MI approach provides the robust tool for matching of images with different intensities based on their joint 2D histograms. But these histograms cannot explain the geometrical idea of image “shape” in some evident form. Such mathematical “shape” formalism is given in evident form in the morphological approach to image comparison proposed by Pyt’ev [62].

The Pyt’ev morphological comparison of images $f(x, y)$ and $g(x, y)$ is performed using the MCC $K_M(g, V_f)$ and $K_M(f, V_g)$. This comparison is invariant relative to intensity transforms. However, the Pyt’ev MCC estimates the closeness of image to the shape of other image, but not the similarity of two shapes. Morphological tools for shape matching were proposed by Vizilter, Rubis, and Zheltov [84, 85]. The original Pyt’ev morphological approach was generalized for obtaining the pure “geometry-to-geometry” shape matching technique. In [85], the transform distance for geometrical difference evaluation of shapes named Geometrical Difference Index (GDI) was proposed and normalized similarity measure of image shapes based on GDI was defined. These new morphological tools were experimentally compared with the Pyt’ev MCC and the MI applying to multispectral image matching problem.

In [81], the geometrical shape comparison approach was developed based on Pyt’ev’s morphological image analysis. Let $f(x, y)$ from V_f is a piecewise-constant 2D function described above. Image $g(x, y)$ from V_g is an analogous 2D function with m as a number of tessellation regions $\{A_{G1}, \dots, A_{Gm}\}$, $\mathbf{g} = (g_1, \dots, g_m)$ is a vector of intensity values, $\chi_{G_j}(x, y) \in \{0, 1\}$ is a support function of j th region.

For brevity in this subsection, let us use the following notation: $F = V_f$ and $G = V_g$ for shapes, $F_i = A_{F_i}$ and $G_j = A_{G_j}$ for tessellation regions, $f_G = P_g f$ and $g_F = P_f g$ for projections. Let us also introduce following additional set of “ S -variables”: S is an area of the whole frame Ω , $S_i = \|\chi_{F_i}(x, y)\|^2$ is an area of tessellation region F_i , $S_j = \|\chi_{G_j}(x, y)\|^2$ is an area of tessellation region G_j , $S_{ij} = (\chi_{F_i}(x, y), \chi_{G_j}(x, y))$ is an area of intersection $F_i \cap G_j$. With account of these S -variables, one can receive the equations mentioned below.

$$\begin{aligned} \|f\|^2 &= \sum_{i=1, \dots, n} f_i^2 S_i & \|f_G\|^2 &= \sum_{j=1, \dots, m} f_{G_j}^2 S_j \\ f_{G_j} &= \left(\sum_{i=1, \dots, n} f_i S_{ij} \right) / S_j & j &= 1, \dots, m \end{aligned}$$

Let us add the following assumptions about the distributions of probability densities for intensity values f_1, \dots, f_n :

1. If $p(f_1, \dots, f_n) = p(f_1) \dots p(f_n)$, then values f_1, \dots, f_n are independent in general.
2. If $p(f_1) = \dots = p(f_n)$, then values f_1, \dots, f_n are equally distributed.

3. If $\forall i = 1, \dots, n: p(f_i) = p(-f_i)$, then values f_1, \dots, f_n are distributed symmetrically to 0.

Then the expectation $\langle f_i \rangle = 0, i = 1, \dots, n$ and the covariance has the form of Eq. 2.54, where σ is a dispersion of probability distribution $p(f_i)$.

$$\langle f_i f_k \rangle = \begin{cases} \sigma^2 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \quad (2.54)$$

Thus, the mean square of norm for image f of shape F has a form of Eq. 2.55.

$$\langle \|f\|^2 \rangle = \sum_{i=1}^n \langle f_i^2 \rangle S_i = \sum_{i=1}^n \sigma^2 S_i = \sigma^2 \sum_{i=1}^n S_i = \sigma^2 S \quad (2.55)$$

The mean square of projection norm for image $f \in F$ and fixed shape G has a form of Eq. 2.56.

$$\begin{aligned} \langle \|f_G\|^2 \rangle &= \sum_{j=1}^m \langle f_{G_j}^2 \rangle S_j = \sum_{j=1}^m \left\langle \left(\sum_{i=1}^n f_i S_{ij} \right)^2 / S_j^2 \right\rangle \\ S_j &= \sum_{j=1}^m \left(\sum_{i=1}^n \sigma^2 S_{ij}^2 \right) / S_j = \sigma^2 \sum_{j=1}^m \sum_{i=1}^n S_{ij}^2 / S_j \end{aligned} \quad (2.56)$$

Let us define the Mean Square Effective Morphological Correlation Coefficient (MSEMCC) for shapes $F = V_f$ and $G = V_g$ provided by Eq. 2.57.

$$K_M^2(F, G) = \frac{\langle \|f_G\|^2 \rangle}{\langle \|f\|^2 \rangle} \quad (2.57)$$

After evident substitutions, the MSEMCC takes the compact form by Eq. 2.58, where $K_\Omega(F_i, G_j) = S_{ij}/S$ is a normalized influence coefficient for pair of regions F_i and G_j , $K_M^2(G_j, F_i) = S_{ij}/S_j$ is a square of normalized morphological correlation for this pair of regions.

$$K_M^2(F, G) = \sum_{j=1}^m \sum_{i=1}^n \frac{S_{ij}}{S} \frac{S_{ij}}{S_j} = \sum_{j=1}^m \sum_{i=1}^n K_\Omega(F_i, G_j) K_M^2(G_j, F_i) \quad (2.58)$$

In [85], the special transform distance for geometrical difference evaluation of shapes named GDI was proposed and some normalized similarity measures of image shapes based on the GDI were presented. The GDI metrics for two image shapes F and G are defined by Eqs. 2.59–2.61.

$$d_H(F, G) = \sum_{j=1}^m \sum_{i=1}^n p_{ij} d_H(G_j, F_i) \quad (2.59)$$

$$d_H(G_j, F_i) = p_i + p_j - 2p_{ij} \quad (2.60)$$

$$p_{ij} = S_{ij}/S, p_i = S_i/S, p_j = S_j/S \quad (2.61)$$

Most useful the GDI-based similarity measure proposed in [85] is a Centered Metrical Similarity Coefficient (CMSC), which is based on comparison with assumption about F and G shapes independence. In this case $d_{HInd}(\cdot)$ has a view below.

$$d_{HInd}(F, G) = \sum_{j=1}^m \sum_{i=1}^n p_i p_j (p_i + p_j - 2p_{ij})$$

Therefore, the CMSC is defined by Eq. 2.62.

$$K_{CHS}(F, G) = \frac{|d_{HInd}(F, G) - d_H(F, G)|}{d_{HInd}(F, G)} \quad (2.62)$$

The CMSC has the following properties:

1. $K_{CHS}(F, G) \in [0, 1]$.
2. $K_{CHS}(F, G) = 1 \Leftrightarrow F = G$.
3. $K_{CHS}(F, G) = K_{CHS}(G, F)$.
4. $\forall i, j: p_{ij} = p_i p_j \Rightarrow K_{CHS}(F, G) = 0$.

For comparison of geometrical correlation techniques and corresponding similarity measures, the MI criterion and the Pyt'ev's MCC were calculated over a set of real images including remote sensing and multispectral images [TeleVision (TV) and InfraRed (IR)] [86]. In all experiments, both the Pyt'ev MCC and geometrical correlation measures (Eqs. 2.58, 2.62) provide matching characteristics (signal-to-noise ratio and elevation of main peak in a correlation field) close to mutual information characteristics (a little bit worse or better). In the case of small TV fragments and noisy IR images, the characteristics of the CMSC (Eq. 2.62) were better than Pyt'ev's morphological coefficient and mutual information characteristics. Figures 2.21 and 2.22 demonstrate the example of such TV-IR matching test and corresponding correlation fields. From the computational viewpoint, the geometrical correlation outperforms the mutual information in calculation speed about 20 %.

Thus, the morphological techniques for change detection and scene recognition are invariant relative to conditions of image registration.

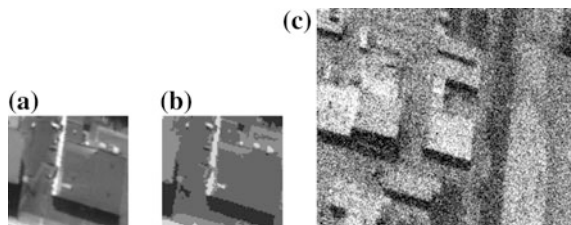


Fig. 2.21 Example of TV-IR matching: **a** a TV etalon, **b** a segmented TV fragment, **c** test IR

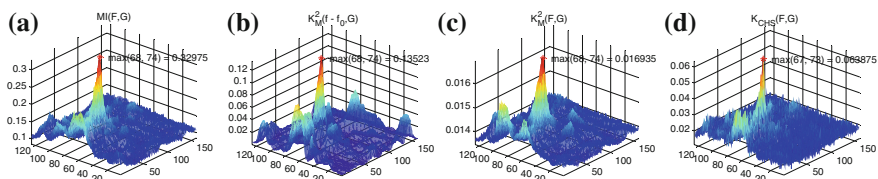


Fig. 2.22 Correlation fields for TV-IR matching: **a** the mutual information $MI(F, G)$, **b** the square of centered Pyt'ev MCC $K_M^2(f - f_0, G)$, **c** the square of MSEMCC $K_M^2(F, G)$, **d** the square of CMSC $K_{CHS}(F, G)$

2.6 Projective Morphologies, Morphological Segmentation and Complexity Analysis

The projective morphology approach was developed based on the Serra MM [1], the Pavel shape theory [87], and the Pyt'ev MIA [62]. It presumes structural image modeling with regularization constrains. This section describes the image segmentation problem from the morphological point of view and introduces the criterion-based morphological filters (projectors) and morphological spectra based on regularization and analysis of shape complexity [79, 80, 84, 88]. The corresponding morphological tools have been successfully applied for different practical computer vision tasks [89–95].

Section 2.6.1 provides the projective morphologies based on morphological decompositions. The image segmentation in the framework of projective morphology is represented in Sect. 2.6.2. The shape regularization and morphological filters are discussed in Sect. 2.6.3. The morphological complexity and filters and spectra by complexity are introduced in Sect. 2.6.4.

2.6.1 Projective Morphologies Based on Morphological Decompositions

The projective space of patterns (images) is as an algebraic system $\langle \Psi, \Omega, \cdot, \vee, \mu, Pr, E \rangle$, where Ψ is a set of scalars including 0 and 1, Ω is the set of patterns with

“zero pattern” \emptyset , ‘ \cdot ’ is the multiplicative group operation of multiplication of scalars $\Psi \times \Psi \rightarrow \Psi$ and a scalar by pattern multiplication $\Psi \times \Omega \rightarrow \Omega$, ‘ \vee ’ \in {‘+’, ‘ \times ’, ‘ \cup ’, ‘ \cap ’, ‘ \vee ’, ‘ \wedge ’, ‘min’, ‘max’, ...} is the additive Abel semi-group of scalars fusion $\Psi \times \Psi \rightarrow \Psi$ and patterns fusion $\Omega \times \Omega \rightarrow \Omega$, μ is the norm of the pattern $\Omega \rightarrow \mathbf{R}$ ($\mu(A) = \|A\|$, $\|\emptyset\| = 0$), set of basic patterns (primitives) $\mathbf{E} = \{E_1, \dots, E_n\}$ is the basis of the morphological pattern decomposition. Let \mathbf{E} denote the corresponding morphological subspace $\mathbf{E} \subseteq \Omega$ generated by the algebraic closure of basis \mathbf{E} relative to ‘ \cdot , \vee ’-combination.

The operator of linear projection of pattern onto the pattern has a form of Eq. 2.63, where $r(A, B) \in \Psi$ is the coefficient of linear dependence of pattern A relative to pattern B .

$$Pr(A, B) = r(A, B) \cdot B: \Omega \rightarrow \mathbf{B} \subseteq \Omega \quad (2.63)$$

The projection of pattern onto subspace

$$Pr(A, \mathbf{E}): \Omega \rightarrow \mathbf{E} \subseteq \Omega, Pr(A, \mathbf{E}) = Pr(Pr(A, \mathbf{E}), \mathbf{E})$$

is the idempotent operator satisfying the decomposition condition Eq. 2.64, where $\mathbf{a}(A, \mathbf{E}) = \langle r(A, E_k) \rangle_{k=1, \dots, n}$ is the vector of morphological decomposition of pattern A in basis \mathbf{E} .

$$Pr(A, \mathbf{E}) = \bigvee_{k=1, \dots, n} Pr(A, E_k) = \bigvee_{k=1, \dots, n} r(A, E_k) \cdot E_k \quad (2.64)$$

The morphological decomposition is a mapping defined by Eq. 2.65.

$$\mathbf{dec}_{\mathbf{E}}(A) = \langle r(A, E_1), \dots, r(A, E_n) \rangle: \Omega \rightarrow \Psi^n \quad (2.65)$$

2.6.2 Image Segmentation in the Framework of Projective Morphology

Let the morphological descriptor of pattern $A \in \Omega$ be a data structure of the form of Eq. 2.66, where $\mathbf{E} = \langle E_1, \dots, E_n \rangle \in \Omega^n$ is a basis of decomposition, $n = \dim(\mathbf{E})$ is the dimension of \mathbf{E} , $d(A, E_i)$ is the descriptor of decomposition element.

$$\mathbf{d}(A, \mathbf{E}) = \langle n, d(A, E_1), \dots, d(A, E_n) \rangle \quad (2.66)$$

The descriptor size $v(\mathbf{d})$ is a memory size required for storing of descriptor \mathbf{d} (Eq. 2.66). The basis \mathbf{X} of decomposition is complete on Ω if $\forall A \in \Omega: Pr(A, \mathbf{X}) = A$. Then for any $A \in \Omega$ its descriptor $\mathbf{d}(A, \mathbf{X})$ is a complete descriptor. Any subbasis $\mathbf{Y} = \langle Y_1, \dots, Y_m \rangle: \mathbf{Y} \subseteq \mathbf{X}$, $\dim(\mathbf{Y}) \leq \dim(\mathbf{X})$ determines a subdescriptor $\mathbf{d}(A, \mathbf{Y})$. The set $\Theta(\mathbf{X}) = \{\mathbf{d}(A, \mathbf{Y}): A \in \Omega, \mathbf{Y} \subseteq \mathbf{X}\}$ is a set of all subdescriptors based on \mathbf{X} .

In the framework of shape theory [87], the morphological segmentation operator can be stated as a mapping of pattern from Ω to subdescriptor from $\Theta(\mathbf{X})$ provided by Eq. 2.67.

$$\varepsilon_s: \Omega \rightarrow \Theta(\mathbf{X}) \quad (2.67)$$

The operator of morphological reconstruction is defined in a following view

$$\delta_s: \Theta(\mathbf{X}) \rightarrow \Omega.$$

Therefore, their combinations will be the morphological filter, and algebraic projector is computed by Eq. 2.68.

$$Pr(A, \Theta(\mathbf{X})) = \psi_s \quad A = \delta_s \varepsilon_s A \quad \psi_s: \Omega \rightarrow \Omega \quad \psi_s^2 = \psi_s \quad (2.68)$$

Let us define a segmentation criterion or shape cost function with regularization parameter α in the form of Eq. 2.69, containing a penalty both for the descriptor size v (complexity criterion) and for deviation of projected pattern from the initial pattern J (reconstruction criterion).

$$\Phi(A, \mathbf{Y}) = J(A, Pr(A, \mathbf{Y})) + \alpha \times v(\mathbf{d}(A, \mathbf{Y})) \rightarrow \min(\{Y: Y \subseteq X\}) \quad (2.69)$$

The optimal morphological segmentation finds an optimal subbasis \mathbf{Y} (Eq. 2.70).

$$\varepsilon_\Phi(A, \mathbf{X}) = \operatorname{argmin}_{\mathbf{Y}} \Phi(A, \mathbf{Y}) \quad (2.70)$$

Then the pattern A as its projection on this subbasis is reconstructed by Eq. 2.71.

$$\psi_\Phi(A, \mathbf{X}) = Pr(A, \varepsilon_\Phi(A, \mathbf{X})) \quad (2.71)$$

In particular, the segmentation without losses provides the exact reconstruction by Eq. 2.72.

$$\varepsilon_v(A, \mathbf{X}) = \operatorname{argmin}_{\mathbf{Y}} \{v(\mathbf{d}(A, \mathbf{Y})): Y \subseteq X \quad Pr(A, \mathbf{Y}) = A\} \quad \psi_v(A, \mathbf{X}) = A \quad (2.72)$$

The segmentation without losses presumes the construction of optimal basis via elimination of zero-coefficient primitives and grouping of equal-coefficient primitives. Each morphological system has the special constraints on grouping.

Example 1 Let initial complete decomposition \mathbf{X} be a pixel tessellation (Eq. 2.73), where $\varphi(i, j, x, y)$ is an indicator function of pixel (x, y) , $a_{ij} = f(i, j)$.

$$f(x, y) = \sum_{ij} a_{ij} \varphi(i, j, x, y) \quad (2.73)$$

The segmentation scheme without losses is based on grouping of neighbor pixels with equal values. This scheme automatically generates the Pyt'ev morphological

shape [62] as an orthogonal projective decomposition of the form of Eq. 2.74, where $\chi_i \in \{0, 1\}$ is an indicator function of i th connected region with intensity a_i .

$$f(x, y) = \sum_{i=1, \dots, n} a_i \cdot \chi_i(x, y) \quad (2.74)$$

Example 2 A granulometry based on the binary Serra MM with disk SE provides a monotonous projective decomposition with operation ‘V’ = ‘U’ and complete basis of binary structuring elements $D(x, y, R)$. In this case, the segmentation scheme without losses is based on following rule for grouping of structuring elements: the greater disks absorb the smaller ones that completely belong to them. Thus, the segmentation provides a minimal set of inscribed disks recovering the pattern A . In other words, this segmentation scheme automatically generates a morphological shape descriptor based on a skeletonization technique.

2.6.3 Shape Regularization and Morphological Filters by Regularization

If optimal segmentation $\varepsilon_{\Phi}(A, \mathbf{X})$ may return the subbasis \mathbf{Y} such that $Pr(A, \mathbf{Y}) \neq A$, then such scheme is called the segmentation with losses or shape regularization. If the operator of shape regularization $\psi_{\Phi}(A, \mathbf{X})$ is idempotent (projector), then it is a morphological filter by regularization. For some certain forms of regularization criterion the projectivity of segmentation operator can be proved. The first type of projective segmentation is a minimal distance regularization provided by Eq. 2.75, where a distance $\rho(A, B) = \|A - B\|$ satisfies the metrics properties by Eq. 2.76.

$$\Phi(A, \mathbf{Y}) = \|A - Pr(A, \mathbf{Y})\| + \alpha \times v(\mathbf{d}(A, \mathbf{Y})) \rightarrow \min(\mathbf{Y}: \mathbf{Y} \subseteq \mathbf{X}) \quad (2.75)$$

$$\forall A, B, C \in \Omega: \rho(A, B) \geq 0, \rho(A, A) = 0, \rho(A, B) + \rho(B, C) \leq \rho(A, C) \quad (2.76)$$

Additionally assume that for any basis \mathbf{E} Eq. 2.77 is executed.

$$\forall A \in \Omega: B = Pr(A, \mathbf{E}) \Leftrightarrow B \in \mathbf{E} \quad \forall C \in E: \|A - B\| \leq \|A - C\| \quad (2.77)$$

As it was proved in [96], the operators $\psi_{\Phi}(A, \mathbf{X})$ satisfying Eqs. 2.75 and 2.77 are idempotent and called the minimal distance projectors (in particular, the Pyt'ev projector from Example 1). Thus, one can speak about a class of morphological systems $\langle \mathbf{X}, \alpha, v, \rho \rangle$ based on a distance ρ , a basis \mathbf{X} , a weight parameter α , and a descriptor size $v(\mathbf{d}(\mathbf{Y}))$.

Example 3 The minimal L1-distance segmentation of 2D functions using Dynamic Programming technique (DP-segmentation) requires the description of 2D image structure by some non-circular graph (tree). Such projective segmentation scheme based on level set trees is described in [97] (see Fig. 2.23).

The second type of projective segmentation is a monotonous regularization provided by Eq. 2.78 or Eq. 2.79, where $A \leq B \Leftrightarrow \forall x, y: A(x, y) \leq B(x, y)$.

$$\forall A, \mathbf{X}: \psi_{\Phi}(A, \mathbf{X}) = Pr(A, \varepsilon_{\Phi}(A, \mathbf{X})) \leq A \quad (2.78)$$

$$\forall A, \mathbf{X}: \psi_{\Phi}(A, \mathbf{X}) = Pr(A, \varepsilon_{\Phi}(A, \mathbf{X})) \geq A \quad (2.79)$$

As it was proved in [96], for any convex criterion J operators $\psi_{\Phi}(A, \mathbf{X})$ satisfying Eqs. 2.78–2.79 are idempotent and called the regularization opening and the regularization closing, respectively. Therefore, one can speak about morphological systems $\langle \mathbf{X}, \alpha, v, J \rangle$ based on an initial basis \mathbf{X} , a weight parameter α , a descriptor size $v(\mathbf{d}(\mathbf{Y}))$, and a reconstruction cost $J(Pr(A, \mathbf{Y}))$.

Example 4 The monotonous segmentation of 1D и 2D functions using the DP technique is similar to one in Example 3 with account of Eq. 2.78 for DP-opening and Eq. 2.79 for DP-closing. Note, that filters in Examples 3 and 4 are the connected filters of the Serra MM.

The DP is not the only effective programming techniques for solution of morphological segmentation tasks. Very efficient solutions can be obtained using the linear programming, the graph cut technique [98–103] and the Hough-like transforms (see Example 5) (Fig. 2.24)

Example 5 A morphological segmentation based on the Hough transform [88] is proposed in [104]. It includes the following steps:

- Step 1: The application of the Hough Transform (HT).
- Step 2: The binarization of the Hough accumulator with fixed threshold value t .
- Step 3: The deletion all dots of source dot pattern, which are not on detected as straight lines.

Such procedure defines the Hough-projector (the Hough-opening MM filter). The optimal segmentation problem is reduced to search of optimal segmentation parameter $t_{\text{opt}} = t(\alpha)$. This approach can be easily expanded to any generalization or modification of the HT. For example, the Recurrent Hough Transform (RHT) in a



Fig. 2.23 The projective DP-segmentation of grayscale image by level set tree with L1-distance criterion and regularization parameter α producing different number of support regions n : **a** $\alpha = 0$, $n = 1,395$, **b** $\alpha = 1,000$, $n = 102$, **c** $\alpha = 10,000$, $n = 23$, **d** $\alpha = 1,000,000$, $n = 2$



Fig. 2.24 Projective monotonous DP-opening of grayscale image by level set tree with regularization parameter α producing different number of support regions n : **a** $\alpha = 0$, $n = 1,584$, **b** $\alpha = 1,000$, $n = 120$, **c** $\alpha = 10,000$, $n = 24$, **d** $\alpha = 100,000$, $n = 8$

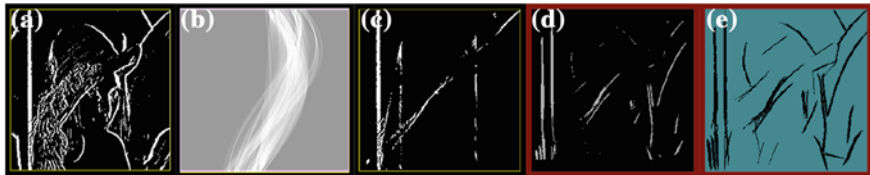


Fig. 2.25 Examples of Hough and RHT-opening: **a** the initial binary image, **b** the Hough accumulator, **c** the Hough-opening (global linear structures are detected), **d** the RHT accumulator, **e** the RHT-opening (local linear structures are extracted)

sliding window produces the morphological RHT-opening [104]. The examples of the Hough-opening and the RHT-opening are shown in Fig. 2.25.

2.6.4 Morphological Complexity, Filters, and Spectra by Complexity

The morphological regularization can be generalized in terms of the morphological complexity proposed by Pyt'ev. Let us consider two projectors φ_1 and φ_2 and corresponding to Pyt'ev shapes (sets of stable elements of projectors). If a shape M_1 contains shape M_2 (i.e. filter φ_1 preserves all patterns filtered by φ_2), then the shape M_1 has less (not greater) morphological complexity than shape M_2

$$M_1 = \{\varphi_1 A : A \in \Omega\}, \quad M_2 = \{\varphi_2 A : A \in \Omega\}: M_2 \subseteq M_1 \Rightarrow \varphi_1 \varphi_2 = \varphi_2.$$

Usually one can speak and think about the structural complexity of patterns: the greater numbers of elements is used in a model of shape, the more complex model is formed. But such complexity definition requires some structural description (for example, in terms of morphological decompositions). In contrast, morphological complexity definition does not require any structural description. It operates just

with projectors and corresponding stable sets but can be easily interpreted in any certain morphological system. For example, if one frame tessellation is a partition of other frame tessellation, it will be relatively more complex both in structural (a number of elements corresponds to a region) and in morphological sense (based on comparison of Pyt'ev projectors). In the case of the Serra MM with disk SE opening, the smaller size of disk corresponds to more complex shape than opening with greater size of disk in both senses: larger number of small disks required for reconstruction and opening with smaller disk preserves results of opening with greater disk.

Thus, one can define the morphological complexity regularization criterion in the most general form (free of structural morphological decomposition terms) provided by Eq. 2.80, where $A, L \subseteq \Omega$ are original and reconstructed patterns correspondingly, $J(A, L)$ is a precision of reconstruction, $Q(L)$ is a complexity of reconstruction, $\alpha \geq 0$ is a parameter of morphological complexity of the operator ψ_α , regulating a compromise between J and Q .

$$F_\alpha(A, L) = J(A, L) + \alpha Q(L) \quad (2.80)$$

The corresponding morphological filter by complexity is defined by Eq. 2.81.

$$\psi_\alpha A = \operatorname{argmin}_{L \in \Omega} F_\alpha(A, L) \quad (2.81)$$

In general, $J(A, L)$ can be any function of L that is monotonously growing with complexity $Q(L)$. The corresponding morphological spectrum by complexity is defined as a derivative of J by complexity parameter α in Eq. 2.82.

$$PS(A, \alpha) = \partial J(A, \psi_\alpha A) / \partial \alpha \quad (2.82)$$

Such definitions are more general than definitions of morphological filters and spectra based on granulometry sieving or any other structural models considered in the MM. Thus, the notions of morphological filters by complexity and morphological spectra by complexity proposed in [105] provide the most general morphological tool for shape analysis based on shape complexity.

2.7 Conclusion

Some original and modern morphological concepts and tools were presented in this chapter as well as required amount of morphological basics. The morphological techniques, which are applicable for real-time technical vision systems, were selected and presented. The continuous skeleton is described by the strict mathematical model. A computation time of continuous skeletonization algorithm outperforms the best samples of discrete skeletonization algorithms by a factor of ten or even hundred. If a figure is polygonal, then continuous skeletonization represents

it as a union of finite number of ANalytical struCTure ELEments (anxels). Such anxel representation allows to define different continuous binary morphologies based on selection and/or transformation of figure or background anxels. The effective approach to the calculation of morphological pattern spectra of binary and grayscale images with the disk structuring elements is described. It is based on the continuous skeletal representation, the thickness map concept and the level decomposition of morphological spectra.

Morphological image analysis (Pyt'ev morphology) is described. The Pyt'ev morphology is developed for matching and comparison of grayscale and color images of 3D scenes. It is based on vector algebra and functional analysis. The Pyt'ev shapes are the hyperplanes in an image space. Morphological techniques for change detection and scene recognition based on image-to-shape and shape-to-shape similarity estimation are described. These techniques are invariant relative to conditions of image registration.

The projective morphology is described as a generalized framework based on the Serra morphology, the Pavel shape theory, and the Pyt'ev morphological analysis. The projective morphology combines ideas of these morphological approaches and allows to construct some new morphological systems and operators based on different image decompositions, transforms, and criterions (energy functions). The morphological shape complexity as a criterion for shape regularization is the basis of tools for shape complexity analysis.

Acknowledgments Authors thank all colleagues from Moscow Morphological Workshop in the Lomonosov Moscow state university (supervised by Prof. Y. Pyt'ev) for many-years fruitful and kind discussions. Special thanks are to Russian Fund of Basic Researches supported the morphological researches by a series of grants.

References

1. Serra J (1982) Image analysis and mathematical morphology. Academic Press, London
2. Matheron G (1975) Random sets and integral geometry. Wiley, New York
3. Serra J (1988) Image analysis and mathematical morphology. Theoretical advances. Academic Press, London
4. Dougherty ER (1992) An introduction to morphological image processing. SPIE Optical Engineering Press, Bellingham, Washington, USA
5. Najman L, Talbot H (2010) Mathematical morphology: from theory to applications. Wiley, Hoboken, NJ
6. Serra J, Soille P (1994) Mathematical morphology and its applications to image processing. Kluwer Academic Publishers, Dordrecht
7. Shih FY, Mitchell OR (1989) Threshold decomposition of gray scale morphology into binary morphology. IEEE Trans Pattern Anal Mach Intell 11(1):31–42
8. Grätzer G (2011) Lattice theory: foundation Springer Basel
9. Nachtgael M, Sussner P, Mélangé T, Kerre E (2011) On the role of complete lattices in mathematical morphology: from tool to uncertainty model. Inf Sci 181(10):1971–1988
10. Ronse C, Najman L, Decencière E (eds) (2005) Mathematical morphology: 40 years on 7th international symposium on mathematical morphology, vol 30

11. Salembier P, Wilkinson MHF (2009) Connected operators: a review of region-based morphological image processing techniques. *IEEE Signal Process Mag* 26(6):136–157
12. Vizilter YV (2002) Design of morphological operators based on selective morphology. In: Dougherty ER, Astola JT, Egiazarian KO (eds) *Proceedings of SPIE—the international society for optical engineering image processing: algorithms and systems*, pp 215–226
13. Lantuéjoul Ch (1977) Sur le modèle de Johnson-Mehl generalize. Internal report of the Centre de Morph. Math., Fontainebleau, France
14. Aichholzer O, Aurenhammer F (1996) Straight skeletons for general polygonal figures in the plane. In: Cai JY, Wong CK (eds) *Computing and combinatorics*, vol 1090. LNCS Springer, pp 117–126
15. Blum H (1967) A transformation for extracting new descriptors of shape. In: Wathen-Dunn W (ed) *Models for the perception of speech and visual form*, pp 362–380
16. Costa L, Cesar R (2001) *Shape analysis and classification*. CRC Press, USA
17. Siddiqi K, Pizer SM (2008) *Medial representations: mathematics, algorithms and applications*. Springer, Berlin
18. Mestetskiy L (2006) Skeletonization of a multiply connected polygonal domain based on its boundary adjacent tree. *Siberian J Numer Math* 9(3):299–314 (in Russian)
19. Mestetskiy L (2007) Shape comparison of flexible objects—similarity of palm silhouettes. In: 2nd international conference on computer vision theory and applications VISAPP'2007, pp 390–393
20. Mestetskiy L (2009) Continuous morphology of binary images: figures, skeletons and circulars. *Fizmatlit, Moscow* (in Russian)
21. Mestetskiy L (2010) Skeleton representation based on compound Bezier curves. In: 5th International conference on computer vision theory and applications VISAPP'2010, vol 1. INSTICC Press, pp 44–51
22. Mestetskiy L, Semenov A (2008) Binary image skeleton—continuous approach. In: 3rd international conference on computer vision theory and applications VISAPP'2008, vol 1. INSTICC Press, pp 251–258
23. Deng W, Iyengar S, Brener N (2000) A fast parallel thinning algorithm for the binary image skeletonization. *Int J High Perform Comput Appl* 14(1):65–81
24. Drysdale R, Lee D (1978) Generalized Voronoi diagrams in the plane. In: 16th Ann Allerton conference on communications, control and computing, pp 833–842
25. Kirkpatrick D (1979) Efficient computation of continuous skeletons. In: 20th Ann IEEE symposium foundations of computer science, pp 18–27
26. Fortune S (1987) A sweepline algorithm for Voronoi diagrams. *Algorithmica* 2:153–174
27. Yap C (1987) An $O(n \log n)$ algorithm for the Voronoi diagram of the set of simple curve segments. *Discrete Comput Geom* 2:365–393
28. Lee D (1982) Medial axis transformation of a planar shape. *IEEE Trans Pattern Anal Mach Intell PAMI-4* 4:363–369
29. Lee DT, Schachter BJ (1980) Two algorithms for constructing a Delaunay triangulation. *Int J Comput Inf Sci* 9(3):219–242
30. Manzanera A, Bernard T, Preteux F, Longuet B (1999) Ultra-fast skeleton based on an isotropic fully parallel algorithm. In: Bertrand G, Couprie M, Perrotin L (eds) *Discrete geometry for computer imagery*, vol 1568. LNCS Springer, Berlin, pp 313–324
31. Karavelas MI (2006) Voronoi diagrams in CGAL. In: 22nd European workshop on computational geometry, pp 229–232
32. Srinivasan V, Nackman L, Tang J, Meshkat S (1992) Automatic mesh generation using the symmetric axis transform of polygonal domains. *Proc IEEE* 80(9):1485–1501
33. Ogniewicz R, Kubler O (1995) Hierarchic Voronoi skeletons. *Pattern Recogn* 28(3):343–359
34. Strzodka R, Telea A (2004) Generalized distance transforms and skeletons in graphics hardware. *Joint eurographics—IEEE TCVG symposium on visualization*
35. Maragos P (1989) Pattern spectrum and multiscale shape representation. *IEEE Trans Pattern Anal Mach Intell* 11:701–715

36. Suruliandi A, Ramar K (2008) Local texture patterns—a univariate texture model for classification of images. In: 16th IEEE International conference on advanced computing and communications, ADCOM 2008, pp 32–39
37. Asano A (1999) Texture analysis using morphological pattern spectrum and optimization of structuring elements. In: 10th international conference on image analysis and processing, ICIAP '99, pp 209–214
38. Mestetskiy LM (2009) Continuous morphology of the binary images. The figures. Skeletons. Circulars. Moscow Physmatlit (in Russian)
39. Shih FY, Mitchell OR (1991) Decomposition of gray-scale morphological structuring elements. *Pattern Recogn* 24(3):195–203
40. Wilkinson MHF (2002) Generalized pattern spectra sensitive to spatial information. In: 16th international conference pattern recognition, vol 1, pp 21–24
41. Urbach ER, Roerdink JBTM, Wilkinson MHF (2007) Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. *IEEE Trans Pattern Anal Mach Intell* 29(2):272–285
42. van Herk M (1992) A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recogn Lett* 13(7):517–521
43. Liang EH, Wong EK (1993) Hierarchical algorithms for morphological image processing. *Pattern Recogn* 26(4):511–529
44. Park H, Chin RT (1995) Decomposition of arbitrarily shaped morphological structuring elements. *IEEE Trans Pattern Anal Mach Intell* 17(1):2–15
45. Soille P, Breen E, Jones R (1996) Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. *IEEE Trans Pattern Anal Mach Intell* 18(5):562–567
46. Van Droogenbroeck M, Talbot H (1996) Fast computation of morphological operations with arbitrary structuring elements. *Pattern Recogn Lett* 17(14):1451–1460
47. Anelli G, Broggi A, Destri G (1998) Decomposition of arbitrarily shaped binary morphological structuring elements using genetic algorithms. *IEEE Trans Pattern Anal Mach Intell* 20(2):217–224
48. Van Droogenbroeck M, Buckley M (2005) Morphological erosions and openings: fast algorithms based on anchors. *J Math Imag Vis* 22:121–142
49. Gil J, Kimmel R (2002) Efficient dilation, erosion, opening and closing algorithms. *IEEE Trans Pattern Anal Mach Intell* 24(12):1606–1617
50. Urbach ER, Wilkinson MHF (2008) Efficient 2-D gray-scale morphological transformations with arbitrary flat structuring elements. *IEEE Trans Image Proc* 17(1):1–8
51. Preparata F, Sheimos M (1985) *Computational geometry: an introduction*. Springer, New York, NY, USA
52. Vizilter YV, Sidiyakin SV, Rubis A Y (2011) Calculation of morphological spectra of flat figures with the use of continuous skeletal representation. In: 15th Russian conference on mathematical methods of pattern recognition, pp 416–420 (in Russian)
53. Vizilter YV, Zheltov SY, Laretina NA (2009) Projective morphology on the basis of the operators filtering and image segmentation, computable by the method of dynamic programming. *Vestnik Comput Inf Technol* 6:18–27 (in Russian)
54. Vizilter YV, Sidiyakin SV (2012) Calculation of morphological pattern spectra of gray scale images. *Vestnik Comput Inf Technol* 4:8–17 (in Russian)
55. Sidiyakin SV (2013) Morphological pattern spectra algorithm development for digital image and video sequences analysis. PhD thesis, Moscow (in Russian)
56. Zingl A (2012) A rasterizing algorithm for drawing curves. *Multimedia und Softwareentwicklung*. Technikum-Wien, Wien
57. Tikhonov AN (1983) *The theory of recovery signals* Moscow, Science (in Russian)
58. Pyt'ev Y (1975) Morphological notions in problems of image analysis. *Reports of USSR Academy of Science* 224(6):1283–1286 (in Russian)
59. Pyt'ev Y (1975) Projection-based image analysis. *Cybernetics* 3:130–139 (in Russian)
60. Pyt'ev Y (1983) Morphological image analysis. *Reports of USSR Academy of Science* 3:1061–1064 (in Russian)

61. Pyt'ev Y, Chulichkov A (2010) Morphological methods for image analysis. Fizmatlit Publisher, Moscow (in Russian)
62. Pyt'ev Yu (1993) Morphological image analysis. *Pattern Recogn Image Anal* 3(1):19–28
63. Pyt'ev Y (1997) The morphology of color (multispectral) images. *Pattern Recogn Image Anal* 7(4):467–473
64. Pyt'ev Y (1998) Methods for morphological analysis of color images. *Pattern Recogn Image Anal* 8(4):517–531
65. Antonjuk V (1984) Hardware and techniques for morphological analysis of experimental multidimensional signals. Ph.D thesis (in Russian)
66. Pyt'ev Yu, Kalinin A, Loginov E, Smolovik V (1998) Morphological analysis of color images in the Chebyshev and quadratic metrics. *Pattern Recogn Image Anal* 8(2):234–235
67. Pyt'ev Y, Kalinin A, Loginov E, Smolovik V (1998) Comparison of black-and-white and Lambertian morphologies in the problem of pattern recognition. *Pattern Recogn Image Anal* 8(2):239–241
68. Pyt'ev Y, Kalinin A, Loginov E, Smolovik V (1998) On the problem of object detection by black-and-white and color morphologies. *Pattern Recogn Image Anal* 8(4):532–536
69. Chulichkov A, Grachev E, Ustinin D, Cheremukhin E (2003) Metrological measurements and signal processing in SEM based on model of signal formation. *Microelectron Eng* 69 (2–4):555–564
70. Pyt'ev YP, Falomkin II, Chulichkov AI (2006) Morphological compression of grayscale images of text. *Pattern Recogn Image Anal* 16(3):523–528
71. Evsegneev SO, Pyt'ev YP (2006) Analysis and recognition of piecewise constant texture images. *Pattern Recogn Image Anal* 16(3):398–405
72. Falomkin II, Pyt'ev YP (2007) Algorithm of adaptive morphological filtering of images. *Pattern Recogn Image Anal* 17(3):408–420
73. Pyt'ev Y, Chulichkov A (2011) Methods of morphological image analysis. Bilateral Russian–Indian scientific workshop on emerging applications of computer vision
74. Visilter Y, Zheltov S, Stepanov A (1994) Shape analysis using Pyt'ev morphologic paradigm and its use in machine vision. *SPIE Proc* 2350:163–167
75. Vizilter Y, Zheltov S (2008) Projective morphologies for image analysis. In: 9th international conference on pattern recognition and image analysis: new information technologies (PRIA-9-2008), vol 2, pp 287–290
76. Vizilter Y, Zheltov S (2010) Image segmentation in the framework of projective morphology. In: 10th international conference on pattern recognition and image analysis: new information technologies (PRIA-10-2010)
77. Vizilter YV (2011) Development of applied computer vision systems using projective morphologies and evidence-based image analysis. Bilateral Russian–Indian scientific workshop on emerging applications of computer vision (EACV-2011), pp 82–94
78. Vizilter YV, Sidyakin SV, Rubis AY, Gorbatsevich V (2011) Skeleton-based morphological shape comparison. *Pattern Recogn Image Anal* 21(2):357–360
79. Vizilter YV, Zheltov SY (2009) The use of projective morphologies for object detection and identification in images. *J Comput Syst Sci Int* 48(2):282–294
80. Vizilter YV (2009) Design of data segmentation and data compression operators based on projective morphological decompositions. *J Comput Syst Sci Int* 48(3):415–429
81. Vizilter YV, Zheltov SY (2012) Geometrical correlation and matching of 2D image shapes. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci* 1–3:191–196
82. Pyt'ev YP (2013) Oblique projectors and relative forms in image morphology. *J Comput Math Phys* 53(1):21916–21937
83. Maes F, Collignon A, Vandermeulen D, Marchal G, Suetens P (1997) Multimodality image registration by maximization of mutual information. *IEEE Trans Med Imaging* 16 (2P):187–198
84. Vizilter Y, Zheltov S (2008) Projective morphologies and their application in structural analysis of digital images. *J Comput Syst Sci Int* 47(6):944–958

85. Vizilter YV, Rubis AY (2009) Metric space of image shapes. *Intell Inf Process IIP* 9:406–409 (in Russian)
86. Vizilter YV, Rubis AY (2013) Comparison of 2D image shape similarity measures. In: 11th international conference on pattern recognition and image analysis: new information technologies (PRIA-11-2013), vol 1, pp 345–348
87. Pavel M (1989) *Fundamentals of pattern recognition*. Marcel Dekker Inc., New York
88. Hough PVC (1962) *Methods, means for recognizing complex patterns*. U.S., patent 3069654
89. Ballard DH, Brown CM (1982) *Computer vision*. Prentice-Hall, Englewood Cliffs, New Jersey
90. Davies ER (1992) Locating objects from their point features using an optimised Hough-like accumulation technique. *Pattern Recogn* 13(2):113–121
91. Davies ER (1993) Computationally efficient Hough transform for 2-D object location. In: 4th conference on British machine vision association, vol 1, pp 259–268
92. Davies ER (2004) *Machine vision: theory, algorithms, practicalities*, 3rd edn. Academic Press, San Diego
93. Visilter Y, Zheltov S, Stepanov A (1996) Object detection and recognition using events-based image analysis. *SPIE Proc* 2823:184–195
94. Visilter Y, Zheltov S, Stepanov A (1996) Events-based image analysis for machine vision and digital photogrammetry. In: *ISPRS Proceedings of international archives of photogrammetry and remote sensing V.XXXI, Part B*, pp 898–902
95. Visilter Y, Zheltov S, Bondarenko AV, Ososkov MV, Morzhin AV (2010) *Image processing and analysis in machine vision applications*. Moscow Phismathkniga (in Russian)
96. Ballard DH (1981) Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recogn* 13(2):111–122
97. Visilter Y, Gorbatshevich V (2011) Morphological image analysis using dynamic programming and stacked representations. *Vestnik Comput Inf Technol* 3:7–15 (in Russian)
98. Ford L, Fulkerson D (1962) *Flows in networks*. Princeton University Press, Princeton
99. Greig D, Porteous B, Seheult A (1989) Exact maximum a posteriori estimation for binary images. *J Royal Stat Soc* 51(2):271–279
100. Boykov Y, Kolmogorov V (2003) Computing geodesics and minimal surfaces via graph cuts. *IEEE Int Conf Computer Vision (ICCV'2003)*, pp 26–33
101. Boykov Y, Kolmogorov V (2004) An experimental comparison of min-cut/max-ow algorithms for energy minimization in vision. *IEEE Trans Pattern Anal Mach Intell (PAMI)* 26(9):1124–1137
102. Darbon J, Sigelle M (2006) Image restoration with discrete constrained total variation part i: fast and exact optimization. *J Math Imaging Vision* 26(3):261–276
103. Darbon J, Sigelle M (2006) Image restoration with discrete constrained total variation part ii: levelable functions, convex and non-convex cases. *J Math Imaging Vision* 26(3):277–291
104. Zheltov SY, Vizilter YV (2004) Robust computer image analysis for flight vehicles navigation and guidance. In: 16th IFAC symposium on automatic control in aerospace, vol 2, pp 164–167
105. Vizilter YV (2008) Generalized projective morphology. *Comput Opt* 32(4):384–399 (in Russian)

Chapter 3

Methods for Detecting of Structural Changes in Computer Vision Systems

Yury S. Radchenko and Aleksey V. Bulygin

Abstract The automation of experimental investigations based on video recording and different artificial vision applications often require that changes in a sequence of frames be detected without the observer's assistance. Variations in brightness, color, and size of an object are easily detectable using energy criteria. Nevertheless, some problems demand the use of algorithms capable of responding to small scale and texture changes of images. These problems can be solved by applying the criteria of Mean Structural Similarity Index Measure (MSSIM) and the developed Mean Nonparametric Structural Similarity Index Measure (MNSSIM), as well as the spectral algorithm for detecting structural changes in a frame, which have been used to good effect in video codec analysis. The profitable features of these criteria are their computational simplicity and their conformance to the human visual system. The criteria have not only a sensitivity for difference of comparing frames, but also have high stability of Gaussian and non-Gaussian (impulse) noises. This chapter describes the MSSIM, the own developed MNSSIM algorithms, and the spectral criterion, which provides the experimental confirmation of operating characteristics and features. The use of these criteria in automatic detection of changes in video captured scientific research scenes, the detection of motion or variable fragments in video frames in the intelligent video systems, and the application in video coding systems are discussed.

Keywords Image quality assessment • Spectral change • Pixel change • Substance phase transitions

Y.S. Radchenko (✉)

Raiophysic Department, Voronezh State University,
Universitetskaya pl. 1, Voronezh 394006, Russian Federation
e-mail: ysradchenko@yandex.ru; deanery@phys.vsu.ru

A.V. Bulygin

Voronezh State University, Universitetskaya pl. 1,
Voronezh 394006, Russian Federation
e-mail: alex987@bk.ru

3.1 Introduction

The automation of scientific researches based on video recording, smart vision, and video coding applications require that changes in a sequence of frames be detected without observer's assistance. Variations in object brightness, color, and size are easily detectable by Mean-Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) energy criteria [1]. However, a number of tasks require algorithms that react on structural (texture) variations of images. Generally, an image region with variable structure is formless.

Another problem of vision systems is a variation of observation conditions and also interference. The problem solution is based on the use of high robust and interference-immunity decision making algorithms. The detection of variations in the image segment structure is based on spectral and correlation analysis of spatial-temporal domain. At present, the quasi-optimum heuristic algorithms applying variations of correlation features exist. However, they are non-invariant to spectrum in various bases in relation to a segment movement and change of texture features.

The structural differences of images can be determined by various techniques. Figure 3.1 shows the criteria and the metrics being a basis to detect these differences. The authors analyzed difference estimation methods and algorithms for images presented by the numbered blocks.

The MSSIM criterion [2–4] and its modification MNSSIM [5] are the tri-criterion functionals that respond to the changes of brightness, contrast, and correlation features of image. Therefore, the MSSIM, the MNSSIM, and other modifications are the energy criteria for detecting image variations. They are sensitive to texture variations as well. The growing popularity of these criteria is proved by their quite appropriate compliance with the human vision system.

The image as a whole or its separate blocks can be expanded in a generalized Fourier series by using the system of orthogonal functions $\varphi_{km}(x, y)$. The following methods can be emphasized among a multitude of orthogonal basis: the Discrete Cosine Transform (DCT) [1] and its integer variant called pseudo-cosine transform [6], Walsh-Hadamard transform [1], wavelet transform [7, 8]. In the researches [9–12], the class of discrete polynomial transforms and easily version discrete Chebyshev transform or the Generalized DCT (GDCT) were proposed. The GDCT has a number of special properties that allow the efficient image processing.

The current research has proved that the spectral algorithms can be a base to implement the image structural variation detectors, which are robust to the change of observation conditions and interference. Let us notice that the MNSSIM algorithms and spectral algorithms are quite simple in computation.

The next Sect. 3.2 covers examples of using the MSSIM and the MNSSIM. Section 3.3 provides the description of spectral criteria of structural image similarity. Spectral field variation detection is discussed in Sect. 3.4. Experimental confirmation of structural similarity criteria is situated in Sect. 3.5. Conclusion is drawn in Sect. 3.6.

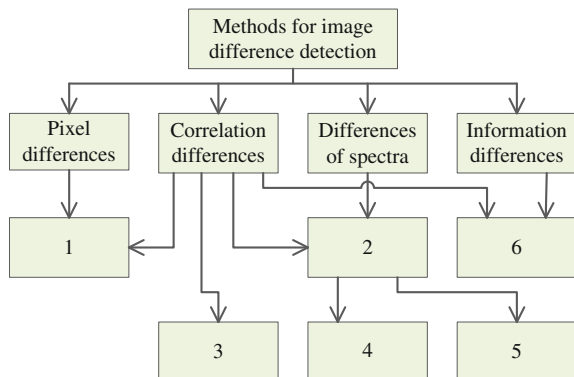


Fig. 3.1 Methods and algorithms for structural image difference detection, where 1 are algorithms based on MSSIM, MNSSIM1, and MNSSIM2 structural criteria, 2 are algorithms based on difference of generalized orthogonal basis spectra, 3 are algorithms based on difference of correlation features of moving video sequence fragments, 4 are spectral algorithms for texture anisotropy detection and estimation, 5 is a spectral algorithm for moving object detection, 6 are algorithms based on the Kullback and the Bhattacharyya information metrics

3.2 Pixel Structural Similarity Criteria

The task of detection and estimation for the structural similarity of two images having uncertain structures is a crucial issue in computer vision. The random images or frames of video sequence can be analyzed. This task cannot be formalized in full and has not yet been solved unambiguously. Objective structural similarity criteria may be classified in the following manner: single factor criteria, multi-factor criteria, and integral criteria being a combination of single factor and multi-factor criteria.

One of the simplest single factor criteria is a deviation of MSE. The MSE is determined by Eq. 3.1 for a separate image brightness or color component, where X, Y are images under a comparison ($X = (x_{ij}), Y = (y_{ij}), i = 1..n, j = 1..m$).

$$MSE(X, Y) = \frac{\sum_{i=1}^n \sum_{j=1}^m (x_{ij} - y_{ij})^2}{n \cdot m} \quad (3.1)$$

This criterion cannot be applied to human vision system. According to the MSE criterion, the images differ from each other, if the brightness reduces by 5 % only (the human vision system does not recognize this while the brightness options of different computer screens vary much more). At the same time, the images with the pronounced color variation of separate point, mild stripes, or frequency distortion resulting in a sharpness loss will be recognized as “almost unchanged”.

The widely used nowadays PSNR for a separate brightness or color component of an image is expressed by Eq. 3.2, where MAX_I is the maximum value assumed by the image component element.

$$PSNR(X, Y) = 10 \cdot \log \left(\frac{MAX_I^2}{MSE} \right) = 10 \cdot \log_{10} \frac{255^2 \cdot n \cdot m}{\sum_{i=1, j=1}^{n, m} (x_{ij} - y_{ij})^2} \quad (3.2)$$

For an RGB image, each component R, G, or B occupies 8 bits and, hence, $MAX_I = 2^8 - 1 = 255$ for this image. This measure is appropriate due to the logarithmic scale. It has the same drawbacks that root-mean-square deviation does [1, 13].

In a number of cases, the criterion should assume a variation of all image colors. The total PSNR for a full color RGB image $PSNR_{RGB}$ is calculated based on the summed squared error of the components provided by Eq. 3.3, where the maximum value is $\max S_{\Sigma}^2 = 3 \cdot 255^2 \cdot n^2 \cdot v$ ($n \times n$ is a number of pixels in a block, for the sake of simplicity the blocks are assumed to be square), v is a number of blocks, $\eta = 1 \dots v$ is a number of the current block, $(x_{i,j}^{(\eta)}, y_{i,j}^{(\eta)})_{R,G,B}$ are brightness of R, G, B components of pixels of blocks of images under comparison.

$$S_{\Sigma}^2 = \sum_{\eta=1}^v \left\{ \sum_{i,j=1}^n (x_{i,j}^{(\eta)} - y_{i,j}^{(\eta)})_R^2 + \sum_{i,j=1}^n (x_{i,j}^{(\eta)} - y_{i,j}^{(\eta)})_G^2 + \sum_{i,j=1}^n (x_{i,j}^{(\eta)} - y_{i,j}^{(\eta)})_B^2 \right\} \quad (3.3)$$

Hence, Eq. 3.3 can be re-written as Eq. 3.4.

$$PSNR_{RGB} = 10 \cdot \lg \left(\frac{\max S_{\Sigma}^2}{S_{\Sigma}^2} \right) = 10 \cdot \lg \left(\frac{3 \cdot 255^2 \cdot n^2 \cdot v}{S_{\Sigma}^2} \right) \quad (3.4)$$

The PSNR can be calculated for images in the YUV color format and other formats by equations similar to Eqs. 3.3–3.4. The combined criterion $PSNR_{RGB}$ provides the image similarity performance, which is more relevant for human vision. However, Eqs. 3.1–3.4 are slightly sensitive to texture image changes.

At the moment one of the criteria closest to the subjective perception of the recovered image quality, if the MSSIM [2–4, 14] characterizes a similarity of X and Y images by brightness, contrast, and structure, i.e. it is tri-factor. It appears as Eq. 3.5, where X_{η}, Y_{η} are the images compared in a block having number $\eta = 1 \dots v$, v is a number of blocks.

$$MSSIM = \frac{1}{v} \cdot \sum_{\eta=1}^v SSIM(X_{\eta}, Y_{\eta}) \quad (3.5)$$

The SSIM criterion is a block criterion. This means that it is applied not to the whole image at once but to its separate parts—equal blocks of the image, and later this value is averaged by all computed blocks producing the resulting MSSIM value for the whole image. In the general case, the $SSIM(X, Y)$ value for each block is calculated by Eq. 3.6, where $l(x, y)$ is a brightness comparison functional, $c(x, y)$ is a contrast comparison functional, $s(x, y)$ is a structure comparison functional, α, β, γ are the control coefficients.

$$SSIM(X, Y) = l(x, y)^\alpha c(x, y)^\beta s(x, y)^\gamma \quad (3.6)$$

In accordance with control coefficients $\alpha = \beta = \gamma = 1$ [4], the comparison functionals in the blocks are calculated by Eq. 3.7.

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (3.7)$$

Here variables have the following meanings:

– μ_x, μ_y are the sample mean for image blocks X_η and Y_η , respectively,

$$\mu_x = \frac{1}{N^2} \cdot \sum_{i,j=0}^{N-1} x_{ij} \quad \mu_y = \frac{1}{N^2} \cdot \sum_{i,j=0}^{N-1} y_{ij}.$$

– σ_x^2, σ_y^2 are sample variance for image blocks X_η and Y_η , respectively,

$$\sigma_x^2 = \frac{1}{N^2} \cdot \sum_{i,j=0}^{N-1} (x_{ij} - \mu_x)^2 \quad \sigma_y^2 = \frac{1}{N^2} \cdot \sum_{i,j=0}^{N-1} (y_{ij} - \mu_y)^2.$$

– σ_{xy} is a moment of correlation between image blocks X_η and Y_η

$$\sigma_{xy} = \frac{1}{N^2} \cdot \sum_{i,j=0}^{N-1} (x_{ij} - \mu_x)(y_{ij} - \mu_y).$$

Within constant C_3 , a functional $s(x, y)$ coincides with the Pearson's sample correlation coefficient. C_1, C_2, C_3 are small constants preventing incorrect behavior of the criterion when the moments are cleared. In accordance with [2–4] one can assume $C_1 = (0.01L)^2, C_2 = (0.03L)^2, C_3 = 0.5C_2, L$ is image bit width.

The MSSIM criterion assumes values from -1 to 1 . The value of 1 is obtained only in the case, when one and the same image is compared. This means that closer an image compared to the original image, closer the criterion value to 1 .

A selection of functional $s(x, y)$ in Eq. 3.7 as a measure of structural difference is mostly justified, when the comparing vectors have the values with multivariate Gaussian distribution. Therefore, the MSSIM criterion perfectly distinguishes textures in a form of Gaussian noise. When the laws of distortion distribution are unknown (non-Gaussian), it is reasonable to apply estimates of the respective structural characteristics based on nonparametric statistics [15]. To estimate mean brightness, it is efficient to use a sample median with higher stability as compared with the sample mean and one of the rank correlation coefficients instead of the Pearson's correlation coefficient. Two modifications of the structural similarity criterion based on the nonparametric MNSSIM1 and MNSSIM2 was proposed in [5, 7].

For MNSSIM1, the functionals below (Eq. 3.8) are suggested instead of Eq. 3.7

$$l(x, y) = \frac{2m_x m_y + C_1}{m_x^2 + m_y^2 + C_1} \quad c(x, y) = \frac{2\sigma m_x \sigma m_y + C_2}{\sigma m_x^2 + \sigma m_y^2 + C_2} \quad s(x, y) = Rs(x, y), \quad (3.8)$$

where $m_x = \text{mediana}(\vec{x})$ and $m_y = \text{mediana}(\vec{y})$ are medians of the brightness vectors in image blocks x and y , respectively, $\sigma m_x^2 = \text{mediana}[(\vec{x} - m_x)^2]$ and $\sigma m_y^2 = \text{mediana}[(\vec{y} - m_y)^2]$ are medians of the squared vector difference of brightness and the median, $Rs(x, y)$ is Spearman's rank correlation coefficient [15]. The Rs values change from -1 to 1 , while $Rs = 0$ means the absence of correlation.

The functionals (Eq. 3.9) are used in the MNSSIM2.

$$l(x, y) = \frac{2m_x m_y + C_1}{m_x^2 + m_y^2 + C_1} \quad c(x, y) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad s(x, y) = Rs(x, y) \quad (3.9)$$

In other words, a contrast comparison functional $c(x, y)$ should remain the same as for the MSSIM (Eq. 3.7). The structure comparison functional $s(x, y)$ and brightness comparison functional $l(x, y)$ should be used as those used in the MNSSIM1 (Eq. 3.8). Constants C_1 and C_2 from Eqs. 3.8–3.9 are identical to those used to calculate the MSSIM [2, 4]. The MNSSIM2 criterion is computationally simpler than the MNSSIM1.

Therefore, the MNSSIM1 and the MNSSIM2 criteria are the tri-factor criteria that use the nonparametric estimations of random field parameters. Nonparametric criteria of the MNSSIM1 and the MNSSIM2 structural similarities are practically identical to the MSSIM criterion at the presence of Gaussian distortion. However, if a point interference or other non-Gaussian statistics interference or a block distortion take place, then the MNSSIM1 and the MNSSIM2 metrics are better for human subjective vision than the estimation by the MSSIM.

3.3 Spectral Criteria of Structural Image Similarity

The image in a block can be expanded into a generalized Fourier series by the system of continuous $\varphi_{km}(x, y)$ or discrete orthogonal functions $\varphi_{km}(i, j)$ [10–12, 17–19].

Let us examine the main types and features of orthogonal transforms for continuous arguments. A sequence of functions $\{\varphi_k(z)\}$, $k = 0, 1, \dots, \infty$ is called orthonormal against $\rho(z)$ provided by Eq. 3.10, where δ_{mn} is a Kronecker symbol, z has no dimension value.

$$(\varphi_m, \varphi_n) = \int \varphi_m(z) \varphi_n(z) \rho(z) dz = \delta_{mn} \quad (3.10)$$

For weight function $\rho(z)$, ratios $\rho(z) \geq 0$ and $0 \leq \int_a^b \rho(z) dz < \infty$ exist. The system of functions $\{\varphi_k(z)\}, \{\psi_k(z)\}, k = 0, 1, \dots, k'$ is called bi-orthogonal against weight $\rho(z)$ (Eq. 3.11).

$$(\varphi_m, \psi_n) = \int \varphi_m(z) \psi_n(z) \rho(z) dz = \delta_{mn} \quad (3.11)$$

The full and closed set of functions $\{\varphi_k(z)\}$ is of great interest.

In the following Sects. 3.3.1 and 3.3.2, the polynomial transform and the discrete transforms are discussed, respectively.

3.3.1 Polynomial Transforms

An important class of functions $\{\varphi_k(z)\}$ with the properties of orthogonality, completeness and closure is orthogonal polynomials $p_k(z)$ satisfying the Eq. 3.12, where d_m is a norm of polynomial $p_m(z)$.

$$\int_a^b \rho(z) p_m(z) p_k(z) dz = d_m \delta_{km} \quad (3.12)$$

To expand the signals, one can apply the following conventional orthogonal polynomials [11, 20, 21]:

- The Hermitian polynomials $H_m(z)$, Eq. 3.13.

$$\rho(z) = \exp(-z^2) \quad p_m(z) = H_m(z) \quad -\infty < z < \infty \quad (3.13)$$

- The Laguerre polynomials, Eq. 3.14.

$$\rho(z) = z^\alpha \exp(-z) \quad p_m = L_m^\alpha(z) \quad 0 \leq z < \infty \quad (3.14)$$

- The Jacobi polynomials $p_m(z) = P_m^{(\alpha, \beta)}(z)$, Eq. 3.15.

$$\rho(z) = (1-z)^\alpha (1+z)^\beta \quad \alpha > -1 \quad \beta > -1 \quad -1 \leq z \leq 1 \quad p_m(z) = P_m^{(\alpha, \beta)}(z) \quad (3.15)$$

The Jacobi polynomials form a wide group of orthogonal polynomials. The important particular cases of Jacobi polynomials are:

- The Legendre polynomials

$$P_n(z) \quad (\alpha = \beta = 0) \quad \rho(z) = 1.$$

- The Chebyshev polynomials of the 1st and 2nd types provided by Eq. 3.16.

$$\begin{aligned} T_n(z) \quad (\alpha = \beta = -1/2) \quad \rho(z) &= (1 - z^2)^{-1/2} \\ U_n(z) \quad (\alpha = \beta = 1/2) \quad \rho(z) &= (1 - z^2)^{1/2} \end{aligned} \quad (3.16)$$

For practical use, the most convenient Jacobi polynomials are the polynomials of Legendre and Chebyshev of the 1st and 2nd types.

The Chebyshev polynomials of the 1st type have a number of useful properties so that their use to expand the signals is very attractive. The Chebyshev polynomial of the 1st type $T_m(z)$ associated with weight function $\rho(z) = 1/\sqrt{1-z^2}$ can be determined variously. One of the most appropriate means has a view of Eq. 3.17.

$$T_m(z) = \cos(m \cdot \arccos(z)) \quad (3.17)$$

For the Chebyshev polynomials $T_m(-z) = (-1)^m \cdot T_m(z)$. According to $T_{m+1}(z) = 2 \cdot z \cdot T_m(z) - T_{m-1}(z)$, one can get Eq. 3.18.

$$\begin{aligned} T_0(z) &= 1 \quad T_1(z) = z \quad T_2(z) = 2z^2 - 1 \quad T_3(z) = 4z^3 - 3z \\ T_4(z) &= 8z^4 - 8z^2 + 1 \end{aligned} \quad (3.18)$$

The orthogonality condition of functions $T_m(z)$ is provided by Eq. 3.18.

$$\int_{-1}^1 \frac{T_m(z)T_k(z)}{\sqrt{1-z^2}} dz = \begin{cases} 0 & k \neq m \\ \pi & k = m = 0 \\ \frac{\pi}{2} & k = m \neq 0 \end{cases} \quad (3.19)$$

Nulls of the Chebyshev polynomials are easily determined from Eq. 3.17, which can be re-written as Eq. 3.20.

$$T_m(z) = \cos(m \cdot \arccos(z)) = 0 \quad (3.20)$$

Based on the above equations, one can get Eq. 3.21.

$$z_k = \cos \frac{2k+1}{2m} \pi \quad k = 0, 1, 2, \dots, m-1 \quad (3.21)$$

Expansion of function $f(z)$ in the Chebyshev polynomials of the 1st type has a form of Eq. 3.22, where $d_m = \begin{cases} \pi/2 & m \neq 0 \\ \pi & m = 0 \end{cases}$ is a norm of the Chebyshev polynomials.

$$C_m = \frac{1}{d_m} \int_{-1}^1 \frac{f(z)T_m(z)}{\sqrt{1-z^2}} dz \quad (3.22)$$

$$f(z) = \sum_{m=0}^{\infty} C_m T_m(z)$$

The expansion in the Chebyshev polynomials of the 1st type $T_m(z)$ is the most converged among all possible expansions in degrees z^k , $k = 0, 1, \dots, \infty$.

The above relations are generalized for a 2D case. Let Ω be a region of the 2D Euclidian space and $z = (z_1, z_2)$ be a point in this space. The basis of function orthonormality can be defined by scalar product (Eq. 3.23).

$$\int_{\Omega} \rho(z) \varphi_{km}(z) \psi_{rn}(z) dz = \delta_{kr} \delta_{mn} \quad (3.23)$$

The system of functions $\varphi_{km}(z)$, $\psi_{rn}(z)$ is bi-orthogonal. The system of functions $\{\varphi_{km}(z), \psi_{rn}(z)\}$ depends on the form of weight function $\rho(z)$ and geometry of region Ω . Then for function $f(z)$ with the finite norm and weight $\rho(z)$, there are possible two equal presentations by Eqs. 3.24–3.25.

$$C_{km} = \int_{\Omega} \rho(z) f(z) \varphi_{km}(z) dz \quad f(z) = \sum_{k,m} C_{km} \psi_{km}(z) \quad (3.24)$$

$$B_{rn} = \int_{\Omega} \rho(z) f(z) \psi_{rn}(z) dz \quad f(z) = \sum_{r,n} B_{rn} \varphi_{rn}(z) \quad (3.25)$$

It should be noted, that a bi-orthogonality is a typical feature of multi-dimensional expansions.

Processing implementation is greatly simplified, when functions $\varphi_{km}(x, y) = \varphi_k(x) \varphi_m(y)$ are factorized. For a signal $f(x, y, \tau)$, a pair of transforms is performed by Eq. 3.26, where (x, y) is a non-normalized coordinates of field's point, τ is a vector parameter of shift, rotation and other affine transformations.

$$f(x, y, \tau) = \sum_m \sum_k C_{mk}(\tau) \varphi_m(x) \varphi_k(y) \quad C_{mk}(\tau) = \int \int_{\Omega} f(x, y, \tau) \varphi_m(x) \varphi_k(y) dx dy \quad (3.26)$$

If a_x, a_y denote a typical size of sub-region Ω , $z_1 = x/a_x$, $z_2 = y/a_y$, then Eq. 3.26 can be re-written by Eq. 3.27, where d_m is a norm of orthogonal $p_m(z)$ polynomials with weight $\rho(z)$.

$$f(x, y) = \sum_{m,k} C_{mk} p_m(x/a_x) p_k(y/a_y) \quad (3.27)$$

$$\begin{aligned} C_{mk} &= (d_m d_k)^{-1} \iint_{\Omega} f(a_x z_1, a_y z_2) \rho(z_1) p_m(z_1) \rho(z_2) p_k(z_2) dz_1 dz_2 \\ &= (d_m d_k)^{-1} \int \rho(z_1) p_m(z_1) dz_1 \int f(a_x z_1, a_y z_2) \rho(z_2) p_k(z_2) dz_2 \end{aligned} \quad (3.28)$$

It is obvious from Eqs. 3.27–3.28, that the spectral coefficients are determined by a sequential integration of x, y coordinates that is greatly simplify the computation.

3.3.2 Discrete Transforms

Let us analyze a generality and a difference of continuous and discrete transforms (Table 3.1). Here $\{\phi_k(n)\}$, $n = 0 \dots N-1$ is a complex basis of orthonormal vectors. In the general case, it is necessary to use a bi-orthogonal basis $\{\vec{\varphi}_k(n)\}$, $\{\vec{\psi}_k(n)\}$ to expand a signal.

Let us consider a vector-matrix transform notation. Let vector \mathbf{F} be a column of signal samples, as show in Eq. 3.29.

$$\mathbf{F} = \begin{pmatrix} f(0) \\ \vdots \\ f(N-1) \end{pmatrix} \quad (3.29)$$

The transform matrix can be written by Eq. 3.30.

$$\Phi = \begin{pmatrix} \varphi_0(0) & \varphi_0(1) & \dots & \varphi_0(N-1) \\ \varphi_1(0) & \varphi_1(1) & \dots & \varphi_1(N-1) \\ \dots & \dots & \dots & \dots \\ \varphi_{N-1}(0) & \varphi_{N-1}(1) & \dots & \varphi_{N-1}(N-1) \end{pmatrix} \quad (3.30)$$

A pair of signal-to-spectrum transform in the matrix form has a view of Eq. 3.31.

Table 3.1 The difference of continuous and discrete transforms

Continuous signal	Discrete signal
$x \in [0, T]$	$x = x_n = n\Delta T, n = 0 \dots N - 1$
$f(x)$	$f(n) = f(x_n)$
$\varphi_k(x)$	$\vec{\varphi}_k = (\varphi_k(0) \varphi_k(1) \varphi_k(2) \dots \varphi_k(N - 1))$
$(\varphi_k, \varphi_m) = \int_T \varphi_k(x) \varphi_k^*(x) dx = \delta_{km}$	$(\vec{\varphi}_k, \vec{\varphi}_m) = \sum_{n=0}^{N-1} \varphi_k(n) \varphi_m^*(n) = \delta_{km}$
$C_k = \int_T f(x) \varphi_k^*(x) dx$	$C_k = \sum_{n=0}^{N-1} f(n) \varphi_k^*(n)$
$f(x) = \sum_{k=0}^{\infty} C_k \varphi_k(x)$	$f(n) = \sum_{k=0}^{\infty} C_k \varphi_k(n)$

$$\mathbf{C} = \mathbf{\Phi F} \quad \mathbf{F} = \mathbf{\Phi}^{-1} \mathbf{C} \Rightarrow \mathbf{\Phi}^T \mathbf{C} \quad (3.31)$$

If $\mathbf{\Phi}^{-1} = \mathbf{\Phi}^T$, $\det(\mathbf{\Phi}) = \pm 1$, then $\mathbf{\Phi}$ is an orthogonal matrix and any two lines of it are orthogonal vectors. For a bi-orthogonal matrix transforms, Eq. 3.32 can be written.

$$\mathbf{C} = \mathbf{\Phi F} \quad \mathbf{F} = \mathbf{\Psi C} \quad \mathbf{\Psi \Phi} = \mathbf{1} \quad (3.32)$$

Let us consider a discrete variant of 2D transform. In this case the function is setup on a 2D discrete point grid, Eq. 3.33.

$$f(x, y) \Rightarrow f(i, j) \quad i, j = 0 \dots N - 1 \quad (3.33)$$

The samples form a square matrix $\mathbf{F} = [f(i, j)]$. The same grid could be uniform or it can be formed non-uniformly by a special law.

A pair of transforms of a signal matrix to a spectral matrix is determined by Eq. 3.34.

$$C_{km} = \sum_{i,j} f(i, j) \varphi_{km}^*(i, j) \quad f(i, j) = \sum_{k,m} C_{km} \varphi_{km}(i, j) \quad (3.34)$$

If the discrete basis functions are factorized, then Eq. 3.35 is accomplished.

$$\varphi_{km}(i, j) = \varphi_k(i) \varphi_m(j) \quad (3.35)$$

Expansion and synthesis are reduced to serial operations in i and j . This operation is of the form below in the matrix form, Eq. 3.36.

$$\mathbf{C} = \mathbf{\Phi F \Phi}^T \quad \mathbf{F} = \mathbf{\Phi}^{-1} \mathbf{C} (\mathbf{\Phi}^{-1})^T \quad (3.36)$$

Let us consider the Discrete Fourier Transform (DFT). The basis orthogonal functions [1, 18, 19] have the form of Eq. 3.37.

$$\begin{aligned} \tilde{\varphi}_k(n) &= \exp\left(j2\pi\frac{kn}{N}\right) = W^{kn} \quad W = \exp\left(j\frac{2\pi}{N}\right) \quad n = 0 \dots N-1 \\ k &= 0 \dots N-1 \end{aligned} \quad (3.37)$$

The orthogonality condition and the functions' norm are defined by Eq. 3.38.

$$\sum_{n=0}^{N-1} W^{kn} W^{*ln} = \begin{cases} 0 & k \neq l \\ N & k = l \end{cases} \quad (3.38)$$

Then the pair of transforms with non-symmetric normalizing coefficients is computed by Eq. 3.39.

$$C_k = \frac{1}{N} \sum_{n=0}^{N-1} f(n) W^{kn} \quad f(n) = \sum_{k=0}^{N-1} C_k W^{-kn} \quad (3.39)$$

For the DFT, there are other variants of normalizing coefficients before the sum up, for example, the expressions from Eq. 3.40.

$$C_k = \sum_{n=0}^{N-1} f(n) W^{kn} \quad f(n) = \frac{1}{N} \sum_{k=0}^{N-1} C_k W^{-kn} \quad (3.40)$$

When the DFT is applied, it is essential that a discrete signal $f(n)$ is considered to be periodically extended with the period of N , while spectrum C_k is also discrete and periodic with the period of N . Therefore, a condition is imposed on border frequency Δf_m of continuous signal $f(t)$ and sampling step T : $\Delta f_m T \leq 0.5$. The DFT with reduced number of operations is called Fast Fourier Transform (FFT) [1, 17, 22]. If a signal $f(n)$ is true, then Eq. 3.41 is executed.

$$|C_k| = |C_{N-k}| \quad \arg(C_k) = -\arg(C_{N-k}) \quad (3.41)$$

Therefore, the FFT for such signal calculates only a half of spectral coefficients.

Let us consider the Discrete Cosine Transform (DCT) in two cases—one-dimensional and two-dimensional DCT.

One-dimensional DCT.

A non-normalized basis functions [1, 18, 19] are determined by Eq. 3.42, where $n = 0 \dots N-1$.

$$\tilde{\varphi}_k(n) = \cos\left(\pi k \frac{2n+1}{2N}\right) = \cos\left(\pi k \frac{n+0.5}{N}\right) \quad (3.42)$$

The parameter N may be both even and odd. However, the transform with even N [1] is often used in practice. The orthogonality condition for these functions is provided by Eq. 3.43.

$$\sum_{n=0}^{N-1} \tilde{\varphi}_k(n) \tilde{\varphi}_l(n) = \begin{cases} N & k = l = 0 \\ N/2 & k = l \neq 0 \\ 0 & k \neq l \end{cases} \quad (3.43)$$

During the calculation of a signal spectrum, the orthonormal basis functions is applied as it shown in Eq. 3.44.

$$\varphi_k(n) = \sqrt{\frac{2}{N}} \cdot g_k \cdot \cos\left(\pi k \frac{n+0.5}{N}\right) \quad g_k = \begin{cases} \sqrt{0.5} & k = 0 \\ 1 & k \neq 0 \end{cases} \quad (3.44)$$

In this case, the direct and the reverse DCTs are expressed by Eqs. 3.45–3.46.

$$C_k = \sqrt{\frac{2}{N}} \cdot g_k \sum_{n=0}^{N-1} f(n) \cos\left(\pi k \frac{n+0.5}{N}\right) \quad (3.45)$$

$$f(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} g_k C_k \cos\left(\pi k \frac{n+0.5}{N}\right) \quad (3.46)$$

Two-Dimensional DCT.

The transforms with symmetric normalization of direct and reverse transform matrices have a view of Eq. 3.47, where $\mathbf{F} = [f(i, j)]$ is a source block and $\mathbf{C} = [C_{km}]$ is its spectrum.

$$\mathbf{C} = \mathbf{\Phi} \mathbf{F} \mathbf{\Phi}^T \quad \mathbf{F} = \mathbf{\Phi}^T \mathbf{C} \mathbf{\Phi} \quad (3.47)$$

The matrices of the direct $\mathbf{\Phi}$ and reverse DCT coincide, Eq. 3.48.

$$\mathbf{\Phi} = [\varphi_k(n)] = \sqrt{\frac{2}{N}} \begin{bmatrix} \sqrt{0.5} \\ \cos\left(\pi k \frac{(n+0.5)}{N}\right) \end{bmatrix} \quad \begin{bmatrix} k = 0 \\ k \neq 0 \end{bmatrix} \quad k = 0 \dots N-1 \\ n = 0 \dots N-1 \quad (3.48)$$

Matrix $\mathbf{\Phi}$ is square and orthogonal $\mathbf{\Phi}^{-1} = \mathbf{\Phi}^T$.

Integer cosine (pseudo-cosine) transform.

Standard H.264 applies a pseudo-cosine transform for blocks transformations. For 4×4 block, the DCT matrix is determined by Eq. 3.49, where $k, n = 0 \dots 3$.

$$\begin{aligned}
\Phi &= \sqrt{\frac{1}{2}} \cdot \begin{bmatrix} \sqrt{\frac{1}{2}} \\ \cos\left(\frac{(2n+1) \cdot k \cdot \pi}{2 \cdot 4}\right) \end{bmatrix} \\
&= \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ \sqrt{1+\frac{1}{\sqrt{2}}} & \sqrt{1-\frac{1}{\sqrt{2}}} & -\sqrt{1-\frac{1}{\sqrt{2}}} & -\sqrt{1+\frac{1}{\sqrt{2}}} \\ 1 & -1 & -1 & 1 \\ \sqrt{1-\frac{1}{\sqrt{2}}} & -\sqrt{1+\frac{1}{\sqrt{2}}} & \sqrt{1+\frac{1}{\sqrt{2}}} & -\sqrt{1-\frac{1}{\sqrt{2}}} \end{bmatrix} \quad (3.49)
\end{aligned}$$

It is proposed to use integer matrix \mathbf{H} [6] instead of matrix Φ , Eq. 3.50.

$$\mathbf{H} = \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad (3.50)$$

On the one hand, such replacement speeds up the integer operations of digital signal transform. On the other hand, this breaks a transform orthogonality. Such break should be compensated by the extra transforms in further steps.

The Hadamard system for Walsh functions.

The Walsh-Hadamard discrete basis is appropriate for computing. It is based on the Hadamard matrix. The matrix is calculated by the recurrent scheme, Eqs. 3.51–3.52.

$$\mathbf{A}_{2^n} = \begin{pmatrix} \mathbf{A}_{2^{(n-1)}} & \mathbf{A}_{2^{(n-1)}} \\ \mathbf{A}_{2^{(n-1)}} & -\mathbf{A}_{2^{(n-1)}} \end{pmatrix} \quad (3.51)$$

$$\mathbf{A}_1 = 1 \quad \mathbf{A}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \mathbf{A}_4 = \begin{pmatrix} \mathbf{A}_2 & \mathbf{A}_2 \\ \mathbf{A}_2 & -\mathbf{A}_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \quad (3.52)$$

Any line (column) of the Hadamard matrix is a discrete sample of the Walsh function of any order. The Hadamard matrix structure results in that only signal sample summation and subtraction operations are performed during the orthogonal transform. However, a convergence rate of series by the Walsh-Hadamard basis is less than the DCT provides. Besides, the spectrums are often non-monotonous, when this basis is applied.

Discrete Chebyshev GDCT.

The above mentioned discrete orthogonal transforms apply a uniform signal sample grid. However, a sample grid with a special non-uniformity allows to get the fast converged generalized Fourier series. The discrete Chebyshev transform called

Generalized DCT (GDCT) [9–12] belongs to this type of transforms. It is a particular case of transforms in the orthogonal polynomials.

To calculate integrals in the orthogonal polynomial transforms, it is proposed to use a Gaussian type quadrature formula with the highest algebraic accuracy [20] expressed by Eq. 3.53, where z_i are nulls of orthogonal polynomial $p_N(z)$ with weight $\rho(z)$, λ_i are the Christoffel numbers.

$$\int f(z)\rho(z)dz = \sum_{i=1}^N \lambda_i f(z_i) \quad (3.53)$$

Knots and weights $\{z_i\}, \{\lambda_i\}$ are clearly defined by the form of polynomial $p_N(z)$. In general, Eq. 3.53 expands in orthogonal polynomial system. A particular case of Eq. 3.53, when the Chebyshev polynomials are used, is called the Gauss-Chebyshev (Meller) formula [20] presented in a view of Eq. 3.54 form, where $z_i = \cos(\pi(i + 0.5)/N)$ are nulls of Chebyshev polynomial $T_N(z) = 0$, $\lambda_i = \pi/N = \text{const}$.

$$\int \frac{f(z)}{\sqrt{1-z^2}} dz = \frac{\pi}{N} \sum_{i=0}^{N-1} f(z_i) \quad (3.54)$$

For the Chebyshev polynomials, the direct and the reverse transforms (Eq. 3.22) can be applied (Eq. 3.54) (one-dimensional variant for normalized interval $z \in [-1, 1]$). Then Eqs. 3.55–3.56 will be received [9, 11], where $g_m = 1$ with $m > 0$ and $g_m = \sqrt{0.5}$ with $m = 0$.

$$C_m = g_m \cdot \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} f(z_i) \cdot \cos(\pi m \frac{i+0.5}{N}) \quad (3.55)$$

$$Y_M(z) = \sqrt{\frac{2}{N}} \sum_{m=0}^{M-1} g_m \cdot C_m \cdot \cos(m \cdot \arccos(z)) \quad (3.56)$$

In accordance with Eqs. 3.55–3.56, the sample points and the Chebyshev samples $z_i = \cos(\pi(i + 0.5)/N)$ of signal $f(z)$ are taken non-uniformly (Fig. 3.2). A synthesis (recovery) of signal $Y(z)$ by M spectral components is performed in random point $z \in [-1, 1]$, but not within a discrete set of sample points as this is done in the DCT. During recovery, any sample grid can be used, for example, uniform $z_n = 2n/(L - 1) - 1$, $n = 0 \dots L - 1$, if $L \neq N$. The recovered image is a subject to geometric scaling to downwards the size and upwards the size as well.

In the case of 2D GDCT $N \times N$, the Chebyshev samples of the image are taken within the sample block of $M1 \times M1$ points (pixels) by Eq. 3.57.

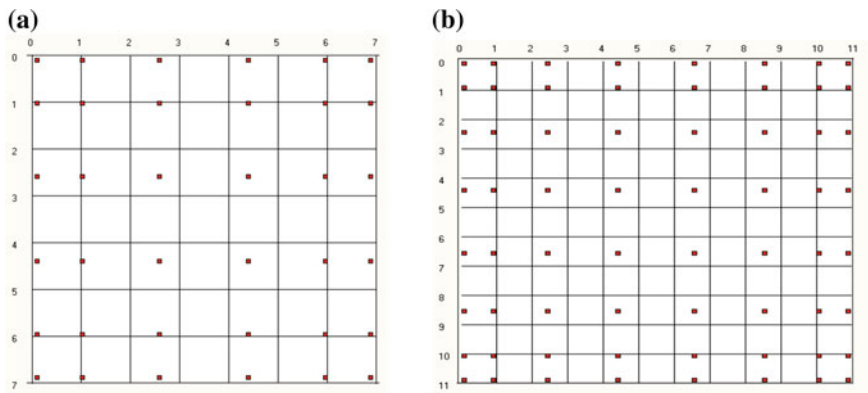


Fig. 3.2 The Chebyshev block sampling: **a** $N1/N = 8/6$, **b** $N1/N = 12/8$

$$\begin{aligned} x_i &= 0.5(N1 - 1) \cdot (1 + \cos(\pi(i + 0.5)/N)) \\ y_j &= 0.5 \cdot (N1 - 1) \cdot (1 + \cos(\pi \cdot (j + 0.5)/N)) \\ i, j &= 0 \dots N - 1 \end{aligned} \quad (3.57)$$

This matrix is transformed to spectral coefficient matrix \mathbf{C} of $M \times M$ size with the use of direct transform rectangular matrix of $M \times N$ size. In the case of reverse transform, a rectangular matrix of $L \times M$ size can be applied. That means that a recovered block $\mathbf{R} = [r_{ij}]$ ($i, j = 1 \dots L$) size is $L \times L$. The Chebyshev samples can be obtained by using linear interpolation by the nearest pixels [20, 23].

The direct and the reverse Chebyshev transforms (GDCT) in the matrix form are defined by the operations Eq. 3.58, i.e. the transform Eq. 3.36 falls into the bi-orthogonal transform class.

$$\mathbf{C} = \mathbf{\Phi}\mathbf{F}\mathbf{\Phi}^T \quad \mathbf{F} = \mathbf{\Psi}^T\mathbf{C}\mathbf{\Psi} \quad (3.58)$$

The matrix $\mathbf{\Phi}$ is a direct transform matrix (Eq. 3.59).

$$\mathbf{\Phi} = \sqrt{\frac{2}{N}} \cdot \begin{bmatrix} \sqrt{0.5} \\ \cos\left(\pi \frac{(i+0.5) \cdot m}{N}\right) \end{bmatrix}, \quad m = 0 \dots M - 1 \quad i = 0 \dots N - 1 \quad (3.59)$$

The matrix $\mathbf{\Psi}$ is a reverse transform matrix (Eq. 3.60).

$$\mathbf{\Psi} = \sqrt{\frac{2}{N}} \cdot \begin{bmatrix} \sqrt{0.5} \\ \cos\left(m \cdot \arccos\left(\frac{2n}{L-1} - 1\right)\right) \end{bmatrix}, \quad m = 0 \dots M - 1 \quad n = 0 \dots L - 1 \quad (3.60)$$

It is obvious from Eqs. 3.59–3.60, that in the general case the GDCT matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are rectangular. The direct GDCT matrix coincides with the direct DCT

matrix (Eq. 3.48) with $M = N$. The reverse transforms differ from each other. It should be noted that in the DCT, the transform matrices are square and have the size of $(N \times N = M \times M)$ as opposed to the GDCT, where M does not equal N in the general case [11, 12]. The GDCT is much close to the ideal de-correlation Karhunen-Loeve [1, 19] transform among all analyzed orthogonal transforms.

3.4 Spectral Image Variation Detection

Let a field $f^{(i)}(\mathbf{r}, t_i)$ being fragment $u(\mathbf{r})I_\Omega(\mathbf{r})\delta(t - t_i)$ of a space-time signal (dynamic image) at discrete time moment t_i , $i = 0, 1, \dots$ be observed in a sub-region $\mathbf{r} = (x, y) \in \Omega$ (a block of any frame). Here, $I_\Omega(\mathbf{r})$ is a sub-region indication function. Additionally let us assume that the white Gaussian noise $\eta(x, y) \equiv \eta(\mathbf{r})$ is available into an image. After comparing a separate block of the $(i-1)$ th and i th frames, one can hypothesize (Eq. 3.61).

$$\begin{aligned} H_0 : \xi(\mathbf{r}) &= \mathbf{f}_1(\mathbf{r}) + \eta(\mathbf{r}) & \mathbf{f}_1(\mathbf{r}) &= \mathbf{f}^{(i)}(\mathbf{r}) = \mathbf{f}^{(i-1)}(\mathbf{r}) \\ H_1 : \xi(\mathbf{r}) &= \mathbf{f}_2(\mathbf{r}) + \eta(\mathbf{r}) & \mathbf{f}_2(\mathbf{r}) &= \mathbf{f}^{(i)}(\mathbf{r}) \neq \mathbf{f}_1^{(i-1)}(\mathbf{r}) \end{aligned} \quad (3.61)$$

Basing on observation of $\xi(\mathbf{r})$, it is necessary to accept or reject the main hypothesis about a block image invariance. If two images are compared rather than video sequence frames are observed, then $f_1(\mathbf{r})$ is an unvaried texture in frames and $f_2(\mathbf{r})$ is a texture different from $f_1(\mathbf{r})$.

After checking the sophisticated hypotheses, signal $f_1(\mathbf{r})$, $f_2(\mathbf{r})$ are formed, and a structure uncertainty arises. Generally, the processing of the unknown form signal cannot be solved without the use of some additional factors. One of the most convenient ways to solve the a priori uncertainty is a signal parameterization. In this case, a signal form uncertainty transforms to a parametric uncertainty, which resolving ways are well designed [13, 19]. The convenient signal parameterization method has its presentation in the form of generalized Fourier series by any orthonormal basis φ_{km} [1, 10, 17, 19].

For different hypotheses, the signals can be presented by Eq. 3.62, where $\langle \eta_{k,m} \rangle = 0$, $D(\eta_{k,m}) = \frac{N_0}{2} \cdot \|\varphi_{km}\|^2$.

$$\begin{aligned} f_1(\mathbf{r}) &= \sum_{k,m} C_{km}^{(1)} \varphi_{km}(\mathbf{r}) & f_2(\mathbf{r}) &= \sum_{k,m} C_{km}^{(2)} \varphi_{km}(\mathbf{r}) \\ \xi(\mathbf{r}) &= \sum_{k,m} X_{km} \varphi_{km}(\mathbf{r}) & \eta(\mathbf{r}) &= \sum_{k,m} \eta_{km} \varphi_{km}(\mathbf{r}) \end{aligned} \quad (3.62)$$

The spectral coefficients for expansion of $C_{k,m}^{(2)}$ are assumed to be unknown. The coefficients $C_{k,m}^{(1)}$ are reference and defined.

Therefore, the hypotheses in a spectral definition are checked by Eq. 3.63.

$$\begin{aligned} H_0 : X_{km} &= C_{km}^{(1)} + \eta_{km} \\ H_1 : X_{km} &= C_{km}^{(2)} + \eta_{km} \end{aligned} \quad (3.63)$$

Section 3.4.1 provides the optimal detection algorithm and quasi-optimal algorithms are discussed in Sect. 3.4.2.

3.4.1 Optimal Detection Algorithm

The maximum likelihood algorithm [17, 24] is an asymptotically optimum rule for hypothesis check provided by Eq. 3.64, where $L(\cdot)$ is a log likelihood ratio functional determined by Eq. 3.65.

$$\max_{C^{(2)}} L(X|C^{(2)}) \underset{>}{<} h_0 \quad (3.64)$$

$$L(X|C^{(2)}) = \ln \left[\frac{W[X_{k,m}|f_2(r|C^{(2)})]}{W[X_{k,m}|f_1(r|C^{(1)})]} \right] \quad (3.65)$$

According to Eq. 3.64 it is required to determine the absolute maximum of log likelihood ratio functional by an unknown vector of parameters $C^{(2)}$ and make a non-randomized decision in favor of the respective hypothesis.

Now our selected spectral signal form parameterization (Eq. 3.62) is taken into account. To solve the hypothesis, let us check task by Eq. 3.66.

$$L(X|C^{(2)}) = (1/N_0) \left\{ 2 \sum_{k,m} X_{km} (C_{km}^{(1)} - C_{km}^{(2)}) - \sum_{k,m} \left((C_{km}^{(1)})^2 - (C_{km}^{(2)})^2 \right) \right\} \quad (3.66)$$

The following maximization by $C_{k,m}^{(2)}$ the decision rule [17] is obtained by Eq. 3.67.

$$(1/N_0) \sum_{k,m} \left(X_{k,m} - C_{k,m}^{(1)} \right)^2 \underset{>}{<} h_0 \quad (3.67)$$

3.4.2 Quasi-optimal Algorithms

Due to the fact that N_0 determines the interference power and typically is unknown in practice, the normalized statistics can be used, They may be written in the general form [17] by Eq. 3.68.

$$D_0 = \frac{\sum_{k,m} (X_{k,m} - C_{k,m}^{(1)})^2}{(C_{0,0}^{(1)})^2} \text{ or } D_E = \frac{\sum_{k,m} (X_{k,m} - C_{k,m}^{(1)})^2}{\sum_{k,m} (C_{k,m}^{(1)})^2} \quad (3.68)$$

As $C_{00}^{(1)}$ is proportional to mean image brightness in the frame block and $\sum (C_{km}^{(1)})^2$ is a block energy, the statistics D_0, D_E are stable to block brightness variation, i.e. to observation conditions. Taking into account Eq. 3.68, the decision rule (Eq. 3.64) is written by Eq. 3.69.

$$D_0 \begin{matrix} \gamma_1 \\ > \\ < \\ \gamma_0 \end{matrix} h_0 \text{ or } D_E \begin{matrix} \gamma_1 \\ > \\ < \\ \gamma_0 \end{matrix} h_E \quad (3.69)$$

The number of summable summands and order of their selection in statistics D_0, D_E can be setup from peculiarities of a certain task.

3.5 Experimental Research of Structural Similarity Algorithms

Let us consider the applications of structural similarity algorithms for image analysis (Sect. 3.5.1), the experimental research of spectral statistics (Sect. 3.5.2), and the experimental research of MSSIM and MNSSIM1(2) criteria (Sect. 3.5.3).

3.5.1 Practical Using of Pixel and Spectral Algorithms in Image Analysis

The MSSIM, MNSSIM1, and two structural similarity criteria, the D_0, D_E spectral criteria were experimentally studied. Such performance as sensitivity to texture variation, interference tolerance, robustness to observation conditions and decision making threshold selection were tested [16].

In our research for detection of substance phase transitions, when heating, the 75-frames from video sequence were selected. It was taken by Infinity 1-3C digital camera during an experiment on heating cesium chloride (CsCl) sample within the temperature range of (250; 710)°C. (Each frame represents the substance at a certain temperature and temperature variation differs from frame to frame).

Two phase transitions take place in the CsCl sample into this temperature range. The sample surface texture changes sharply (stepwise), and, hence, a texture of successive frames corresponding to this transition also changes abruptly. Figure 3.3

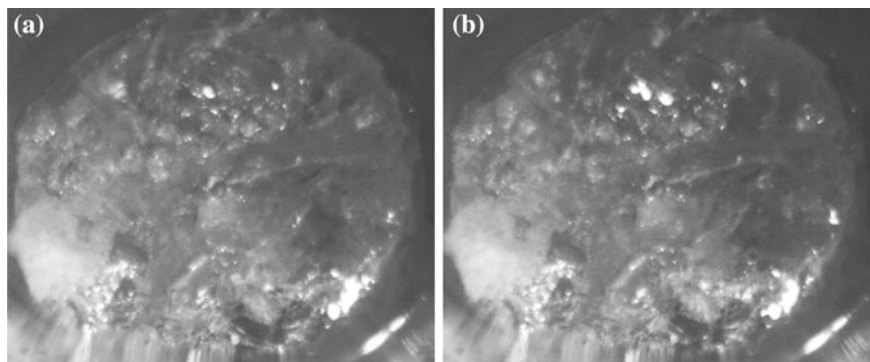


Fig. 3.3 Fragment of video sequence corresponding to the phase transition (polymorphous transformation) of the CsCl sample at temperatures: **a** 462.8 °C; **b** 468.4 °C

presents the images with small visual differences in the texture of some regions (blocks), which correspond to the first effect, polymorphous transformation of the substance (i.e., transition from one solid state to another followed by reconstructive re-arrangement of the crystal mosaic structure or “grains” in the image).

Figure 3.4 presents the images with noticeable visual differences, which correspond to the second observed effect, substance melting (transition from the solid state to the liquid one followed by crystal fracture and disappearance of grains in the image).

At the phase of transition moments, this image structure should have a step. The MSSIM index, its modification MNSSIM1(2) and spectral metric for detecting changes in video sequence D_0 were used.

Figure 3.5 shows the $MSSIM(T)$, the $MNSSIM1(T)$ and the $MNSSIM2(T)$ plots. Figure 3.6 demonstrates the $D_0(T)$ plot. All these dependences have two clearly

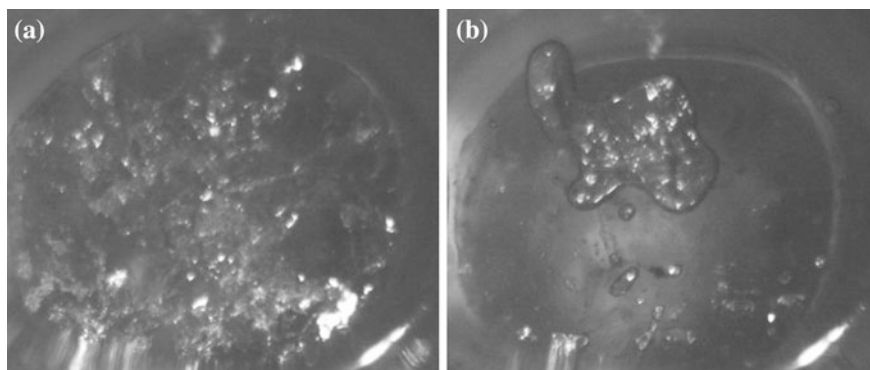


Fig. 3.4 Fragment of video sequence corresponding to the phase transition (melting) of the CsCl sample at temperatures: **a** 642.7 °C; **b** 645.2 °C

discernible peaks (extrema) corresponding to the phase transitions temperatures. It is the left jump that has an important physical meaning, since it represents a phase transition (significant changes appeared); and the right jump means that the changes ceased. The fluctuations (small peaks) are explained by the random changes in frame brightness as well as random changes of the substance structure. The oscillations (see $MSSIM(T)$, $MNSSIM1(T)$, $MNSSIM2(T)$ plot) after the second high peak at 645.2°C temperature can be attributed to the fact that, upon melting, the changes still proceed on the surface of the substance under observation, thus reflecting on the image texture. It is obvious from Fig. 3.5, that for the given video sequence, the $MSSIM$, the $MNSSIM1$, and the $MNSSIM2$ dependences are similar. Therefore, the $MNSSIM1$ and the $MNSSIM2$ plots are almost identical. However, if the analyzed video sequence is subjected to some distortion factors (e.g., pulse noise) during a video recording and an artificial improvement of images, then the $MNSSIM1$ and the $MNSSIM2$ criteria seem to be more preferable.

Peak values of the $MSSIM$, the $MNSSIM1$, the $MNSSIM2$ and D_0 , their mean values before the peaks (to estimate jumps) and temperatures of the peaks are provided in Tables 3.2 and 3.3.

The temperature values detected by the analysis of the sharp jumps in the $MSSIM(T)$ the $MNSSIM1(T)$, the $MNSSIM2(T)$, and the $D_0(T)$ dependences, correlate well with data from literature sources [16] $T_1 = 469^\circ\text{C}$, $T_2 = 645^\circ\text{C}$.

Figures 3.5 and 3.6 also result that the $MSSIM$ criterion and its $MNSSIM1(2)$ modification are inferior to spectral criterion the D_0 in the scale of jumps. In the peaks, the $MSSIM$ and the $MNSSIM1(2)$ curves change several-fold, whereas the D_0 curve changes several tens of times. This feature of the spectral criterion can be useful in automatic detection of process change moment. Even such a low automatically set threshold as $6 \div 8$ min (D_0) ensures that fluctuations of the image brightness and color not followed by texture changes will not belong to useful effects. Due to higher fluctuations of the $MSSIM$, the $MNSSIM1$ and the $MNSSIM2$ criteria, a threshold selection requires the additional data analysis by the researcher.

Fig. 3.5 The plots of $MSSIM(T)$, $MNSSIM1(T)$, $MNSSIM2(T)$

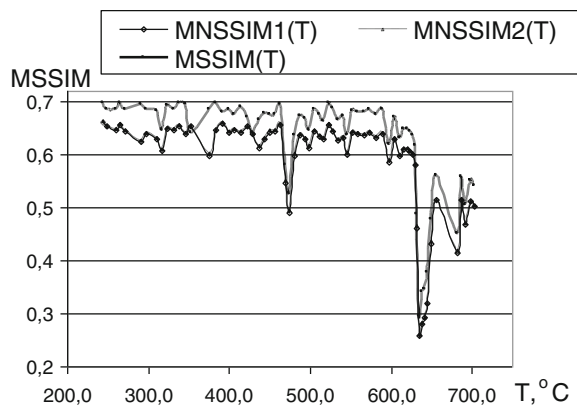


Fig. 3.6 The plot of $D_0(T)$

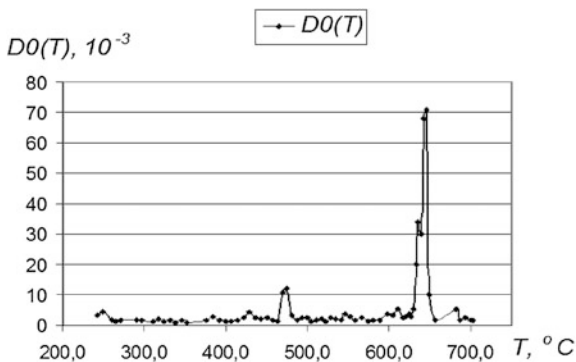


Table 3.2 Values before the peaks

Peak no	Peak temperature, °C	MSSIM peak value	MNSSIM1 peak value	MNSSIM2 peak value	$D_0, 10^{-3}$ peak value
1 peak	468.4	0.526	0.489	0.496	12
2 peak	645.2	0.296	0.259	0.261	70

Table 3.3 Peak values

Peak no	MSSIM mean value before peak	MNSSIM1 mean value before peak	MNSSIM2 mean value before peak	D_0 mean value before peak, 10^{-3}
1 peak	0.686	0.642	0.644	1
2 peak	0.668	0.626	0.628	1

3.5.2 Experimental Research of Spectral Statistics D_0 and D_E

To estimate the algorithm provided by Eq. 3.68, the experiments with real dynamic fields were done [17]. During experiment, values of statistics D_0 and D_E (Eq. 3.68) for block of brightness components of two diverted video sequence frames were calculated. Blocks characterizing by such change types as practically unvaried, weakly, or strongly varied were selected as blocks for analysis. Figure 3.7 shows an example of a frame from the analyzed dynamic sequence. The distinctive blocks, where statistics values were computed and differences were detected, are marked with boxes. The GDCT discrete transform is used.

Figure 3.7 shows a frame of video sequence with varied blocks (marked in boxes) for different thresholds: The results are tuned for $h_E = 0.01$ and $h_E = 0.04$

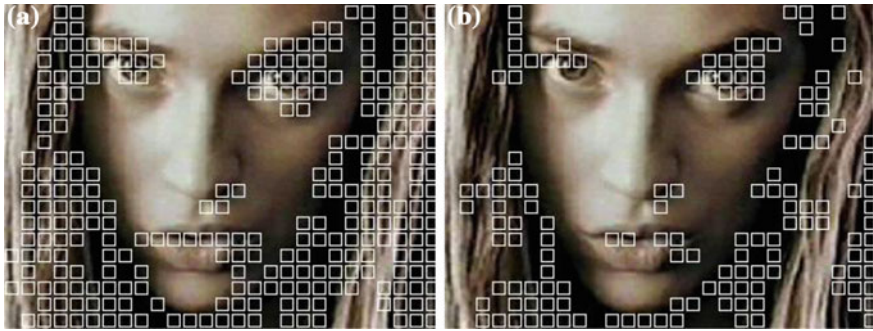


Fig. 3.7 Image frames with varied blocks obtained by the full spectrum: **a** $h_E = 0.01$, **b** $h_E = 0.04$

(Fig. 3.7a, b, respectively.) Four different blocks were analyzed and average statistics were calculated for each variation type. The obtained statistics D_0 and D_E for the brightness component for the image (Fig. 3.7) are presented in Table 3.4.

The given results prove that for blocks with low changes, metrics D_0 and D_E are quite close. Their difference from metrics for varied (flexible) blocks is 2–3 orders. This allows to select easily the threshold separating flexible and inactive blocks. It should be noted that metric D_0 change range is 1.7–2 times more than that metric D_E . Higher change range of statistics D_0 proves its preference. Thresholds h_0 and h_E are connected with the following relation:

$$h_0 = h_E \frac{\sum_{k,m} (C_{k,m}^{(1)})^2}{(C_{0,0}^{(1)})^2}$$

Table 3.5 presents the h_E thresholds averaged by frames, calculated thresholds \bar{h}_0 , and percentage of varied block.

It is evident from Table 3.5, that 10-fold threshold change results in number of blocks varied in less than 3 times. This phenomenon proves a non-criticality of threshold selection, which separates flexible and inactive blocks being a result of the above described test.

The possibility for use of decision rules (Eq. 3.68) with the truncated spectrum was analyzed in different video sequences. Our analysis demonstrated that detection

Table 3.4 Brightness component’s statistics for the image (Fig. 3.7)

Average statistics	Flexible fragments			Inactive fragments	
	Eyes	Lips	Hair	Cheeks	Forehead
$D_0 \cdot 10^3$	158.90878	74.04290	39.61026	0.82546	0.11304
$D_E \cdot 10^3$	91.14991	62.19916	25.68643	0.81213	0.11298

Table 3.5 Thresholds averaged by frame and percentage of block varied

Parameter	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7
h_E	0.005	0.010	0.020	0.025	0.030	0.040	0.050
\bar{h}_0	0.011	0.021	0.042	0.053	0.064	0.085	0.106
Varied blocks, %	60.56	50.74	38.52	33.15	30.00	26.11	21.11

can be performed by the truncated spectrum and even by one spectral component (mean block brightness). Figure 3.8 shows an example of video sequence frames with varied blocks defined by metrics D_E with the full spectrum ($N = 64$) (see Fig. 3.8a) and one spectral component ($N = 1$) (see Fig. 3.8b). A threshold value was $h_E = 0.01$ for both figures.

Table 3.6 provides the percentage of the changed blocks detecting by the whole spectrum ($D(N)$, %) and by one spectral component ($D(1)$, %) as well as by a relative ($\Delta D/D_N$) detection error by one spectral component.

Here, $\frac{\Delta D}{D(N)} = \frac{D(N) - D(1)}{D(N)}$. It is evident from the Table 3.6 that the maximum relative error of variation detection by one spectral component does not exceed 0.16. Therefore, the block changes can be satisfactorily detected by one spectral component. Test performed with test video sequences “Container”, “Foreman”, and “Suzie” demonstrated the similar results.

The spectral video sequence block variation detection algorithm has been used to implement the MGDCT video coding concept [25]. Our proposed concept applies the DCT/GDCT orthogonal transforms in a video codec structure.

To decrease a video codec bit-rate, the algorithm (Eq. 3.68, 3.69) was used to detect the reliable inter-frame video sequence variations. Figures 3.9–3.10 show video frames with marked blocks that have varied because of moving objects. Figure 3.9 applied “Container” video sequence, and Fig. 3.10 presents a frame of a remote video monitoring system. The detectors apply the GDCT transform.

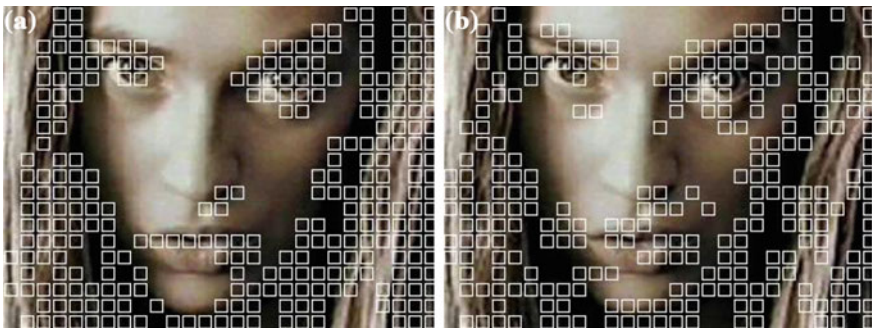
**Fig. 3.8** Image frames with varied blocks obtained by full and truncated spectrum: **a** full spectrum, **b** truncated spectrum

Table 3.6 Percentage of changed blocks

h_E	0.005	0.01	0.02	0.025	0.03	0.04	0.05
$D(N)$, %	60.56	50.74	38.52	33.15	30.00	26.11	21.11
$D(1)$, %	56.67	47.96	39.63	36.11	31.85	27.78	24.44
$\Delta D/D_N$	0.0642	0.0547	-0.0288	-0.0894	-0.0617	-0.0638	-0.1579

Combining the Chebyshev sampling and spectral detection allows to reduce information processing in 4–8 times depending on frame variations.

3.5.3 Experimental Research of MSSIM and MNSSIM1(2) Criteria

Notwithstanding that the MSSIM structural similarity criterion is one of the criteria closest to the human vision system; it has drawbacks of certain types. For example, in the case of image blur, pulse noise, or blocking, the MSSIM criterion provides values that are not quite similar with vision system. The MNSSIM 1(2) criteria have not such effects [5]. Let us investigate the MSSIM, MNSSIM1(2) criteria under various artifacts.

The MSSIM, MNSSIM1(2) criteria at pulse interference.

The image called “Lena” was used to test impact of pulse interference on the MSSIM, the MNSSIM1, and the MNSSIM2 criteria. Figure 3.11 shows images disturbed by pulse noise of “pepper” and “salt/pepper” types with probability $p = 0.05$, and the MSSIM, the MNSSIM1, and the MNSSIM2 criteria values corresponding to them.

A pulse noise was setup by two characteristics: an intensity determined by noise probability in pixel p and a noise type with three variants such as “salt”, “pepper”, and “salt/pepper”. The mathematical model of pulse noise for different noise peaks is determined as follows. Let $X = [x_{ij}]$ be a non-distorted image and $Y = [y_{ij}]$ be a distorted image. For the “pepper” type noise pulse, the model is setup by Eq. 3.70.

$$y_{ij} = \left\{ \begin{array}{l} 0 : \text{prob.} p \\ x_{ij} : \text{prob.} 1 - p \end{array} \right\} \quad (3.70)$$

For the “salt” type noise pulse, the model is setup by Eq. 3.71.

$$y_{ij} = \left\{ \begin{array}{l} 255 : \text{prob.} p \\ x_{ij} : \text{prob.} 1 - p \end{array} \right\} \quad (3.71)$$

For the “salt/pepper” type noise pulse, the model is setup by Eqs. 3.72–3.73.

Fig. 3.9 Spectral variation detection in “Container” test sequence

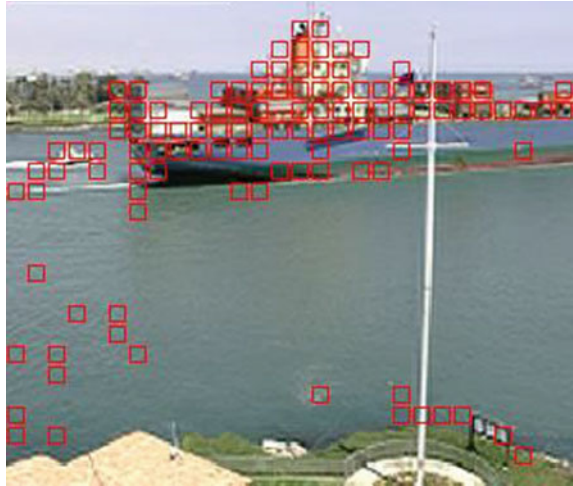


Fig. 3.10 Spectral variation detection in the video monitoring system



$$f_0 = \left\{ \begin{array}{l} 0, \text{prob.}0.5 : \text{noise} \backslash \text{pepper} \\ 255, \text{prob.}0.5 : \text{noise} \backslash \text{salt} \end{array} \right\} \quad (3.72)$$

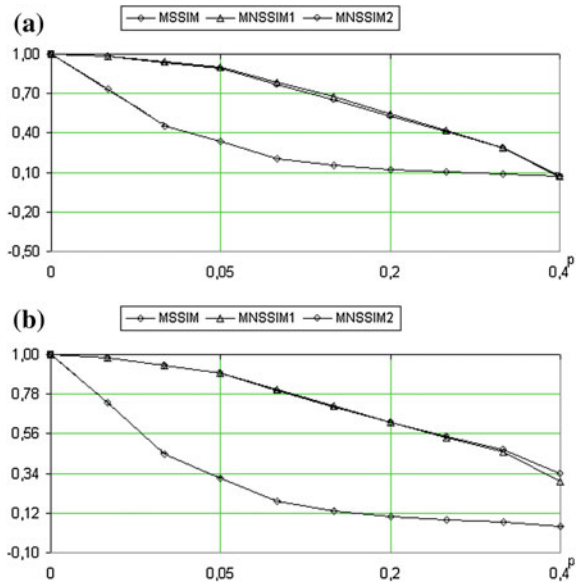
$$y_{ij} = \left\{ \begin{array}{l} f_0 : \text{prob.}p \\ x_{ij} : \text{prob.}1 - p \end{array} \right\} \quad (3.73)$$

Figure 3.12 shows a behavior of criteria depending on probability of pulse interference p . Plots from Fig. 3.12a, b confirm that the MSSIM criterion greatly reduces the image quality at low interference intensity. The MNSSIM1 and the MNSSIM2 matrices demonstrate values more identical to human vision system and close to each other (a human vision filters the low interference intensity). At higher interference intensity, the MSSIM criterion passes to the “saturation” mode.



Fig. 3.11 “Lena” image: **a** “pepper” noise with $p = 0.05$, MSSIM = 0.371, MNSSIM1 = 0.899, MNSSIM2 = 0.902, **b** “salt/pepper” noise with $p = 0.05$, MSSIM = 0.315, MNSSIM1 = 0.899, MNSSIM2 = 0.898

Fig. 3.12 The MSSIM, MNSSIM1 and MNSSIM2 values versus pulse noise intensity and noise type: **a** “salt” type of pulse noise, **b** “salt/pepper” type of pulse noise



The MNSSIM1(2) criteria values within region of high p values almost reduce, when this parameter increases

The MSSIM and the MNSSIM1(2) criteria at quasi-Gaussian noise.

Let us clarify the term of quasi-Gaussian interference in a digital image. Each of color components RGB takes on a value from 0 to $2^8 - 1 = 255$ at integer 8-bit signal level presentation. Therefore, each color RGB component at quasi-Gaussian

interference is formed by Eq. 3.74, where $\eta_{ij} \in N(0, 1)$ is the independent Gaussian values with zero mean and unit variance for each image pixel, $\sigma_0 \leq 1$ is a normalizing factor that control noise variance, $[z]$ is an integral part of number z .

$$y_{ij} = x_{ij} + [\sigma_0 \cdot \eta_{ij} \cdot 2^5] + 2^7 \quad (3.74)$$

Coefficient $2^5 = 32$ and deviation $2^7 = 128$ are selected so that resulting values with probability $p = 0.995$ are within the interval $0 \leq y_{ij} \leq 255$. If a resulting value is beyond this interval, i.e. $y_{ij} < 0$ or $y_{ij} > 255$, then a rounding is performed to $y_{ij} = 0$ or $y_{ij} = 255$, respectively.

The images from database of Laboratory for Image and Video Engineering [26] were used as test images. Subjective quality values of Difference Mean Opinion Score (DMOS) values for images from this database were also obtained. For images noisy with Gaussian noise, the MSSIM, the MNSSIM1, and the MNSSIM2 criteria values as well as the rank Spearman correlation coefficients between these criteria values and DMOS values were calculated.

The rank Spearman correlation coefficients between the MSSIM, the MNSSIM1, and the MNSSIM2 criteria and DMOS values are computed as follows. The calculated values of the MSSIM, the MNSSIM1, and the MNSSIM2 criteria are assigned with ranks, and ranks are also set to respective DMOS values. Then the Spearman correlation coefficient is calculated. A number of elements in sequence n equals 49 for all quality criteria and their respective DMOS values. For each of the MSSIM, the MNSSIM1, and the MNSSIM2 quality criteria, the obtained correlation coefficients were checked. Test results of image “Parrots” are provided by Table 3.7.

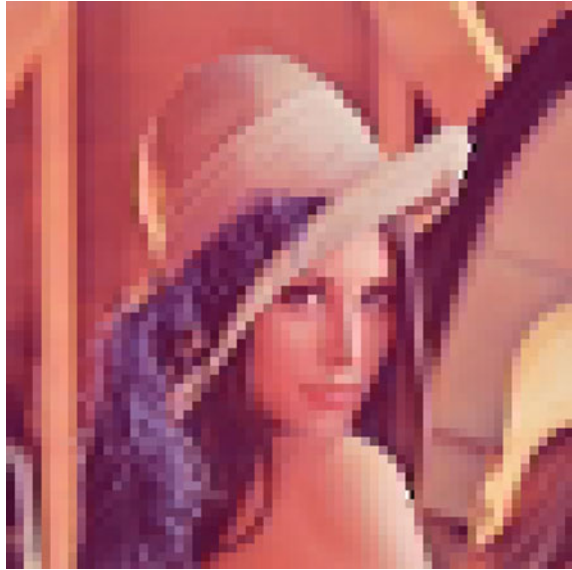
It is evident from Table 3.7, that the MSSIM criterion values and variation range are higher than those of the MNSSIM1 and the MNSSIM2 criteria at quasi-Gaussian noise. This is explained by the fact that the MSSIM criteria structure is more suitable for images with Gaussian statistics. All criteria are strongly correlated with DMOS.

With strong image block distortion or blue, the MSSIM criterion produces an overestimate. At the same time the MNSSIM1 and the MNSSIM2 criteria have low values, thus corresponding to low visual quality of the distorted images. Figure 3.13 presents an image with block distortion and high JPEG algorithm compression.

Table 3.7 Correlation results for image “Parrots”

σ_0	DMOS	MSSIM	MNSSIM1	MNSSIM2
1.000	68.72	0.027	0.012	0.017
0.129	47.03	0.252	0.152	0.208
0.063	38.93	0.518	0.314	0.359
0.031	28.50	0.788	0.520	0.541
Spearman correlation coefficient		-0.955	-0.929	-0.921

Fig. 3.13 “Lena” image recovered after JPEG compression.
 MSSIM = 0.634,
 MNSSIM1 = 0.073,
 MNSSIM2 = 0.079



Possibilities of use of structural similarity criteria to detect changes of actions.

During research, an issue concerning the ability of structural similarity criteria to track the contextual similarity of images was investigated. To solve this task, the MSSIM, the MNSSIM1, and the MNSSIM2 criteria were applied to different images. It was discovered that, when comparing absolutely different images with the same spatial sizes the MSSIM criteria gives unreasonably high values, whereas the MNSSIM1 and the MNSSIM2 criteria values are practically equal to 0. Figure 3.14 shows an example of images compared by the MSSIM, the MNSSIM1, and the MNSSIM2 criteria values. The obtained data proves that the non-parametric modifications are more relevant to the name of structural similarity criterion.

Comparison of pixel and spectral image analysis algorithms.

Multiple tests with real images and video sequences were made by the authors to discover the features and the abilities provided by the above described algorithms. The following intermediate conclusion can be performed:

1. The MSSIM and the MNSSIM1(2) structural similarity criteria are efficient to detect changes in frame and video sequence fragments, when images are processed without compression.
2. The non-parametric the MNSSIM1(2) criteria require more operations as compared with the MSSIM and the spectral algorithm. However, their values are more compatible with human perception.
3. Among all analyzed algorithms, the MNSSIM1(2) has the highest immunity to pulse and other non-Gaussian interference.
4. The MNSSIM1(2) non-parametric criteria could be applied to determine change in a video sequence scene and to detect external frames added to a video sequence.



Fig. 3.14 Comparison of two images with $MSSIM = 0.339$, $MNSSIM1 = -0.004$, $MNSSIM2 = -0.006$

5. The algorithms for spectral structural variation detection obtained by maximum likelihood method are optimal with Gaussian interference.
6. The quasi-optimal detection algorithms applying statistics are similar to the optimal ones by their characteristics. They are not sensitive for threshold selection and image type. The spectral algorithms are more sensitive to change of an image type as compared to the $MSSIM$ and the $MNSSIM1(2)$.
7. The difference in statistics values with or without texture variations are tens/hundreds times in case of spectral algorithms and several times in the case of the $MSSIM$ and the $MNSSIM1(2)$ algorithms.
8. As for computation expenses, the spectral and pixel algorithms are approximately similar. They can operate in the real-time mode.
9. The spectral algorithms are very efficient in real-time operation, especially when they are embedded to video codecs. It has been found that these algorithms can operate in the truncated spectrum width with a few components.
10. A promising spectral basis is the discrete Chebyshev transform (GDCT). The GDCT spectrum is the most fast decreasing for orthogonal transforms.
11. A combining of the Chebyshev sampling and a spectral detection allows to reduce information processing in 4–8 times depending on a frame variation nature.
12. To solve a certain task, the proposed algorithms should be selected by their resource consumption, computation automation degree, and image distortions.

3.6 Conclusion

The chapter covers the use of the MSSIM, the MNSSIM1, and the MNSSIM2 criteria and the spectral algorithms D_0 and D_E to detect changes in a video sequence or to compare textures of various images. The developed MNSSIM1(2) criteria and the spectral algorithms D_0 and D_E can be used in artificial vision systems, in many other scopes connecting with variations of texture, spectral and correlation parameters of recorded signals. To meet a certain challenge, the selection of the proposed algorithm should be determined by the required resource intensity, an automation degree, and availability of image distortions. The proposed algorithms are used to design novel video codecs and intelligent video record systems. To conclude one can emphasize that evolution of basic PSNR, the MSSIM criteria and the spectral criteria, etc. is proceeding in [27–31].

References

1. Pratt WK (2001) Digital image processing. PIKS Inside, 3rd edn. Wiley, New York
2. Wang Z, Bovik AC (2009) Mean squared error: love it or leave it. *IEEE Signal Process Mag* 26(1):98–117
3. Wang Z, Bovik AC (2002) A universal image quality index. *IEEE Signal Process Lett* 9 (3):81–84
4. Wang Z, Bovik AC, Sheikh H, Simoncelli E (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
5. Bulygin AV, Radchenko TA, Radchenko YuS (2008) Comparative analysis of structural likelihood index modification. *Digit Signal Proc* 4:11–14 (in Russian)
6. Valvar HS, Hallapuro A, Karczewicz M (2003) Low-complexity transform and quantization in H.264/AVC. *IEEE Trans Circ Syst Video Technol* 13(7):598–603
7. Chui CK (1997) Wavelets: a mathematical tool for signal analysis. SIAM, Philadelphia, PA
8. Stark HG (2005) Wavelets and signal processing. Springer, Berlin, Heidelberg
9. Radchenko YuS (2002) Image reconstruction and compression method based on fast Chebyshev transform. *Optoelectron Instrum Data Process* 4:27–33
10. Radchenko YuS (2002) Research of signal recovery, Suppression and processing algorithms based on polynomial transformations. In: World multi-conference of systemics, cybernetics and informatics, VIX:262–266
11. Radchenko YuS (2002) The algorithm of image compression based on polynomial transformations. *Digit Signal Process* 1:2–6 (in Russian)
12. Radchenko YuS (2010) Method for discrete signal transformation based on orthogonal polynomials. RU Patent No 2405248 27 Nov 2010 (in Russian)
13. Keith J (2004) Video demystified, 4th edn. Newnes, Elsevier
14. Sheikh HR, Sabir MF, Bovik AC (2006) A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans Image Process* 15(11):3440–3451
15. Kobzar AI (2006) Applied mathematics statistics. For engineers and scientists. Physmatit, Moscow (in Russian)
16. Radchenko YuS, Ryazhskikh MV (2013) Methods for detecting structural changes in frames of video sequences when recording physical and chemical experiments. *Instrum Exp Tech* 56 (1):41–45

17. Bulygin AV, Radchenko TA, Radchenko YuS (2009) Research of spectral algorithm of video sequence modification detection. *Radioelectronics Commun Syst* 52(7):371–378
18. Ptacheck M (1990) *Digital television. Theory and engineering*. Radio I Svyaz, Moscow (in Russian)
19. Stark H (1987) *Image recovery: theory and application*. Academic Press, Orlando
20. Abramowitz M, Stegun I (1964) *Handbook of mathematical functions with formulas, Graphs and mathematical tables*. National Bureau of Standarts, Washington
21. Bateman H, Erdelyi A (1953) *Higher transcendental functions*. McGraw-Hill, New York
22. Miyahara M, Kotani K (1985) Block distortion on orthogonal transform coding—analysis, minimization and distortion measure. *T-COM* 33(1):90–96
23. Godsill SJ, Kokaram AC (1996) Joint interpolation, motion, and parameter estimation for degraded image sequence with missing data. *IEEE Trans Signal Process* 1(8):1–4
24. Proakis JG (1995) *Digital communication*. McGraw-Hill Higher Education, New York
25. Radchenko YuS, Sokhnyshev SV, Bulygin AV (2010) Concept of composite video codec MGDCT construction. In: 12th international conference digital signal processing and its application XII-2:151–155
26. LIVE Video Quality Database. http://live.ece.utexas.edu/research/quality/live_video.html Accessed 12 June 2014
27. Channappayya SS, Bovik AC, Heath RW (2008) Rate bounds on SSIM index of quantized images. *IEEE Trans Image Process* 17(9):1624–1639
28. Changhoon Y, Bovik AC (2011) Quality assessment of deblocked images. *IEEE Trans Image Process* 15(2):430–444
29. Sheikh HR, Bovik FC (2011) Image information and visual quality. *IEEE Trans Image Process* 20(1):88–98
30. Chen GH, Yang CL, Po LM, Xie SL (2006) Edge-based structural similarity for image quality assessment. *Int Conf Acoust Speech Signal Process* 2:933–936
31. Laboratory for Image & Video Engineering Publications. <http://live.ece.utexas.edu/publications/>. Accessed 12 June 2014

Chapter 4

Hierarchical Adaptive KL-Based Transform: Algorithms and Applications

Roumen Kountchev and Roumiana Kountcheva

Abstract In this research, a novel approach for processing of single images and image sequences by using the Hierarchical Adaptive Karhunen-Loeve (KL)-based Transform (HA-KLT) is presented. This approach is also suitable for block image coding, and for intra-frame processing of groups of correlated images. The main objective of the new transform is to achieve a high decorrelation of the image blocks (resp., images in the processed groups) as a result of the multi-level processing through adaptive KL transform of the sub-blocks (resp., sub-groups), on which the corresponding blocks (groups) are divided. After the execution of each hierarchical transform level, the sub-blocks (sub-groups) are rearranged in correspondence to their mutual correlation, starting from these, with maximum correlation. The kernel of the multi-level transform is the Adaptive KL Transform (AKLT), whose transform matrix could be a size 2×2 or 3×3 . In the work, the algebraic method is presented, which calculates the elements of the AKLT matrix of size 2×2 and 3×3 . Here the algorithms for 2D HA-KLT and 3D HA-KLT are also given in detail for the blocks of a single image and for the inter-frame processing of groups of images. The evaluations of the computational complexity of these algorithms are obtained, compared to that of the famous KLT. The basic advantages of the new approach for image processing through HA-KLT are a lower computational complexity and the parallel recursive calculations with small number of hierarchical levels, which opens the ability for real-time coding of video sequences. The HA-KLT algorithms are a basis for the creation of new efficient methods for compression of groups of correlated images and single images without visual loss, compression with motion compensation for TV and multi-view images, compression of medical and multispectral images, image fusion, face recognition in the reduced features' space, object tracking in video sequences, among others.

R. Kountchev (✉)
Department of Radio Communications and Video Technologies,
Technical University of Sofia, 8 Bld. Kliment Ohridski, 1000 Sofia, Bulgaria
e-mail: rkountch@tu-sofia.bg

R. Kountcheva
T&K Engineering, Mladost 3, Pob 12, 1712 Sofia, Bulgaria

Keywords Images decorrelation · Karhunen-Loeve transform · Adaptive Karhunen-Loeve transform · Hierarchical adaptive Karhunen-Loeve transform · Image compression

4.1 Introduction

The Karhunen-Loeve Transform (KLT), known also as Hotelling transform or Principal Component Analysis (PCA), is an object of large number of investigations presented in many scientific monographs [1–8] and papers [9–18]. This transform is related to the class of linear statistical orthogonal transforms for groups of vectors, obtained, for example, from the pixels of one image or from a group of matrix images. The KLT has significant role in the analysis and processing of digital images, and also in the systems for computer science and pattern recognition. It has a wide variety of application areas: for the creation of optimal models in the image color space [9], the compression of signals and groups of correlated images [2–7, 10, 11], the creation of the recognized objects descriptors in the reduced features' space [14–16], image fusion [17] and segmentation [18], image steganography [19], etc.

The KLT has some significant properties:

- It is an optimal orthogonal transform for a group of vectors, because as a result of the transform, the maximum part of their energy is concentrated in a minimum number of their components.
- After reduction of the low-energy components of the transformed vectors, the corresponding restored vectors have minimum Mean Square Error (MSE).
- The components of the transformed vectors are not correlated. In particular, in case that the probability distribution of the vectors is Gaussian, after the KLT their components become decorrelated and independent.

The Independent Components Analysis (ICA) [20] is very close to the KLT in respect of their calculation complexity and properties.

Analysis of the image transform methods based on the KLT is presented in Sect. 4.2. The Hierarchical Adaptive Karhunen-Loeve based Transform (HA-KLT) for a group of images is developed in Sect. 4.3. Section 4.4 provides the analytical definition of the adaptive KLT matrix of size 2×2 and 3×3 . The HA-KLT applications for sequences of motion-compensated TV frames are discussed in Sect. 4.5. Section 4.6 includes the evaluations of the computational complexity of the two-level HA-KLT 3×3 for a group of images, and experimental results for group of consecutive images are located in Sect. 4.7. Section 4.8 involves the principle of the HA-KLT for single image. Section 4.9 provides the algorithm for recursive 2D HA-KLT, and evaluation of the computational complexity of 2D HA-KLT is drawn in Sect. 4.10. The experimental results for single images are situated in Sect. 4.11. Conclusion is located in Sect. 4.12.

4.2 Analysis of the Image Transform Methods Based on the KLT

One of the basic problems, which limit the use of the KLT, is due to its high computational complexity, which grows up together with the number of the vectors' components. Various approaches are offered to overcome this problem. One of them is based on the KLT calculation through iterative methods, which do not require the definition of the characteristic polynomial of the vectors' covariance matrix (the first approach). In this case, the KLT is executed in two consecutive stages. In the first stage, the original image matrix is transformed into a three-diagonal form through QR decomposition [21, 22], and after that—into a bi-diagonal, by using the Householder's transforms [23]. In the second stage, on the bi-diagonal matrix are applied iterative methods, for which the iterations are stopped, after the needed accuracy is achieved. The iterative KLT calculation through the methods of Jacobi and Givens [24, 25] is based on the execution of a sequence of orthogonal transforms with rotational matrixes of size 2×2 .

One well-known approach is based on the KLT calculation by using neural networks [26] as a type of generalized hebbian or multilayer perceptron networks (the second approach). They both use iterative learning algorithms, for which the number of needed operations can reach several hundreds. The third approach is based on an algorithm, known as sequential Karhunen-Loeve Transform/Singular Value Decomposition (KLT/SVD) [27]. The basic idea is as follows: the image matrix is divided into blocks of small size, and on each the KLT based on the QR decomposition is applied. In the start of the processing, the KLT is calculated for the first block (for example, the upper left), and after that the iterative KLT calculations for each of the next blocks, using the transform matrix and already calculated for the preceding block, are used. In the iterative process, the KLT components with very small eigen values are neglected. The fourth approach based on the recursive calculation of the covariance matrix of the vectors, its eigen values, and eigen vectors is presented in [28, 29]. In the paper [30], a hierarchical recursive block processing of matrices is introduced.

The next fifth approach is based on the so-called Distributed KLT [31, 32], where each vector is divided into sub-vectors and on each a partial KLT is applied. Then the global iterative approximation of the KLT through conditional KLT, based on side information, is executed. This approach was further developed in [33], where one algorithm for adaptive two-stage KLT, combined with JPEG2000, and aimed at the compression of hyper-spectral (HS) images is offered. A similar algorithm for enhanced search is the Integer Sub-optimal KLT (Int SKLT) [34], which uses the lifting factorization of matrices. This algorithm is basic for the idea of KLT, executed through a multilevel strategy, also called "divide-and-conquer" (D&C) [35, 36]. In correspondence with this idea, the KLT for a long sequence of images is executed after dividing it into smaller groups, for which the corresponding KLT have lower computational complexity. By applying the KLT on

each group, a local decorrelation is obtained only. For this reason, the eigen images for the first half of each group in the first decomposition level are used as an input for the next (second) level of the multi-level transform, etc. In the case, when the KLT group contains two components only, the corresponding multilevel transform is called the Pair-wise Orthogonal Transform (POT) [37].

The experimental results obtained for this transform, when used for HS images, show that it is more efficient than the Wavelet Transform (WT) in respect of rate-distortion performance, computational cost, component scalability, and memory requirements. Another approach is based on the Iterative Thresholding Sparse PCA (ITSPCA) [38] algorithm, aimed at the reduction of the features' space dimension with minimum dispersion loss. A fast calculation algorithm (Fast KLT) is known for the particular case, when the images are represented through the first order Markov model [39]. In correspondence with the algorithm for the PCA randomization [40], a certain number of rows (or columns) of the covariance matrix are selected on the basis of an accidental choice, and on the basis of this approximation, the computational complexity of the KLT is reduced. In the works [41–43], hybrid methods for compression of multi-component images through the KLT combined with SVD, wavelets, adaptive mixture of principal components model, and JPEG2000 are presented.

The analysis of the famous KLT methods shows that:

- In case of iterative calculations, the number of iterations depends on the covariance matrix of the vectors. In many cases, this number is very high, which makes the real-time KLT calculation extremely difficult.
- In case that the method for multilevel D&C is used, the eigen images from the second half of each group are not transformed in the next levels and as a result, they are not completely decorrelated. Something more—the selection of the length of each group of images is not optimized.

In this research, a novel approach for decorrelation of groups (sequences) of correlated images by using a transform based on the HA-KLT [44, 45] is offered. This approach permits to reduce the computational complexity, and to use a parallel processing for all sub-groups of images obtained after dividing the initial sequence into smaller parts in correspondence with their mutual correlation. The same approach is generalized for the block coding of single images [46].

The image processing through the KLT could be executed in various directions: it could be applied on a group of images or on a single image only. In the first case, the KLT is used for inter-frame (3D) processing of the group, and in the second case,—for intra-frame image processing, which could be also 1D or 2D.

The Karhunen-Loeve Transform for inter-frame (3D) processing of a group of correlated images is investigated in Sect. 4.2.1, and the Karhunen-Loeve transform for intra-frame (2D) processing of single images is presented in Sect. 4.2.2.

4.2.1 Karhunen-Loeve Transform for Inter-frame (3D) Processing of a Group of Correlated Images

The way used to define the vectors $\vec{X}_s = [x_{1s}, x_{2s}, \dots, x_{Rs}]^T$, $s = 1, 2, \dots, S$ ($S = N^2$) for a group of R correlated images $[X_r]$ (for $r = 1, 2, \dots, R$), each of size $N \times N$ is shown in Fig. 4.1. In this case, the components of the vector \vec{X}_s comprise the pixels with same spatial position in each of the images from the processed group $[X_r]$. Through applying the KLT on these vectors, the transformed vectors $\vec{Y}_s = [y_{1s}, y_{2s}, \dots, y_{Rs}]^T$, which represent the group of the decorrelated images, are obtained. They are represented by the group of matrices $[EI_r]$ of size $N \times N$ called “eigen” (or “principal”) images. To reduce the information surplus in the group $[EI_r]$, the first k “eigen” images only are retained, which contain the basic part of the energy of the whole group, and the remaining $(R - k)$ are cut-off (set to be equal to zero).

The direct/inverse KLT is applied on the vectors \vec{X}_s , resp. \vec{Y}_s , is defined by Eq. 4.1, where $[\Phi]^T = [\vec{\Phi}_1, \vec{\Phi}_2, \dots, \vec{\Phi}_R]$ is the transposed KLT matrix of size $R \times R$.

$$\vec{Y}_s = [\Phi](\vec{X}_s - \vec{\mu}) \quad \vec{X}_s = [\Phi]^T \vec{Y}_s + \vec{\mu} \quad (4.1)$$

The transposed KLT matrix contains the eigen vectors $\vec{\Phi}_r = [\Phi_{1r}, \Phi_{2r}, \dots, \Phi_{Rr}]^T$ for $r = 1, 2, \dots, R$ of the covariance matrix $[K_X]$ of the vectors \vec{X}_s . The last matrix is of size $R \times R$ and is defined by Eq. 4.2, where $\vec{\mu} = E(\vec{X}_s) = (1/S) \sum_{s=1}^S \vec{X}_s = [\mu_1, \mu_2, \dots, \mu_R]^T$ is the mean vector of length R , $E(\cdot)$ is the meaning operator.

$$[K_X] = (1/S) \sum_{s=1}^S \vec{X}_s \vec{X}_s^T - \vec{\mu} \vec{\mu}^T \quad (4.2)$$

Each of the orthonormalized eigen vectors $\vec{\Phi}_r$ of the covariance matrix $[K_X]$ is the solution of the linear system of $(R + 1)$ Eq. 4.3, where λ_r are the eigen vectors of $[K_X]$.

$$[K_X] \vec{\Phi}_r = \lambda_r \vec{\Phi}_r \quad \|\vec{\Phi}_r\|^2 = \Phi_{1r}^2 + \Phi_{2r}^2 + \dots + \Phi_{Rr}^2 = 1 \quad (4.3)$$

Each eigen value λ_r of the matrix $[K_X]$ is the root of its' characteristic equation (a polynomial of degree R) provided by Eq. 4.4, where $[I]$ is a singular matrix of size $R \times R$.

$$\det|[K_X] - \lambda_r [I]| = 0 \quad (4.4)$$

Since $[K_X]$ is a symmetrical matrix towards the main diagonal, its eigen values λ_r are real numbers [47], whose sequential numbers are defined by Eq. 4.5.

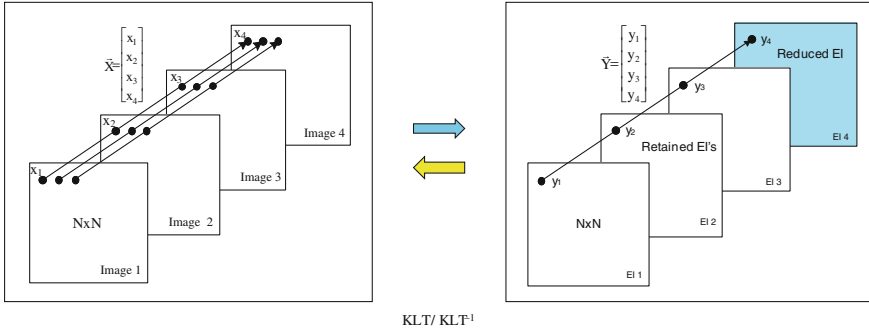


Fig. 4.1 Direct and inverse KLT for a group of $R = 4$ images, each of size $N \times N$ pixels. The number of the retained eigen images is $k = 3$, and the number of the 4-dimensional vectors for each group of images is $S = N^2$

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_R \tag{4.5}$$

The algorithm for inter-frame (3D) processing of a group of R correlated images through KLT comprises the following basic steps:

- Calculation of the covariance matrix $[K_X]$ in correspondence with Eq. 4.2 on the basis of the vectors \vec{X}_s .
- Calculation of all roots λ_r of the characteristic Eq. 4.4, numbered in accordance with Eq. 4.5.
- Solving the system of Eq. 4.3 for each eigen value λ_r and definition of the components of the corresponding eigen vector $\vec{\Phi}_r = [\Phi_{1r}, \Phi_{2r}, \dots, \Phi_{Rr}]^T$ for $r = 1, 2, \dots, R$. When all vectors $\vec{\Phi}_r$ are calculated, is built the KLT matrix $[\Phi]^T = [\vec{\Phi}_1, \vec{\Phi}_2, \dots, \vec{\Phi}_R]$ of size $R \times R$.
- The direct KLT is applied on each vector $\vec{X}_s = [x_{1s}, x_{2s}, \dots, x_{Rs}]^T$ for $s = 1, 2, \dots, S$, and, as a result, the group of eigen images $[EI_r]$ for $r = 1, 2, \dots, R$ in accordance with Eq. 4.1 is obtained.
- The first $k < R$ eigen images $[EI_r]$ for $r = 1, 2, \dots, k$ are retained and is executed the inverse KLT on the vectors \vec{Y}_s in correspondence with Eq. 4.1. On the basis of the so obtained vectors $\hat{\vec{X}}_s$ is restored the whole group of images $[\hat{X}_r]$ for $r = 1, 2, \dots, R$.

Let us assume that the components of the vectors \vec{X}_s and \vec{Y}_s have same dimension. Then the Compression Ratio (CR) for the group of R images of size $N \times N$ obtained in result of the use of KLT, is defined by Eq. 4.6, where $R(1 + R)$ is the global number of the components $[\Phi]$ of the matrix and of the vector $\vec{\mu}$, needed for the execution of the inverse KLT on the vectors \vec{Y}_s .

$$CR(k) = \frac{N^2 R}{N^2 k + R(1 + R)} = \frac{R}{k + (R/N^2)(1 + R)} \quad (4.6)$$

Since practically $R \ll N^2$ from Eq. 4.6 it follows.

$$CR(k) \approx R/k \quad \text{for } k = 1, 2, \dots, R - 1 \quad (4.7)$$

For small values of k , the compression ratio is high, but then also grows up the MSE of the approximation $\overline{\varepsilon^2}$, which for the KLT is minimum in respect of other famous linear transforms [11]. This error is defined in accordance with Eq. 4.8, where $\sigma_r^2(y)$ represents the variance of the components y_{rs} of the transformed vectors \vec{Y}_s .

$$\overline{\varepsilon^2}(k) = \sum_{r=k+1}^R \lambda_r = \sum_{r=k+1}^R \sigma_r^2(y) \quad (4.8)$$

For example, if $k = R/2$ from Eqs. 4.7 to 4.8, then Eq. 4.9 is obtained.

$$CR\left(\frac{R}{2}\right) \approx 2 \quad \overline{\varepsilon^2}\left(\frac{R}{2}\right) = \sum_{r=(R/2)+1}^R \lambda_r \quad (4.9)$$

If $k = R - 1$, then $CR \approx 1$ and $\overline{\varepsilon^2} \approx 0$. In this case, there is no compression and no information loss. Then for the group of eigen images $[EI_r]$ for $r = 1, 2, \dots, R$, could be used only algorithms for the lossless compression. The main problem of the described algorithm for 3D KLT is the high computational cost, which grows up together with the number of images R in the group.

4.2.2 Karhunen-Loeve Transform for Intra-frame (2D) Processing of Single Images

For every single image, represented by the matrix $[X]$ of size $N \times N$, could be applied 2D KLT on blocks of size $R \times R$ for $R \ll N$, and usually N/R is set to be integer. The implementation of the 2D-KLT is simplified, executing it as two 1D KLTs. For this, the 2D KLT is calculated in two consecutive steps. In the first step, the 1D KLT is applied on the horizontally oriented vectors $\vec{X}_s = [x_{1s}, x_{2s}, \dots, x_{Rs}]^T$, for $s = 1, 2, \dots, S = N^2/R$, which are contained in each image block $[X]$ in correspondence with Fig. 4.2. From the transformed vectors $\vec{Y}_s = [y_{1s}, y_{2s}, \dots, y_{Rs}]^T$, the matrix $[Y]$ of size $N \times N$ is built. In the second step, the 1D KLT is applied on the vertically oriented vectors of the blocks of the matrix $[Y]$.

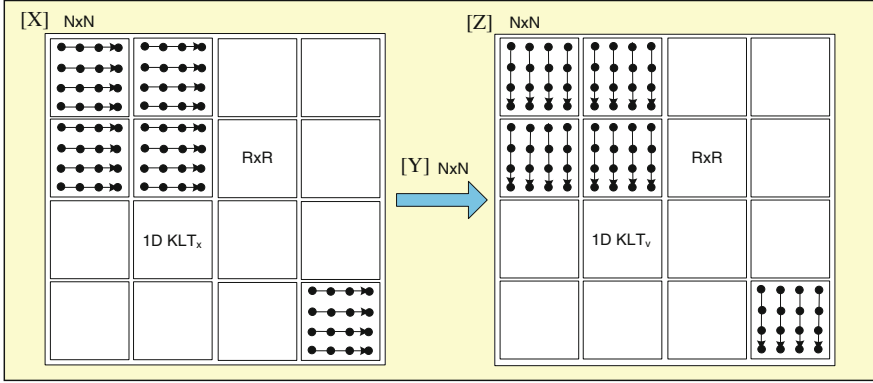


Fig. 4.2 Two-step transform through the direct 1D KLT of the vectors for each block of size $R \times R$: first step is in a horizontal direction, second step is in a vertical direction

The calculation of the 1D KLT in each step is executed in correspondence with Eqs. 4.1–4.5 for $S = N^2/R$. In the first step, the direct 1D KLT is applied in horizontal direction (1D KLT_x), Eq. 4.10, where $\vec{\mu}_x = E(\vec{X}_s)$ is an R -dimensional mean vector, $[\Phi_x]$ is a matrix of size $R \times R$, which contains the eigen vectors $\vec{\Phi}_{r,x}$ of the covariance matrix $[K_X]$. This matrix is defined by Eq. 4.10.

$$\vec{Y}_s = [\Phi_x](\vec{X}_s - \vec{\mu}_x) \quad (4.10)$$

The vectors $\vec{Y}_s = [y_{1s}, y_{2s}, \dots, y_{Rs}]^T$ build the matrix $[Y]$ of size $N \times N$, which is divided again into blocks of size $R \times R$. In the second step on the vertically oriented vectors of each block is applied the direct 1D KLT in vertical direction (1D KLT_v), Eq. 4.11, where $\vec{\mu}_v = E(\vec{Y}_s)$ is an R -dimensional mean vector, $[\Phi_v]$ is a matrix of size $R \times R$ built from the eigen vectors $\vec{\Phi}_{r,v}$ of the covariance matrix $[K_Y]$.

$$\vec{Z}_s = [\Phi_v](\vec{Y}_s - \vec{\mu}_v) \quad (4.11)$$

The covariance matrix $[K_Y]$ is defined by Eq. 4.12.

$$[K_Y] = (1/S) \sum_{s=1}^S \vec{Y}_s \vec{Y}_s^T - \vec{\mu}_v \vec{\mu}_v^T \quad \text{for } S = N^2/R \quad (4.12)$$

The vectors $\vec{Z}_s = [z_{1s}, z_{2s}, \dots, z_{Rs}]^T$ build the matrix $[Z]$ of size $N \times N$, which contains blocks of size $R \times R$. The energy of the elements of each block is concentrated in a small number only, placed in the upper left corner of the block.

When $k < R^2$ high-energy coefficients from each block are retained, the image $[\hat{X}]$ could be restored after inverse 2D KLT based on the inverse 1D KLT (executed in two steps in vertical and horizontal directions) with minimum MSE. The compression ratio for the image $[X]$ after using the block 2D KLT is defined by

Eq. 4.13, where $2R(1 + R)$ is the global number of the components of the matrices $[\Phi_x]$ and $[\Phi_y]$ of vectors $\vec{\mu}_x, \vec{\mu}_y$, needed for the execution of the two-step inverse 1D KLT in horizontal and vertical directions.

$$CR = \frac{N^2}{kN^2/R^2 + 2R(R+1)} = \frac{R^2}{k + [2R^3(R+1)]/N^2} \quad (4.13)$$

Practically $R^2 \ll N^2$, and as a result from Eq. 4.13 it follows.

$$CR(k) \approx R^2/k \quad \text{for } k = 1, 2, \dots, R^2 \quad (4.14)$$

The basic difficulties in the use of the 2D KLT for image processing are mentioned below:

- The higher computational complexity of the KLT compared to that of the determined orthogonal transforms of the kind Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Hadamard Transform, etc.
- In general, there is no “fast” KLT algorithm (there is such only for the class of images, which can be represented as a first order Markov process [39]).

4.3 Hierarchical Adaptive Karhunen-Loeve Based Transform for a Group of Images

In this research, a novel method for decorrelation of groups (sequences) of correlated images [44, 45] is proposed, which reduces the computational complexity of the transform and permits a parallel processing of the sub-groups of images obtained after dividing the original sequence into smaller parts (Sect. 4.3.1). The novel approach called as the HA-KLT and based on sub-groups of three images or two images is investigated in Sects. 4.3.2 and 4.3.3, respectively.

4.3.1 Determination of the Length of the Group of Images, Processed with the Hierarchical Adaptive KLT

It is assumed that a sequence of P images will be processed. For this, the sequence is divided into groups, which contain $R < P$ correlated images each. Prior to execution of the HA-KLT for each group, it is necessary to select the optimal group length R , which is defined on the basis of the cross-correlation between the images $[X_1]$ and $[X_u]$ in the sequence P , all of them of same size ($N \times N$).

- The normalized cross-correlation coefficient $\rho_{1,u}$, where $(0 \leq \rho_{1,u} \leq 1)$ for the couple of images $[X_1]$ and $[X_u]$ ($u = 1, 2, \dots, P$), is defined by Eq. 4.15 [47], where $S = N \times N$ is the number of pixels in the images $[X_u]$, $\bar{x}_1 = E(x_s^1)$, $\bar{x}_u = E(x_s^u)$, x_s^1 , x_s^u are the values of the pixels with same spatial position in the images $[X_1]$ and $[X_u]$.

$$\rho_{1,u} = \frac{\sum_{s=1}^S (x_s^1 - \bar{x}_1) \times (x_s^u - \bar{x}_u)}{\sqrt{\sum_{s=1}^S (x_s^1 - \bar{x}_1)^2} \times \sqrt{\sum_{s=1}^S (x_s^u - \bar{x}_u)^2}} \quad \text{for } u = 1, 2, \dots, P \quad (4.15)$$

- The number of images R in the group is defined by Eq. 4.16, where δ is a preset threshold with a small value $(0 \leq \delta \ll 1)$, experimentally defined. In case, that the so obtained value for R does not satisfy the condition $R/2$ or $R/3$ to be integer number, a new value is calculated for the number of images in the “extended” group $R_e = R + m_{int}$, where $R_e/2$ or $R_e/3$ is integer. However, the number of added images m_{int} should be minimal.

$$|\rho_{1,u}(u = R)| \leq \delta \quad \text{for } u = 1, 2, \dots, P \quad (4.16)$$

An example for $m_{int} = 1$ (group extension with one interpolated image “*int*”, colored in green) is shown in Fig. 4.3. The “extended” images could be defined in various ways, for example, by using zero interpolation (in this case the last R th image in the group substitutes each interpolated image) or through linear interpolation (the pixels of each interpolated image are calculated through weighted mean of the corresponding pixels with same spatial position in the images R and $(R + 1)$). For $m_{int} = 0$, $R_e = R$, and the group is not extended. One example for the definition of R_e for $R = 4, 5, \dots, 16$ and changing number of images in the sub-groups (two or three) in each hierarchical level of the HA-KLT is given in Table 4.1.

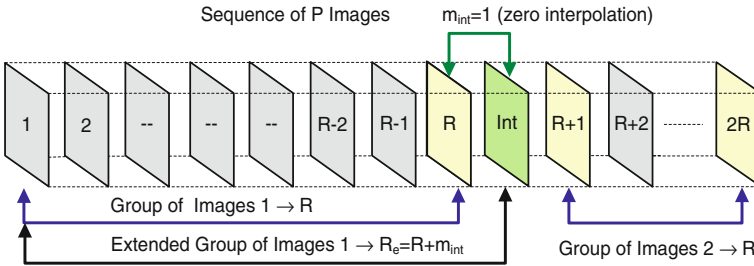


Fig. 4.3 One example for setting the number of images R_e in the extended group on the basis of the cross-correlation coefficient value

Table 4.1 Setting the number of images in the group, the hierarchical levels, and the number of interpolated images for sub-groups of two or three images

R	Number of hierarchical levels, n	Number of sub-groups with two images in one level	Number of sub-groups with three images in one level	m_{int}	R_e
4	2	2	–	0	4
5	3	3	–	1	6
6	3	3	–	0	6
7	3	4	–	1	8
8	3	4	–	0	8
9	2	–	3	0	9
10	3	–	4	2	12
11	3	–	4	1	12
12	3	–	4	0	12
13	3	–	5	2	15
14	3	–	5	1	15
15	3	–	5	0	15
16	4	8	–	0	16

4.3.2 Hierarchical Adaptive KLT Based on Sub-groups of Three Images

Let on the basis of the already done correlation analysis, the sequence P is divided on Groups Of Pictures (GOP) with length $R = 9$. Then, each GOP is divided into 3 sub-groups, each containing three images. In this case the HA-KLT algorithm with $n = 2$ hierarchical levels is shown in Fig. 4.4.

In the first HA-KLT hierarchical level each group of three images is processed by the AKLT with a transform matrix of size 3×3 . As a result, three eigen images from one sub-group (colored in yellow, blue, and green, respectively) are obtained. All nine eigen images are then rearranged so that in the first group to be placed the images, colored in yellow color only, which are on first place in their initial sub-group; in the next new sub-group—only the eigen images, colored in blue, which were on second place in their initial group, etc.

In the next (second) hierarchical HA-KLT level, the AKLT with a new transform matrix of size 3×3 calculated on the basis of the eigen images in the new sub-groups is applied on each three eigen images from all new sub-groups of the processed GOP. In result, three new eigen images for each sub-group (colored in yellow, blue, and green again) are obtained. After that a new rearrangement of the so obtained nine eigen images in the way, already explained for the first HA-KLT level, follows. The new group of 9 images is decorrelated, and the main part of its energy is concentrated in the first image, a small part is in the second image, and the

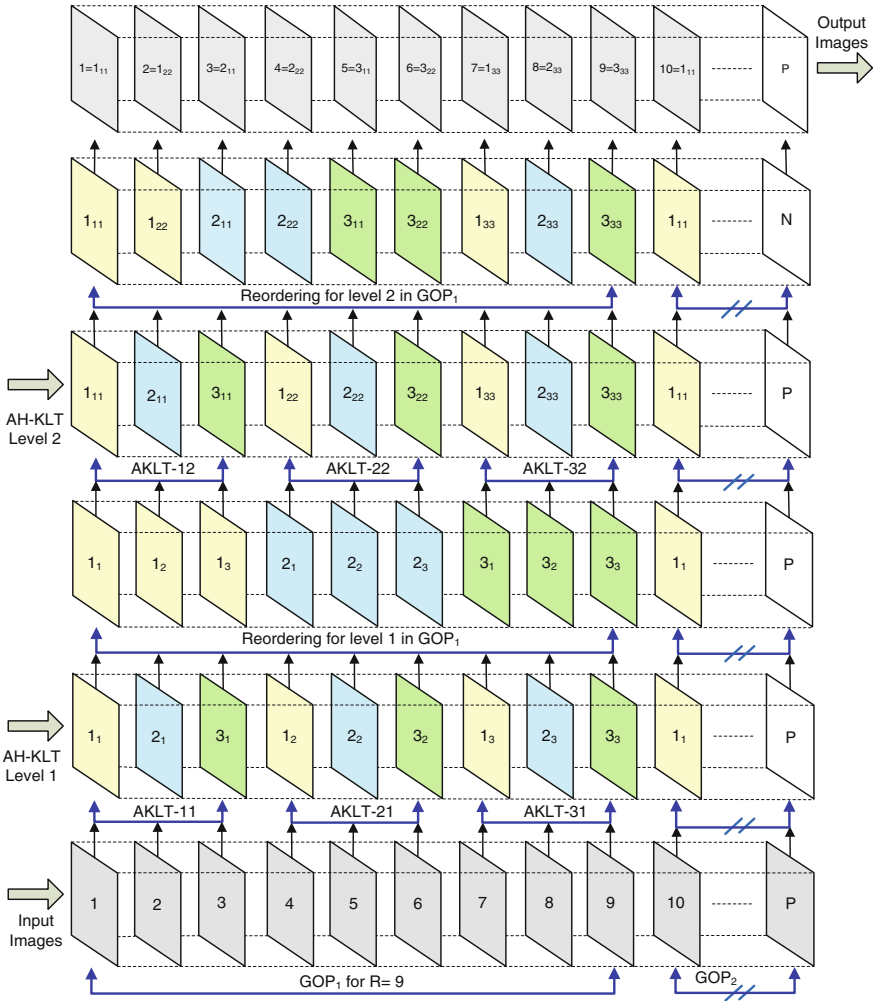


Fig. 4.4 Processing of a group of $R = 9$ images through two-level hierarchical adaptive KLT 3×3

remaining 7 images contain a negligible part of the global group energy only. In this way, a possibility for efficient compression of the processed GOP and its successful restoration after inverse transform is created, because the HA-KLT is a reversible. For the restoration of the processed GOP, it is needed additional information about the transform matrix of each three images in all hierarchical levels. For the HA-KLT processing of a GOP of $R = 9$ images, the global number of transform matrices of size 3×3 is 6, and they contain 54 elements in total. In case, that the elements of each transform matrix are represented through 3 Euler rotation angles, their number is reduced down to 18.

4.3.3 Hierarchical Adaptive KLT Based on Sub-groups of Two Images

In this case, let us assumed that a sequence P divided on the basis of the correlation analysis into GOPs of length $R = 8$ is processed. Then each GOP is divided into sub-groups of two images each. In this case, the HA-KLT algorithm with $n = 3$ hierarchical levels is shown in Fig. 4.5. In the first hierarchical level of the HA-KLT, the AKLT with a transform matrix of size 2×2 is applied on each sub-group of two images [48]. As a result, two eigen images for each sub-group colored in yellow and blue, respectively, are obtained. After the rearrangement of all eight eigen images, only the first images from the first two initial sub-groups (yellow) are placed in the first new sub-group, and in the next new sub-group—only the second images from the first sub-groups (blue), etc. In the next hierarchical levels of the HA-KLT, the processing of the couples of eigen images from each sub-group of the GOP is executed in similar way through the AKLT with a matrix of size 2×2 , calculated on the basis of the two eigen images in the corresponding new sub-group. The group of 8 images obtained in the third hierarchical level is decorrelated, and the main part of its energy is concentrated in the first image; much smaller part is in the second, and the remaining 6 images contain a negligible part of the group energy only. In this way, an opportunity to achieve efficient compression of correlated GOPs and full restoration through inverse HA-KLT is created. For the GOP restoration, additional information about the transform matrix for each couple of images in all hierarchical levels is needed: in total these 12 matrices of size 2×2 for a GOP of $R = 8$ images. In this case, the global number of the elements in the transform matrices is 48, and after their representation through corresponding rotational angles, it is reduced to 12. The comparison with the algorithm, shown in Fig. 4.4, proves that the required additional information is smaller, but the number of hierarchical levels is increased by one.

4.4 Analytical Definition of the AKLT Matrix of Size 2×2 and 3×3

For the KLT matrices of small size (2×2 and 3×3), the exact analytical solution for the calculation of their elements exists. The obtained results for the calculation of the elements of the transform AKLT matrices of size 3×3 and 2×2 used for the processing of the sub-groups of images in accordance with the algorithms and shown in Figs. 4.4 and 4.5.

The algebraic method for calculation of the AKLT matrix of size 3×3 is developed in Sect. 4.4.1. The evaluation of the decorrelation of the images transformed through HA-KLT 3×3 is represented in Sect. 4.4.2. Section 4.4.3 provides the algebraic method for determination of the AKLT matrix of size 2×2 .

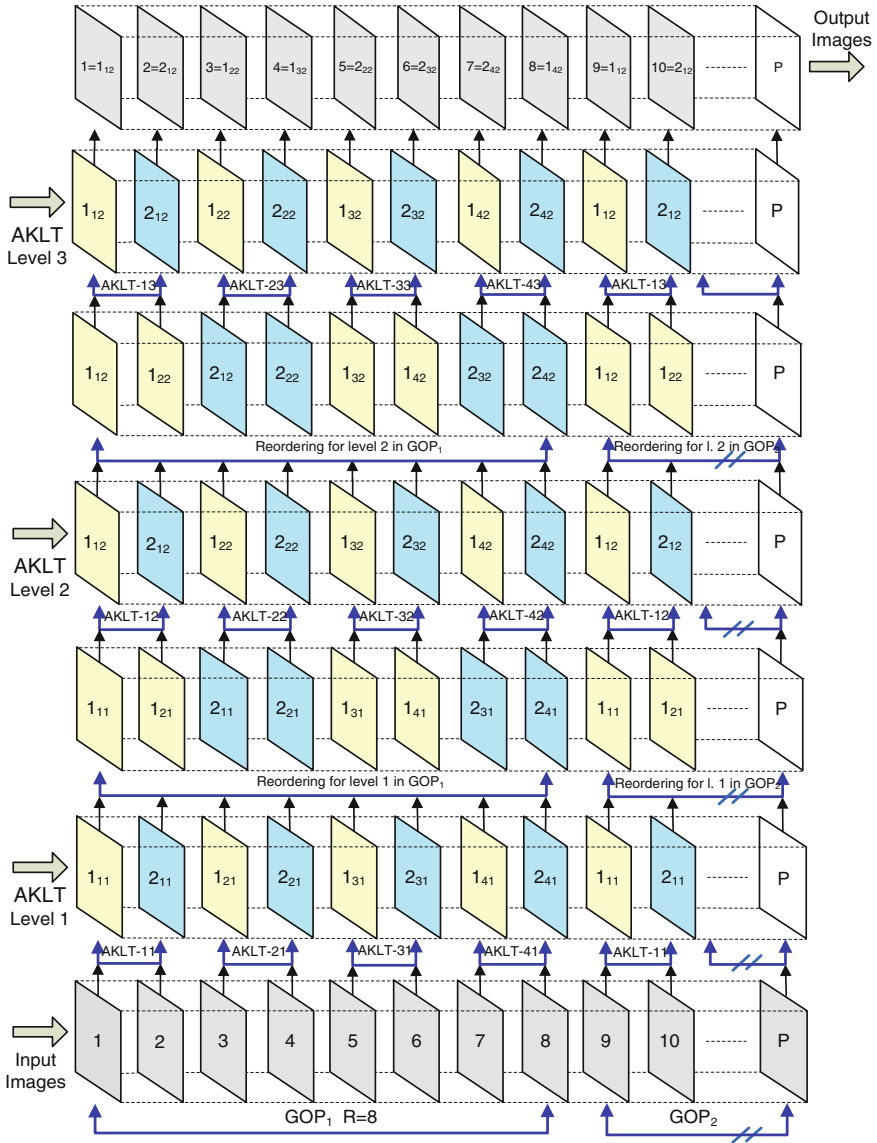


Fig. 4.5 Processing of a group of $R = 8$ images through three-level hierarchical adaptive KLT 2×2

4.4.1 Algebraic Method for Calculation of the AKLT Matrix of Size 3×3

For better explanation, here the structure of a sub-group of three images is shown in Fig. 4.6. From the three images of S pixels each, the vectors $\vec{C}_s = [C_{1s}, C_{2s}, C_{3s}]^T$ for $s = 1, 2, \dots, S$ are obtained. The vectors \vec{C}_s are transformed into vectors $\vec{L}_s = [L_{1s}, L_{2s}, L_{3s}]^T$ through direct AKLT using the matrix $[\Phi]$ of size 3×3 [48] by Eq. 4.17.

$$\begin{bmatrix} L_{1s} \\ L_{2s} \\ L_{3s} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{21} & \Phi_{31} \\ \Phi_{12} & \Phi_{22} & \Phi_{32} \\ \Phi_{13} & \Phi_{23} & \Phi_{33} \end{bmatrix} \begin{bmatrix} (C_{1s} - \bar{C}_1) \\ (C_{2s} - \bar{C}_2) \\ (C_{3s} - \bar{C}_3) \end{bmatrix} \quad \text{for } s = 1, 2, \dots, S \quad (4.17)$$

The elements Φ_{ij} and $\bar{C}_1, \bar{C}_2, \bar{C}_3$ are defined by Eqs. 4.18–4.28.

$$\Phi_{1m} = A_m/P_m \quad \Phi_{2m} = B_m/P_m \quad \Phi_{3m} = D_m/P_m \quad \text{for } m = 1, 2, 3 \quad (4.18)$$

$$A_m = (k_3 - \lambda_m)[k_5(k_2 - \lambda_m) - k_4k_6] \quad B_m = (k_3 - \lambda_m)[k_6(k_1 - \lambda_m) - k_4k_5] \quad (4.19)$$

$$D_m = k_6[2k_4k_5 - k_6(k_1 - \lambda_m)] - k_5^2(k_2 - \lambda_m) \quad P_m = \sqrt{A_m^2 + B_m^2 + D_m^2} \quad (4.20)$$

$$\bar{C}_1 = E(C_{1s}) \quad \bar{C}_2 = E(C_{2s}) \quad \bar{C}_3 = E(C_{3s}) \quad (4.21)$$

$$k_1 = E(C_{1s}^2) - (\bar{C}_1)^2 \quad k_2 = E(C_{2s}^2) - (\bar{C}_2)^2 \quad k_3 = E(C_{3s}^2) - (\bar{C}_3)^2 \quad (4.22)$$

$$k_4 = E(C_{1s}C_{2s}) - (\bar{C}_1)(\bar{C}_2) \quad k_6 = E(C_{2s}C_{3s}) - (\bar{C}_2)(\bar{C}_3) \quad (4.23)$$

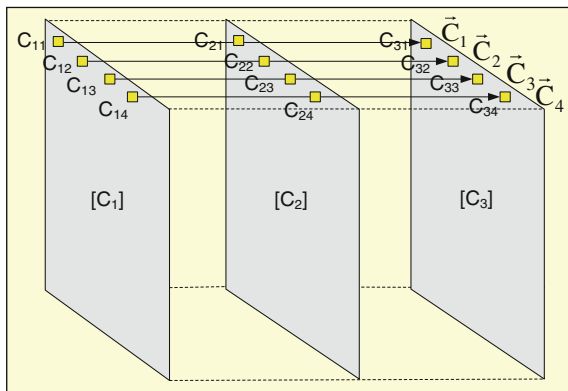
$$k_5 = E(C_{1s}C_{3s}) - (\bar{C}_1)(\bar{C}_3) \quad (4.24)$$

$$\begin{aligned} \lambda_1 &= 2\sqrt{\frac{|p|}{3}} \cos\left(\frac{\varphi}{3}\right) - \frac{a}{3} \\ \lambda_2 &= -2\sqrt{\frac{|p|}{3}} \cos\left(\frac{\varphi + \pi}{3}\right) - \frac{a}{3} \\ \lambda_3 &= -2\sqrt{\frac{|p|}{3}} \cos\left(\frac{\varphi - \pi}{3}\right) - \frac{a}{3} \end{aligned} \quad (4.25)$$

$$\begin{aligned} q &= 2(a/3)^3 - (ab)/3 + c \quad p = -(a^2/3) + b \\ \varphi &= \arccos\left[-q/2 \Big/ \sqrt{(|p|/3)^3}\right] \end{aligned} \quad (4.26)$$

$$a = -(k_1 + k_2 + k_3) \quad b = k_1k_2 + k_1k_3 + k_2k_3 - (k_4^2 + k_5^2 + k_6^2) \quad (4.27)$$

Fig. 4.6 A sub-group of three images from the processed GOP



$$c = k_1 k_6^2 + k_2 k_5^2 + k_3 k_4^2 - (k_1 k_2 k_3 + 2k_4 k_5 k_6) \quad (4.28)$$

Using the inverse AKLT, the vectors \vec{L}_s are transformed into vectors \vec{C}_s (Eq 4.29).

$$\begin{bmatrix} C_{1s} \\ C_{2s} \\ C_{3s} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \\ \Phi_{31} & \Phi_{32} & \Phi_{33} \end{bmatrix} \begin{bmatrix} L_{1s} \\ L_{2s} \\ L_{3s} \end{bmatrix} + \begin{bmatrix} \bar{C}_1 \\ \bar{C}_2 \\ \bar{C}_3 \end{bmatrix} \quad \text{for } s = 1, 2, \dots, S \quad (4.29)$$

The matrix of the inverse AKLT is provided by Eq. 4.30.

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \\ \Phi_{31} & \Phi_{32} & \Phi_{33} \end{bmatrix} = [\Phi]^{-1} = [\Phi]^T \quad (4.30)$$

For the restoration of vectors $\vec{C}_s = [C_{1s}, C_{2s}, C_{3s}]^T$ through inverse AKLT are needed not only the vectors $\vec{L}_s = [L_{1s}, L_{2s}, L_{3s}]^T$, but also the elements Φ_{ij} of the matrix $[\Phi]$ and the values of $\bar{C}_1, \bar{C}_2, \bar{C}_3$ as well. The total number of these elements could be reduced representing the matrix $[\Phi]$ as the product of the matrices $[\Phi_1(\alpha)]$, $[\Phi_2(\beta)]$, $[\Phi_3(\gamma)]$ and the rotation around the coordinate axes for each transformed vector in Euler angles α, β and γ , respectively (Eq 4.31), where $[\Phi_1(\alpha)]$, $[\Phi_2(\beta)]$, $[\Phi_3(\gamma)]$ are calculated by Eq. 4.32.

$$[\Phi] = \begin{bmatrix} \Phi_{11} & \Phi_{21} & \Phi_{31} \\ \Phi_{12} & \Phi_{22} & \Phi_{32} \\ \Phi_{13} & \Phi_{23} & \Phi_{33} \end{bmatrix} = [\Phi_1(\alpha)][\Phi_2(\beta)][\Phi_3(\gamma)] = [\Phi(\alpha, \beta, \gamma)] \quad (4.31)$$

$$\begin{aligned}
[\Phi_1(\alpha)] &= \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
[\Phi_2(\beta)] &= \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \\
[\Phi_3(\gamma)] &= \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{4.32}$$

In this case the elements of the matrix $[\Phi]$ are represented by the relations included in Eq. 4.33.

$$\begin{aligned}
\Phi_{11} &= \cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \gamma & \Phi_{21} &= -\cos \alpha \cos \beta \sin \gamma - \sin \alpha \cos \gamma \\
\Phi_{31} &= -\cos \alpha \sin \beta \\
\Phi_{12} &= \sin \alpha \cos \beta \cos \gamma + \cos \alpha \sin \gamma & \Phi_{22} &= -\sin \alpha \cos \beta \sin \gamma + \cos \alpha \cos \gamma \\
\Phi_{32} &= -\sin \alpha \sin \beta \\
\Phi_{13} &= \sin \beta \cos \gamma & \Phi_{23} &= -\sin \beta \sin \gamma & \Phi_{33} &= \cos \beta
\end{aligned} \tag{4.33}$$

The matrix of the inverse AKLT is defined by Eq. 4.34.

$$[\Phi]^{-1} = [\Phi_3(-\gamma)][\Phi_2(-\beta)][\Phi_1(-\alpha)] \tag{4.34}$$

For the calculation of the elements of the inverse matrix $[\Phi]^{-1}$, it is enough to know the values of the three rotation angles α , β and γ defined by Eq. 4.35.

$$\begin{aligned}
\alpha &= -\arcsin\left(\Phi_{32} / \sqrt{1 - \Phi_{33}^2}\right) & \beta &= \arccos(\Phi_{33}) \\
\gamma &= \arccos\left(\Phi_{13} / \sqrt{1 - \Phi_{33}^2}\right)
\end{aligned} \tag{4.35}$$

In result, the number of the needed values for the calculation of the matrix $[\Phi]^{-1}$ is reduced from 9 down to 3, i.e. 3 times reduction. The elements L_{1s} , L_{2s} , L_{3s} for $s = 1, 2, \dots, S$ comprise the pixels of the first, second, and third eigen image in the sub-group of images with elements C_{1s} , C_{2s} , C_{3s} .

4.4.2 Evaluation of the Decorrelation of the Images Transformed Through HA-KLT 3×3

For the first level of HA-KLT 3×3 , the corresponding covariance matrices of size 3×3 for each group of vectors $\vec{C}_{s,p}$ for $p = 1, 2, 3$ and $s = 1, 2, \dots, S$, are computed by Eqs. 4.36–4.38, respectively (here $\lambda_1^{1,p}$, $\lambda_2^{1,p}$, $\lambda_3^{1,p}$ are the corresponding eigen values of the covariance matrices $[K_L^{1,p}]$), where $[K_{C_{1,p}}]$ is defined by Eq. 4.39.

$$[K_L^{1,1}] = [\Phi_1][K_{C_{1,1}}][\Phi_1]^t = \begin{bmatrix} \lambda_1^{1,1} & 0 & 0 \\ 0 & \lambda_2^{1,1} & 0 \\ 0 & 0 & \lambda_3^{1,1} \end{bmatrix} \quad (4.36)$$

$$[K_L^{1,2}] = [\Phi_2][K_{C_{1,2}}][\Phi_2]^t = \begin{bmatrix} \lambda_1^{1,2} & 0 & 0 \\ 0 & \lambda_2^{1,2} & 0 \\ 0 & 0 & \lambda_3^{1,2} \end{bmatrix} \quad (4.37)$$

$$[K_L^{1,3}] = [\Phi_3][K_{C_{1,3}}][\Phi_3]^t = \begin{bmatrix} \lambda_1^{1,3} & 0 & 0 \\ 0 & \lambda_2^{1,3} & 0 \\ 0 & 0 & \lambda_3^{1,3} \end{bmatrix} \quad (4.38)$$

$$\begin{aligned} [K_{C_{1,p}}] &= \frac{1}{S} \sum_{s=1}^S \vec{C}_{s,p} \vec{C}_{s,p}^t - \vec{m}_{c,p} \vec{m}_{c,p}^t \\ &= \begin{bmatrix} k_{11}(p) & k_{12}(p) & k_{13}(p) \\ k_{21}(p) & k_{22}(p) & k_{23}(p) \\ k_{31}(p) & k_{32}(p) & k_{33}(p) \end{bmatrix} \quad \text{for } p = 1, 2, 3 \end{aligned} \quad (4.39)$$

The covariance matrix of size 9×9 of the 9-component vectors \vec{C}_s , in the group of initial images GOP in the first level of the HA-KLT 3×3 is represented by Eq. 4.40, where $[K_{L_k, L_p}^1]$ is the cross-correlation matrix of size 3×3 of the 3-component vectors \vec{L}_{sk}^1 and \vec{L}_{sp}^1 , from the groups k and p in the first level of HA-KLT 3×3 after rearrangement of the calculated eigen images and is computed by Eq. 4.41.

$$\begin{aligned}
[K_L^1] &= \frac{1}{S} \sum_{s=1}^S \vec{C}_s \vec{C}_s^t - \vec{m}_c \vec{m}_c^t \\
&= \begin{bmatrix} \begin{bmatrix} \lambda_1^{1,1} & 0 & 0 \\ 0 & \lambda_2^{1,1} & 0 \\ 0 & 0 & \lambda_3^{1,1} \end{bmatrix} & [K_{L_1, L_2}^1] & [K_{L_1, L_3}^1] \\ [K_{L_1, L_2}^1] & \begin{bmatrix} \lambda_1^{1,2} & 0 & 0 \\ 0 & \lambda_2^{1,2} & 0 \\ 0 & 0 & \lambda_3^{1,2} \end{bmatrix} & [K_{L_2, L_3}^1] \\ [K_{L_1, L_3}^1] & [K_{L_2, L_3}^1] & \begin{bmatrix} \lambda_1^{1,3} & 0 & 0 \\ 0 & \lambda_2^{1,3} & 0 \\ 0 & 0 & \lambda_3^{1,3} \end{bmatrix} \end{bmatrix}
\end{aligned} \tag{4.40}$$

$$[K_{L_k, L_p}^1] = E\{\vec{L}_{sk}^1 \cdot (\vec{L}_{sp}^1)^T\} - E\{\vec{L}_{sk}^1\}E\{(\vec{L}_{sp}^1)^T\} \quad \text{for } k, p = 1, 2, 3, k \neq p \tag{4.41}$$

For the second level of the HA-KLT 3×3 the covariance matrices of size 3×3 for each group of rearranged vectors, obtained after the first HA-KLT level is executed, are calculated by Eq. 4.42.

$$\begin{aligned}
[K_L^{2,1}] &= \begin{bmatrix} \lambda_1^{2,1} & 0 & 0 \\ 0 & \lambda_2^{2,1} & 0 \\ 0 & 0 & \lambda_3^{2,1} \end{bmatrix} & [K_L^{2,2}] &= \begin{bmatrix} \lambda_1^{2,2} & 0 & 0 \\ 0 & \lambda_2^{2,2} & 0 \\ 0 & 0 & \lambda_3^{2,2} \end{bmatrix} \\
[K_L^{2,3}] &= \begin{bmatrix} \lambda_1^{2,3} & 0 & 0 \\ 0 & \lambda_2^{2,3} & 0 \\ 0 & 0 & \lambda_3^{2,3} \end{bmatrix}
\end{aligned} \tag{4.42}$$

The covariance matrix of size 9×9 of the 9-component vectors for the GOP in the second transform level is represented by Eq. 4.43, where $[K_{L_k, L_p}^2]$ is the mutual cross-correlation matrix of size 3×3 for the 3-component vectors \vec{L}_{sk}^2 and \vec{L}_{sp}^2 from the groups k and p in the second level of HA-KLT 3×3 after the rearrangement of the obtained eigen images and is computed by Eq. 4.44.

$$[K_L^2] = \begin{bmatrix} \begin{bmatrix} \lambda_1^{2,1} & 0 & 0 \\ 0 & \lambda_2^{2,1} & 0 \\ 0 & 0 & \lambda_3^{2,1} \end{bmatrix} & [K_{L_1, L_2}^2] & [K_{L_1, L_3}^2] \\ [K_{L_1, L_2}^2] & \begin{bmatrix} \lambda_1^{2,2} & 0 & 0 \\ 0 & \lambda_2^{2,2} & 0 \\ 0 & 0 & \lambda_3^{2,2} \end{bmatrix} & [K_{L_2, L_3}^2] \\ [K_{L_1, L_3}^2] & [K_{L_2, L_3}^2] & \begin{bmatrix} \lambda_1^{2,3} & 0 & 0 \\ 0 & \lambda_2^{2,3} & 0 \\ 0 & 0 & \lambda_3^{2,3} \end{bmatrix} \end{bmatrix} \quad (4.43)$$

$$[K_{L_k, L_p}^2] = E\{\vec{L}_{sk}^2 \cdot \vec{L}_{sp}^2\}^T - E\{\vec{L}_{sk}^2\}E\{\vec{L}_{sp}^2\}^T \quad \text{for } k, p = 1, 2, 3; k \neq p \quad (4.44)$$

On the basis of the so obtained matrix $[K_L^2]$, it could be evaluated the decorrelation degree of the corresponding eigen images in the processed GOP.

When the full decorrelation of these images is achieved, the matrix $[K_L^2]$ is diagonal, for which is necessary to be satisfied the condition from Eq. 4.45.

$$[K_{L_k, L_p}^2] = E\{\vec{L}_{sk}^2 \cdot (\vec{L}_{sp}^2)^T\} - E\{\vec{L}_{sk}^2\}E\{(\vec{L}_{sp}^2)^T\} = [0] \quad \text{for } k, p = 1, 2, 3; k \neq p \quad (4.45)$$

The limitation of the number of HA-KLT 3×3 levels could be done even without achieving the full decorrelation in the processed group of images, if Eq. 4.46, where δ is a pre-selected threshold with a small value, is satisfied.

$$|E\{\vec{L}_k^2 \cdot (\vec{L}_p^2)^T\} - E\{\vec{L}_k^2\}E\{(\vec{L}_p^2)^T\}| \geq \delta \quad (4.46)$$

4.4.3 Algebraic Method for Determination of the AKLT Matrix of Size 2×2

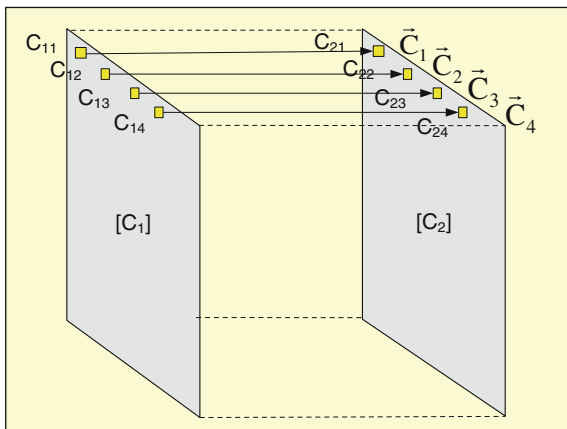
From each two digital images of S pixels each, shown in Fig. 4.7, the vectors $\vec{C}_s = [C_{1s}, C_{2s}]^T$ for $s = 1, 2, \dots, S$ are defined.

Each vector \vec{C}_s is transformed into the corresponding vectors $\vec{L}_s = [L_{1s}, L_{2s}]^T$ through direct AKLT, using the matrix $[\Phi]$ of size 2×2 by Eq. 4.47, where $\vec{m}_c = [\bar{C}_1, \bar{C}_2]^T$ is the mean vector with components $\bar{C}_1 = E(C_{1s})$ and $\bar{C}_2 = E(C_{2s})$.

$$\begin{bmatrix} L_{1s} \\ L_{2s} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{21} \\ \Phi_{12} & \Phi_{22} \end{bmatrix} \begin{bmatrix} (C_{1s} - \bar{C}_1) \\ (C_{2s} - \bar{C}_2) \end{bmatrix} \quad (4.47)$$

Through the inverse AKLT, the vectors \vec{L}_s ($s = 1, 2, \dots, S$) are transformed into the vectors \vec{C}_s provided by Eq. 4.48.

Fig. 4.7 A sub-group of two images from the original GOP



$$\begin{bmatrix} C_{1s} \\ C_{2s} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \begin{bmatrix} L_{1s} \\ L_{2s} \end{bmatrix} + \begin{bmatrix} \bar{C}_1 \\ \bar{C}_2 \end{bmatrix} \quad (4.48)$$

The elements Φ_{ij} of matrix $[\Phi]$ are computed by Eq. 4.49 with parameters defined in Eqs. 4.50–4.51.

$$[\Phi] = \begin{bmatrix} \Phi_{11} & \Phi_{21} \\ \Phi_{12} & \Phi_{22} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (4.49)$$

$$\theta = \text{arctg} \left(\frac{\Phi_{21}}{\Phi_{11}} \right) = \text{arctg} \left(\frac{2k_3}{k_1 - k_2 + \sqrt{(k_1 - k_2)^2 + 4k_3^2}} \right) = \text{arctg} \left(\frac{\beta}{\alpha + \gamma} \right) \quad (4.50)$$

$$\begin{aligned} k_1 &= E(C_{1s}^2) - (\bar{C}_1)^2 & k_2 &= E(C_{2s}^2) - (\bar{C}_2)^2 & k_3 &= E(C_{1s}C_{2s}) - (\bar{C}_1)(\bar{C}_2) \\ \alpha &= k_1 - k_2 & \beta &= 2k_3 & \gamma^2 &= \alpha^2 + \beta^2 \end{aligned} \quad (4.51)$$

The eigen values of the covariance matrix of the vectors \vec{C}_s are calculated by Eq. 4.52.

$$\begin{aligned} \lambda_1 &= \frac{1}{2} \left[(k_1 + k_2) + \sqrt{(k_1 - k_2)^2 + 4k_3^2} \right] \\ \lambda_2 &= \frac{1}{2} \left[(k_1 + k_2) - \sqrt{(k_1 - k_2)^2 + 4k_3^2} \right] \end{aligned} \quad (4.52)$$

4.5 HA-KLT Applications for Sequences of Motion-Compensated TV Frames

The AKLT transform with a 3×3 matrix could be used to enhance the video compression efficiency in accordance with the standards MPEG-1/2/4 and H.26x. For this, from each GOP/frames in the TV video sequence should be extracted the P frames only, which are motion compensated.

Practically, the number of frames in one GOP is usually 12, and correspondingly, the number of P -frames is 3. On each sub-group of frames P_1, P_2, P_3 could be applied the direct AKLT with a matrix of size 3×3 , in result of which are obtained the frames L_1, L_2, L_3 in the sub-group, which substitute the corresponding P -frames in the GOP. In the first frame L_1 is concentrated the main part of the energy in the three P -frames, which permits to increase the compression after entropy coding for the frames L_1, L_2, L_3 . In the decoding part, the inverse AKLT with a matrix of size 3×3 is executed on the frames L_1, L_2, L_3 , and, as a result, the frames P_1, P_2, P_3 are restored. The direct transform of the frames in the original sequence is shown in Fig. 4.8.

The vectors $\vec{C}_s = [C_{1s}, C_{2s}, C_{3s}]^T$ for $s = 1, 2, \dots, S$, calculated from frames P_1, P_2, P_3 , extracted from the processed GOP for the first 4 pixels: $\vec{C}_1 = [C_{11}, C_{21}, C_{31}]^T$, $\vec{C}_2 = [C_{12}, C_{22}, C_{32}]^T$, $\vec{C}_3 = [C_{13}, C_{23}, C_{33}]^T$, $\vec{C}_4 = [C_{14}, C_{24}, C_{34}]^T$ are represented in Fig. 4.9.

For the example from Fig. 4.9, the pixels C_{12}, C_{13}, C_{14} in the frame P_1 are on the surface of a moving object (or of the moving background, if a moving TV camera is used). In this case, it is necessary to define the position of these pixels in the next frames P_2 and P_3 by using one of the Motion Compensation (MC) algorithms [3, 4]. As a result, the movement vector for every pixel (or for a block of pixels), on the basis of which is defined the position of the pixel (block) in the next frame, is calculated. In case that the vectors $\vec{C}_s = [C_{1s}, C_{2s}, C_{3s}]^T$ for the sub-group of frames P_1, P_2, P_3 are defined, the applying of the MC and the efficiency of the AKLT in respect of the decorrelation in the image sub-group is significantly enhanced. Together with this, the ability to get higher compression ratio is increased, if this information is losslessly coded. The described approach for enhancement of the compression efficiency for TV video sequences, obtained in accordance with the standards of the kind MPEG-1/2/4, H.26x, could be generalized for all frames in same GOP. For this, the similar three-level HA-KLT shown in Fig. 4.4 could be applied on the sequence of 12 frames in the GOP. In result, it is opened the ability to get a higher compression ratio of a TV video sequence than that the obtained one with the approach shown in Fig. 4.9, where the hierarchical transform is not used. This new approach will be further investigated in theory and experimentally.

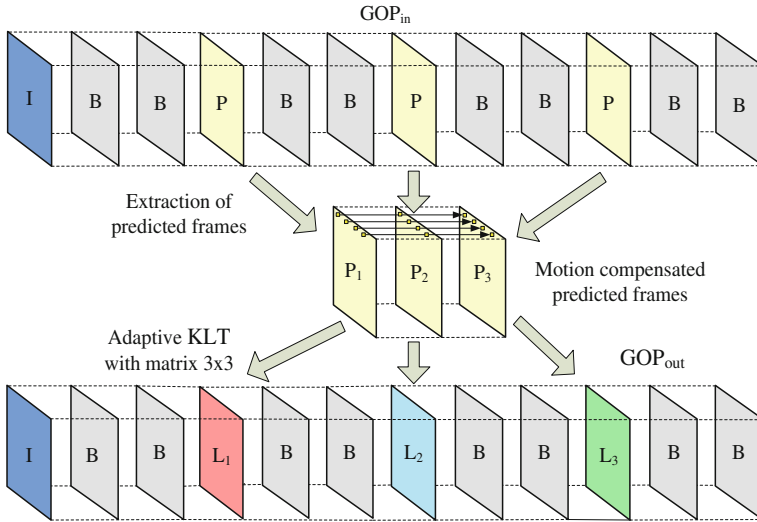
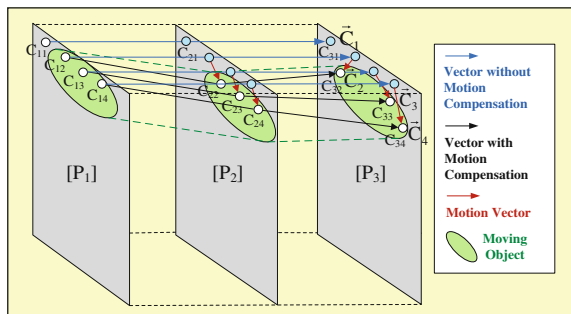


Fig. 4.8 APCA transform for a sub-group of MC P-frames extracted from the GOP

Fig. 4.9 Sub-group of three P-images from the original GOP



4.6 Evaluation of the Computational Complexity of the Two-Level HA-KLT 3×3 for a Group of Images

The computational complexity of the two-level HA-KLT algorithm based on 3×3 matrices is compared with that of the KLT algorithm with a matrix of size 9×9 , because the two-level HA-KLT is equivalent to the KLT with a 9-component vector. For this, both algorithms are compared in respect to the number of operations O (additions and multiplications) [49] needed for the calculation of the following components:

- The covariance matrices $[K_C]$ —in total 6 for the first algorithm, each of size 3×3 , and one matrix $[K_C]$ of size 9×9 —for the second algorithm.
- The eigen values and eigen vectors of the corresponding matrices $[K_C]$.

- The eigen images of each obtained GOP by using both algorithms.

On the basis of the computational complexity analysis given in [44] for the AKLT with a matrix of size 3×3 and for the KLT with a matrix of size $R \times R$, it follows that for the HA-KLT with matrices of size 3×3 and for the KLT with a matrix of size 9×9 we have:

- The number of operations needed for the calculation of the elements k_{ij} for all 6 matrices $[K_C]$ of size 3×3 (for the HA-KLT) and for one matrix $[K_C]$ of size 9×9 (for the KLT), is determined by Eqs. 4.53–4.54.

$$O_k(R)|_{R=3} = 3R(R+1)[R(R-1) + 2(R+2)] = 576 \quad (4.53)$$

$$O_k(R)|_{R=9} = (1/2)R(R+1)[R(R-1) + 2(R+2)] = 4,230 \quad (4.54)$$

- The number of operations needed for the calculation of the eigen values of the HA-KLT matrices $[K_C]$ and of the KLT matrix $[K_C]$, when the QR decomposition and the Householder transform of $(R-1)$ steps were used [50], is calculated in Eqs. 4.55–4.56.

$$O_{val}(R)|_{R=3} = 282 \quad (4.55)$$

$$O_{val}(R)|_{R=9} = (R-1)\left(\frac{4}{3}R^2 + \frac{17}{6}R + 7\right) = 1,124 \quad (4.56)$$

- The number of operations needed for the calculation of the eigen vectors of the matrices $[K_C]$ for the HA-KLT and for the matrix $[K_C]$ of KLT, in case that iterative algorithm with four iterations is used, is defined by Eqs. 4.57–4.58.

$$O_{vec}(R)|_{R=3} = 275 \quad (4.57)$$

$$O_{vec}(R)|_{R=9} = R[2R(4R+5) - 1] = 6,633 \quad (4.58)$$

- The number of operations needed for the calculation of a group of nine eigen images (each of S pixels) and obtained in result of the direct HA-KLT and the KLT for zero mean vectors, is provided by Eqs. 4.59–4.60.

$$O_{HA-KLT}(R)|_{R=3} = 6SR(2R-1) = 90S \quad (4.59)$$

$$O_{KLT}(R)|_{R=9} = SR(2NR-1) = 153S \quad (4.60)$$

Then the Total number of Operations (TO), needed for the HA-KLT and for the KLT, is computed by Eqs. 4.61–4.62.

$$TO_1(3) = [O_k(3) + O_{val}(3) + O_{vec}(3) + O_{HA-KLT}(3)] = 1,133 + 90S \quad (4.61)$$

$$TO_2(9) = [O_k(9) + O_{val}(9) + O_{vec}(9) + O_{KLT}(9)] = 11,996 + 153S \quad (4.62)$$

The reduction of the total number of operations needed for the HA-KLT and compared to that of the KLT, could be evaluated using the coefficient $\eta(S)$ (Eq 4.63).

$$\eta(S) = \frac{TO_2(9)}{TO_1(3)} = \frac{11,996 + 153S}{1,133 + 90S} \quad (4.63)$$

For example, for $S = 100$ $\eta(100) = 2.96$, for $S = 1,000$ $\eta(1,000) = 1.81$, and for $S = 1,000$ $\eta(\infty) \rightarrow 1.7$, respectively. Hence, $TO_1(S)$ is at least 1.7 times smaller than $TO_2(S)$ for each value of S (in average, about 2 times). For higher values of N (the number of images in one group) in the range from 9 up to 16 and for larger values of S , the coefficient $\eta(S)$ increases from 1.7 up to 2.1.

The memory volume needed for the KLT execution on a group of R images each of S pixels is $R^2 + RS + R = R(R + S + 1)$ in accordance with [37]. For the execution of the HA-KLT based on sub-groups of three images and combined with recursive calculation of the hierarchical levels, the needed memory is $3[(R/3)^2 + (R/3)S + (R/3)] = R[R/3 + S + 1]$. From this follows, that the HA-KLT does not need more memory than the KLT. Something more—in fact, there is a small reduction of the needed memory, as it could be seen from the relations, given above.

4.7 Experimental Results for Group of Consecutive Images

On the basis of the two-levels HA-KLT algorithm, shown in Fig. 4.4, various experiments were executed with sequences of medical Computer Tomography (CT) images of size 512×512 pixels, 8 bpp. In case, that the model of the exponential approximation of the function $\rho_{0,u} = f(u)$ is assumed [51], which in accordance with Eq. 4.15 represents the normalized cross-correlation coefficient of the images in the group u , then the optimum group length R is defined by the so-called “range of statistical dependence”, for which $\rho_{0,u}$ goes down to $\delta = 1/e = 0.37$, compared to its maximum value $\rho_{0,0} = 1$. The averaged relation $\rho_{0,u} = f(u)$ obtained experimentally for the examined test sequence of CT medical images is shown in Fig. 4.10.

In this case from Eq. 4.16 it follows, that the group length is $R = 9$. In accordance with this, the test sequence was divided into groups (Set 1, ..., Set 7), containing $R = 9$ consecutive CT images each. As an example, one of the groups—Set 3, which contains CT image 1, 2, ..., 9, is shown in Fig. 4.11. The eigen images obtained after applying the two-level HA-KLT on the test group from Set 3 are shown in Fig. 4.12.

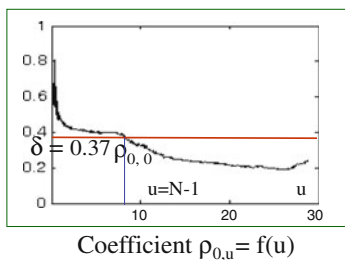


Fig. 4.10 Setting the group length $R = 9$, through comparing the value of the coefficient $\rho_{0,u}$ and the threshold $\delta = \rho_{0,0}/e = 0.37$

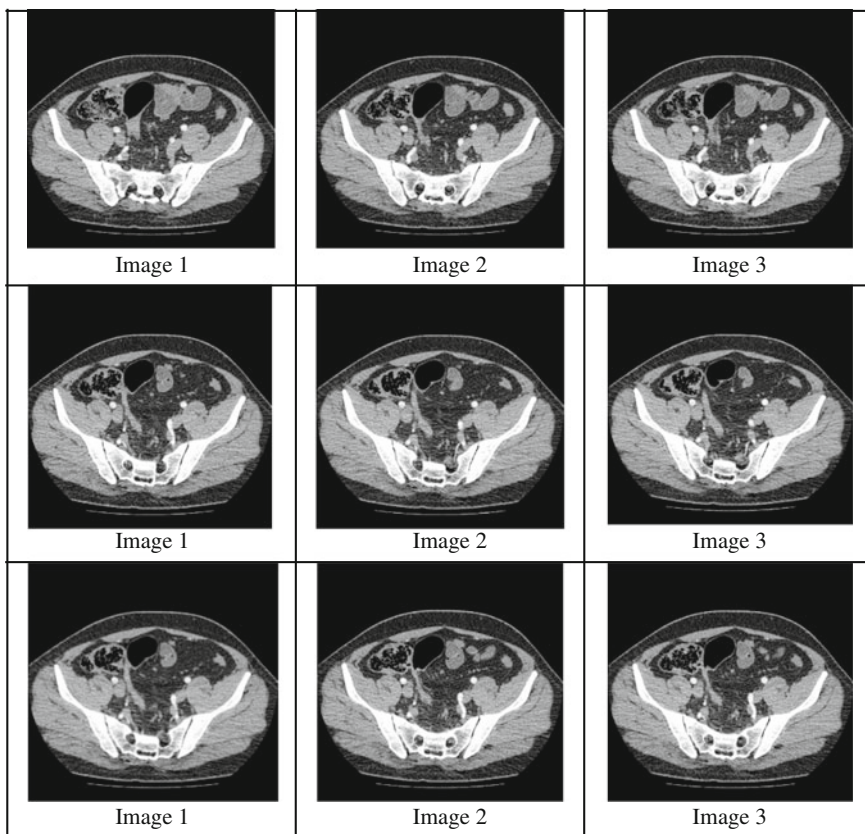


Fig. 4.11 Group of $R = 9$ consecutive CT images from Set 3

As it could be seen from Fig. 4.12, the main part of the energy of all nine images is concentrated in the first eigen image, and the energy of each next eigen image is decreased quickly. This is confirmed by the data shown in Table 4.2, for the power

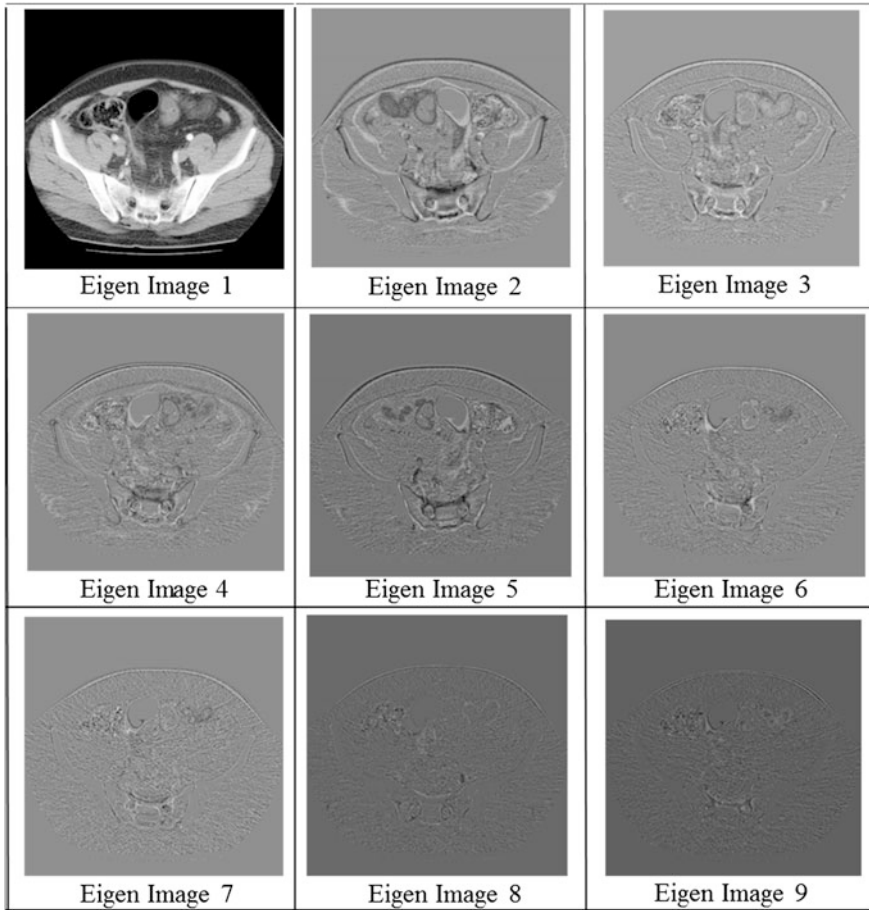


Fig. 4.12 Eigen images obtained for Set 3 after performing two-levels HA-KLT

distribution of the pixels of the eigen images from Set 3 after the first and second HA-KLT level, before and after their rearrangement (Fig. 4.4).

In Table 4.2, the power distribution of all eigen images in Set 3 before and after each operation, and the relative mean power distribution are represented. On the basis of the data given in Table 4.2, the corresponding graphics shown in Figs. 4.13, 4.14, and 4.15, which represent the power distribution of all nine eigen images, are built.

The mean and the relative mean power distribution of the pixels from all nine eigen images in Set 1, ..., Set 7 are given in Table 4.3. The data in the last column show, that in the first three eigen images are concentrated 95.7 % of the total mean power of all nine images in the GOP. From Table 4.3, it follows that the mean power of the first eigen image for all sets is more than 250 times larger than that of

Table 4.2 Power distribution of all eigen images from Set 3, before and after each operation

Eigen image	Level 1 (not arranged)	Level 1 (arranged)	Level 2 (not arranged)	Level 2 (arranged)	Relative mean
1	18,170	18,170	53,041	53,041	220
2	715	18,056	686	1,100	5
3	341	18,029	316	686	3
4	18,056	715	1,100	710	3
5	748	389	710	316	1
6	389	694	305	305	1
7	18,029	341	523	523	2
8	694	389	326	326	1
9	394	394	242	242	1

Fig. 4.13 Power distribution for Set 3, level 1: **a** not arranged, **b** arranged

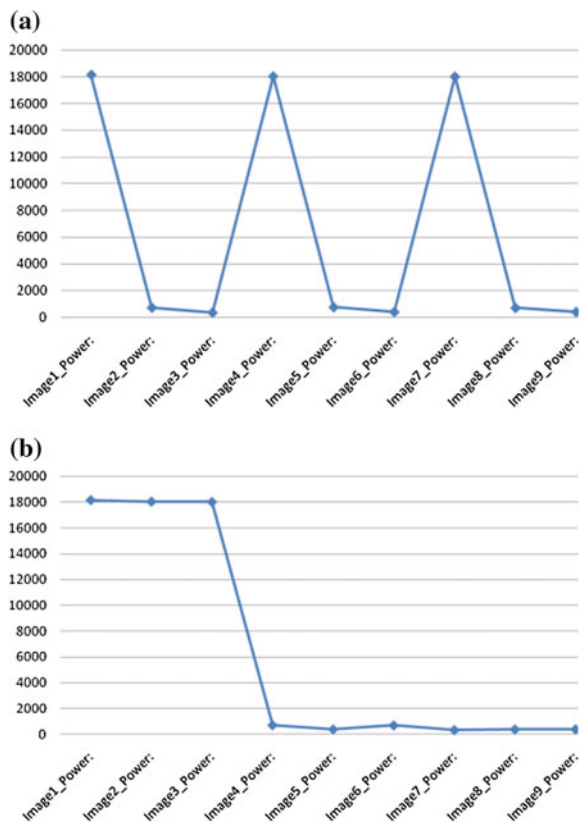


Fig. 4.14 Power distribution for Set 3, level 2: **a** not arranged, **b** arranged

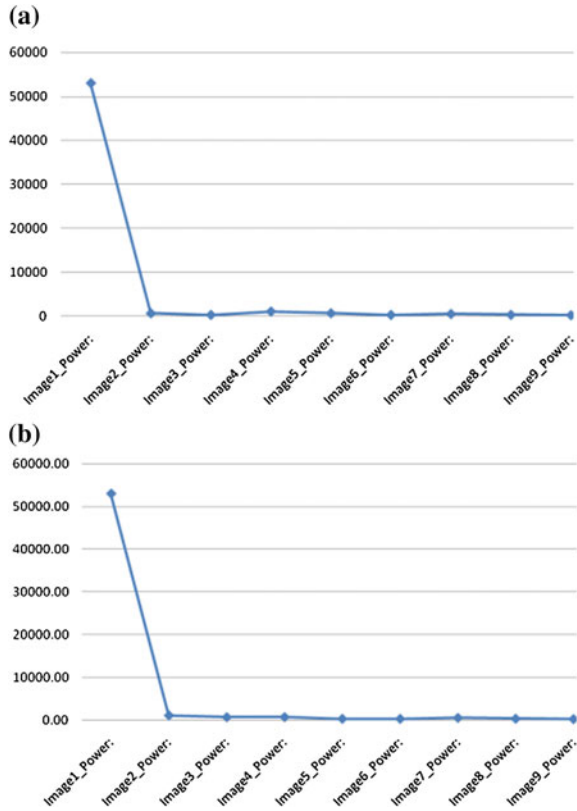
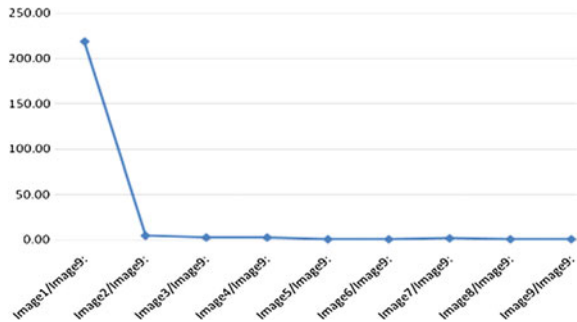


Fig. 4.15 Relative mean power distribution for Set 3, level 2 (arranged)



each of the next eight eigen images. The experimental results were obtained with the software implementation of HA-KLT in Visual C and Windows environment.

The values for the pixels of the eigen images obtained in result of the direct two-level HA-KLT were calculated with full accuracy, and after corresponding rounding they were transformed into 8-bit numbers. If on the 8 bpp eigen images is applied the inverse two-level HA-KLT, then the quality of corresponding restored

Table 4.3 Power distribution, mean power distribution (M), relative mean power distribution (RM), and relative mean % (RM %) of power distribution for all eigen images (EI) in Set 1, Set 2, ..., Set 7

EI	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	M	RM	RM %
1	49,992	49,749	53,041	53,547	53,774	43,272	37,701	48,725	259.6	91.4
2	949	811	1,100	875	2,331	1,770	1,094	1,276	6.8	93.8
3	683	2,325	686	1,062	625	834	1,144	1,051	5.6	95.7
4	808	710	710	512	460	811	950	709	3.8	97.1
5	522	566	316	425	300	442	364	419	2.2	97.8
6	350	529	305	306	317	402	435	378	2.0	98.6
7	206	222	523	317	554	306	430	365	1.9	99.2
8	172	198	326	261	312	251	218	248	1.3	99.6
9	130	171	242	173	254	167	177	188	1.0	100.0

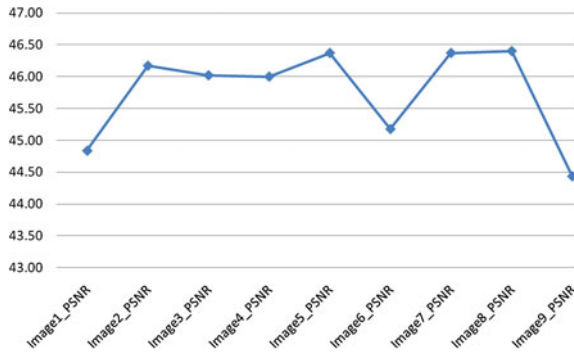


Fig. 4.16 Evaluation of the quality (PSNR in dB) of the restored images from Set 3 after the inverse two-levels HA-KLT executed on the eigen images from Fig. 4.11

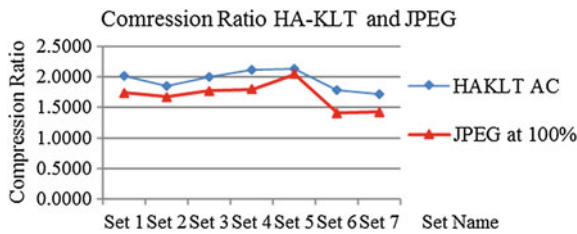


Fig. 4.17 Evaluation of the compression ratio for sequences of CT images Set 1, ..., Set 7 processed with two-level HA-KLT and JPEG with the highest quality

images in the GOP evaluated by their Peak Signal-to-Noise Ratio (PSNR) is ≥ 45 dB. This was confirmed by the results from Fig. 4.16 obtained for the eigen images in Set 3 of Fig. 4.11 after the inverse HA-KLT in correspondence with the algorithm and shown in Fig. 4.4. Hence, the sequence of 9 images could be restored with retained visual quality. This result illustrates the ability for efficient compression of a sequence of CT images, when the HA-KLT is used.

The results for the compression ratio of the test CT images (Set 1, ..., Set 7), obtained through two-level HA-KLT combined with arithmetic coding and through JPEG with 100 % quality of the restored images are compared by plots in Fig. 4.17. The results show higher compression ratio of the new method for same quality of the restored images.

The basic qualities of the new HA-KLT for image sequences are following:

1. The lower computational complexity than KLT for the whole GOP due to the lower complexity of the AKLT compared to the case, for which for the calculation of the KLT matrix are used numerical methods [50].
2. The structure of the HA-KLT algorithm is suitable for parallel implementations, which is of high importance for all applications aimed at real time processing of image sequences.

3. The HA-KLT could be combined with the Branched Inverse Pyramid Decomposition (BIPD) [52] with nonlinear pre-processing and post-processing based on the pixel-by-pixel Adaptive Histogram Matching (AHM) transform for achievement of efficient compression.
4. The HA-KLT could be also used for efficient lossless compression of image sequences in the case, when the number of eigen images in the group is not reduced.
5. There is also a possibility for further development of the HA-KLT algorithms through use of Integer KLT for lossless coding of image sequences by analogy with [53–54] in compression of video sequences from stationary TV camera, compression of multi-view images, image fusion, face recognition, etc.

4.8 Principle of the Hierarchical Adaptive KL-Based Transform for Single Image

The kernel of the HA-KLT for single images is the AKLT, which is one-dimensional. It is related to halftone images, divided into blocks of size 2×2 (for $n = 1$). In this case, from each row of the matrix image $[C]$ of size $N \times N$ ($N = 2^p$) the number $S = N^2/2 = 2^{2p-1}$ of two-component vectors $\vec{C}_s = [C_{1s}, C_{2s}]^T$, $s = 1, 2, \dots, S$ could be defined. For example, from the image $[C]$ of size 4×4 shown in Fig. 4.18 the $S = 8$ horizontally oriented vectors are obtained: $\vec{C}_1 = [C_{11}, C_{21}]^T$, $\vec{C}_2 = [C_{12}, C_{22}]^T, \dots, \vec{C}_8 = [C_{18}, C_{28}]^T$. Each vector is transformed into the corresponding vectors $\vec{L}_s = [L_{1s}, L_{2s}]^T$ through AKLT with a matrix $[\Phi]$, of size 2×2 (AKLT 2×2). To apply the 2D-KLT on a halftone image, its matrix is divided into blocks, for which is twice executed the 1D-KLT (first, on the rows and after that on the columns of the transformed blocks).

The basic difficulty in the execution of the 1D-KLT on each row or column for a block of large size is related to the high computational complexity of the transform matrix. To solve this problem, in [46] it is offered the 2D HA-KLT for halftone images based on the AKLT 2×2 . The principle of this transform is as follows: all blocks (each of size $2^n \times 2^n$) of the matrix image $[C]$ of size $N \times N$ are transformed

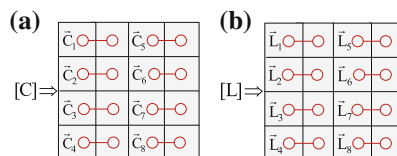


Fig. 4.18 An example matrix image of size 4×4 , which contains eight vectors: **a** before transform, **b** after the AKLT with a 2×2 matrix

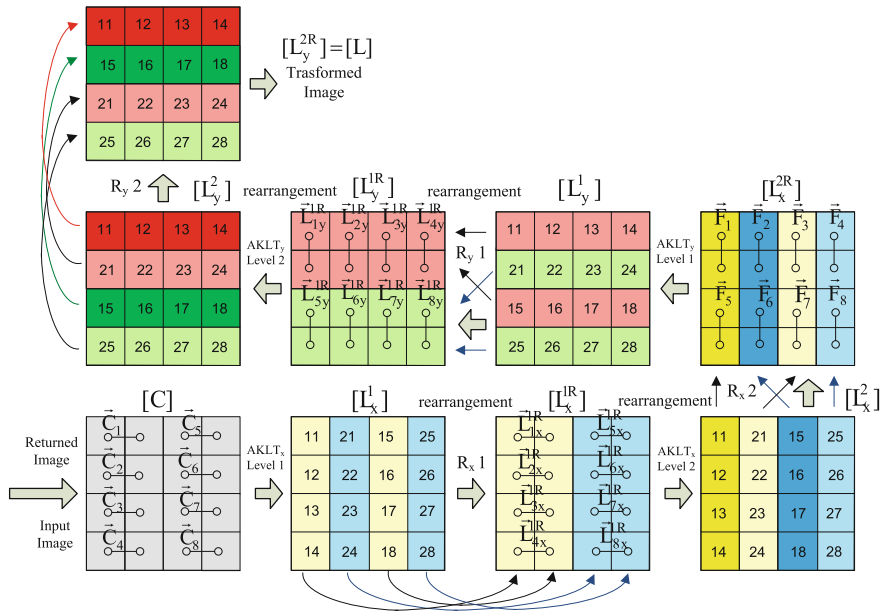


Fig. 4.19 2D HA-KLT for a block of size 4×4

sequentially (by their rows and columns) in n consecutive steps by using KL matrices of size 2×2 . For this, from each row of the processed block a vector of 2^n components is obtained. The number of all vectors obtained from one block is 2^n , and from all blocks it is $2^n \times (N^2/2^n)$. The components of each vector are divided into 2^{n-1} groups, from which are obtained 2-component vectors or $N^2/2^n$ vectors in total. On the vectors from each group the adaptive KLT with a matrix of size 2×2 (AKLT 2×2) is applied. The transformed 2^n components of each vector, which correspond to one row of the block, are rearranged so that their neighboring components have maximum mutual correlation. On all rearranged vectors of 2^n components each, the next step of the n -step transform is applied in similar way, and etc. When all n steps on the rows are finished, it is executed again on the columns of such transformed blocks. Each AKLT 2×2 starts with the verification of the condition $\beta \neq 0$ (in accordance with Eq. 4.51). In case that it is not satisfied, the next steps of the hierarchical transform in the selected direction (x or y) are not executed. In result, a spatial decorrelation of the elements of the transformed blocks is achieved. The original image $[C]$ is restored applying the inverse 2D HA-KLT on the matrix of size $N \times N$ by using all transformed blocks.

To illustrate the principle described above, it is shown an example for 2D HA-KLT of the image $[C]$ of size 4×4 pixels ($p = 2$) in Fig. 4.19. In this case, in the first stage of the transform it is necessary to apply the AKLT 2×2 two times sequentially on all two-component vectors oriented horizontally and then (in the second stage)—on all transformed vectors oriented vertically. After each AKLT 2×2 , one half of the columns or rows should be rearranged.

The rearrangement with a group of four arrows oriented horizontally or vertically is shown in Fig. 4.18.

Each arrow shows the position of a given column or a row, before and after the rearrangement. In Fig. 4.19, the figures of the 8 two-component vectors, on which the AKLT 2×2 for each level is applied, are also shown. The first two components of each transformed vector are marked with two numbers, which indicate the number of the component and this vector, respectively. The wide arrows show the transform directions of the matrix $[C]$ for each level of the 2D HA-KLT. In result of the execution of all levels of HA-KLT on the rows and columns, the final transformed matrix $[L]$ is obtained. The algorithm for the inverse 2D HA-KLT is executed in accordance with Fig. 4.19, but in this case all arrows should be in inverse directions.

For the example from Fig. 4.18, the direct 2D HA-KLT for the block $[C]$ of size 4×4 executes the following steps below:

- Step 1. From the elements of the matrix $[C]$ shown in Fig. 4.18, all horizontally oriented two-component vectors $\vec{C}_s = [C_{1s}, C_{2s}]^T$, $s = 1, 2, \dots, 8$ are defined.
- Step 2. In the first level, the direct AKLT $_x 2 \times 2$ is applied on all vectors \vec{C}_s by Eq. 4.64, where Φ_{ji}^{1x} , \bar{C}_1 , \bar{C}_2 , are defined through vectors \vec{C}_s on the basis of the Eqs. 4.50–4.51.

$$\begin{bmatrix} L_{1s}^{1x} \\ L_{2s}^{1x} \end{bmatrix} = \begin{bmatrix} \Phi_{11}^{1x} & \Phi_{21}^{1x} \\ \Phi_{12}^{1x} & \Phi_{22}^{1x} \end{bmatrix} \begin{bmatrix} (C_{1s} - \bar{C}_1) \\ (C_{2s} - \bar{C}_2) \end{bmatrix} \quad (4.64)$$

In result of the execution of Eq. 4.64, the matrix $[L_x^1]$ is obtained.

- Step 3. Half of the columns of the matrix $[L_x^1]$ are rearranged, and in result the matrix $[L_x^{1R}]$ is obtained. The relation between these two matrices is presented by Eq. 4.65 or Eq. 4.66, where $[P_x]$ is a matrix of size 4×4 used for the rearrangement of the columns of $[L_x^1]$.

$$\begin{bmatrix} L_{11}^{1R_x} & L_{12}^{1R_x} & L_{13}^{1R_x} & L_{14}^{1R_x} \\ L_{21}^{1R_x} & L_{22}^{1R_x} & L_{23}^{1R_x} & L_{24}^{1R_x} \\ L_{31}^{1R_x} & L_{32}^{1R_x} & L_{33}^{1R_x} & L_{34}^{1R_x} \\ L_{41}^{1R_x} & L_{42}^{1R_x} & L_{43}^{1R_x} & L_{44}^{1R_x} \end{bmatrix} = \begin{bmatrix} L_{11}^{1x} & L_{21}^{1x} & L_{15}^{1x} & L_{25}^{1x} \\ L_{12}^{1x} & L_{22}^{1x} & L_{16}^{1x} & L_{26}^{1x} \\ L_{13}^{1x} & L_{23}^{1x} & L_{17}^{1x} & L_{27}^{1x} \\ L_{14}^{1x} & L_{24}^{1x} & L_{18}^{1x} & L_{28}^{1x} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.65)$$

$$[L_x^{1R}] = [L_x^1][P_x] \quad (4.66)$$

In this case, the rearrangement is executed as a place exchange for columns 2 and 3 of $[L_x^1]$.

From the matrix $[L_x^{1R}]$, the vectors $\vec{L}_{sx}^{1R} = [L_{1s}^{1R_x}, L_{2s}^{1R_x}]^T$, $s = 1, 2, \dots, 8$ (see Fig. 4.18) are defined, whose components are calculated by Eqs. 4.67–4.68.

$$L_{1s}^{1R_x} = L_{1s}^{1x} \quad \text{for } s = 1, 2, 3, 4 \quad L_{2s}^{1R_x} = L_{1s}^{1x} \quad \text{for } s = 5, 6, 7, 8 \quad (4.67)$$

$$L_{1(s+4)}^{1R_x} = L_{2s}^{1x} \quad \text{for } s = 1, 2, 3, 4 \quad L_{1(s-4)}^{1R_x} = L_{2s}^{1x} \quad \text{for } s = 5, 6, 7, 8 \quad (4.68)$$

Step 4. In the second level, the direct AKLT_x2 × 2 is applied again on each of the rearranged vectors $\vec{L}_s^{1R_x}$, as shown in Eq. 4.69, where Φ_{ji}^{2x} , $\bar{L}_1^{1R_x}$ and $\bar{L}_2^{1R_x}$ are defined through the vectors $\vec{L}_s^{1R_x}$ on the basis of Eqs. 4.50–4.51.

$$\begin{bmatrix} L_{1s}^{2x} \\ L_{2s}^{2x} \end{bmatrix} = \begin{bmatrix} \Phi_{11}^{2x} & \Phi_{21}^{2x} \\ \Phi_{12}^{2x} & \Phi_{22}^{2x} \end{bmatrix} \begin{bmatrix} (L_{1s}^{1R_x} - \bar{L}_1^{1R_x}) \\ (L_{2s}^{1R_x} - \bar{L}_1^{1R_x}) \end{bmatrix} \quad \text{for } s = 1, 2, \dots, 8 \quad (4.69)$$

Step 5. The columns of such obtained matrix $[L_x^2]$ are rearranged in accordance with $[P_x]$ (Eq 4.70).

$$[L_x^{2R}] = [L_x^2][P_x] \quad (4.70)$$

In result the second rearranged matrix is obtained by Eq. 4.71.

$$[L_x^{2R}] = \begin{bmatrix} L_{11}^{2x} & L_{15}^{2x} & L_{21}^{2x} & L_{25}^{2x} \\ L_{12}^{2x} & L_{16}^{2x} & L_{22}^{2x} & L_{26}^{2x} \\ L_{13}^{2x} & L_{17}^{2x} & L_{23}^{2x} & L_{27}^{2x} \\ L_{14}^{2x} & L_{18}^{2x} & L_{24}^{2x} & L_{28}^{2x} \end{bmatrix} \quad (4.71)$$

Step 6. From the elements of the matrix $[L_x^{2R}]$, the vectors $\vec{F}_s = [F_{1s}, F_{2s}]^T$, $s = 1, 2, \dots, 8$ are defined. They are shown in Fig. 4.19, whose components are calculated by Eqs. 4.72–4.73.

$$\vec{F}_1 = [L_{11}^2, L_{12}^2]^T \quad \vec{F}_2 = [L_{15}^2, L_{16}^2]^T \quad \vec{F}_5 = [L_{13}^2, L_{14}^2]^T \quad \vec{F}_6 = [L_{17}^2, L_{18}^2]^T \quad (4.72)$$

$$\vec{F}_3 = [L_{21}^2, L_{22}^2]^T \quad \vec{F}_4 = [L_{25}^2, L_{26}^2]^T \quad \vec{F}_7 = [L_{23}^2, L_{24}^2]^T \quad \vec{F}_8 = [L_{27}^2, L_{28}^2]^T \quad (4.73)$$

The execution of steps 7–10 shown in Fig. 4.19 requires on level 1 to be applied the direct AKLT_y2 × 2 on each vertically oriented vector $\vec{F}_s = [F_{1s}, F_{2s}]^T$, $s = 1, 2, \dots, 8$ and rearrangement of the rows of the obtained matrix $[L_y^1]$, and in level 2—the same transform on the rearranged vectors $\vec{L}_{s_y}^{1R} = [L_{1s}^{1R_y}, L_{2s}^{1R_y}]^T$, $s = 1, 2, \dots, 8$ with second rearrangement of the rows of the matrix $[L_y^2]$, so that to obtain the matrix $[L]$. In this case, the matrix used for the rearrangement of the rows is $[P_y] = [P_x]^T$, and the rearrangement is executed in correspondence with Eq. 4.74.

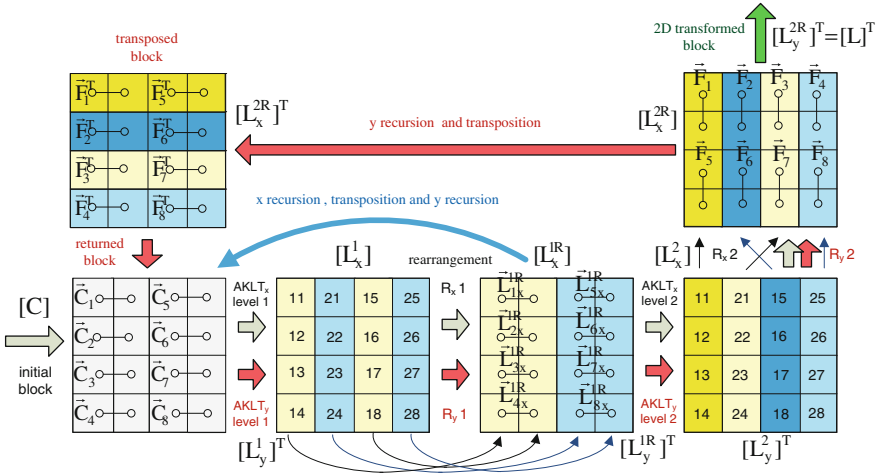


Fig. 4.20 Recursive calculation of the 2D HA-KLT for a block of size 4×4 through the processing with 1D AKLT 2×2 on the columns and rows

$$[L_y^{iR}] = [P_y][L_y^i] \quad \text{for } i = 1, 2 \quad (4.74)$$

The rearrangement is executed by exchanging the places of rows 2 and 3 of the matrix $[L_y]$.

4.9 Algorithm for Recursive 2D Hierarchical Adaptive KLT

To simplify the implementation of the 2D HA-KLT algorithm shown in Fig. 4.19, here let us suppose that it will be executed recursively in correspondence with the example, shown in Fig. 4.20. Two approaches exist for the implementation of the recursive transform, recursion in direction y only or recursion first in horizontal direction and after that—in vertical direction. In the first case (shown in Fig. 4.20 with red arrows), the matrix $[L_x^{2R}]$ is transposed after the finalized step 6. As a result, the vectors $\vec{F}_s = [F_{1s}, F_{2s}]^T$ are oriented in horizontal direction (in similar way as for $\vec{C}_s = [C_{1s}, C_{2s}]^T$). After that, on $\vec{F}_s^T = [F_{1s}, F_{2s}]$ same operations are executed as these, given in steps 2–5 for the vectors $\vec{C}_s = [C_{1s}, C_{2s}]^T$. In similar way, the recursive inverse 2D HA-KLT in direction y is executed, but with inverse direction of the calculations in each step. In the second case (recursion in both directions x and y indicated with blue arrows), it is necessary to transpose the obtained intermediate matrix between both recursive cycles for x and y .

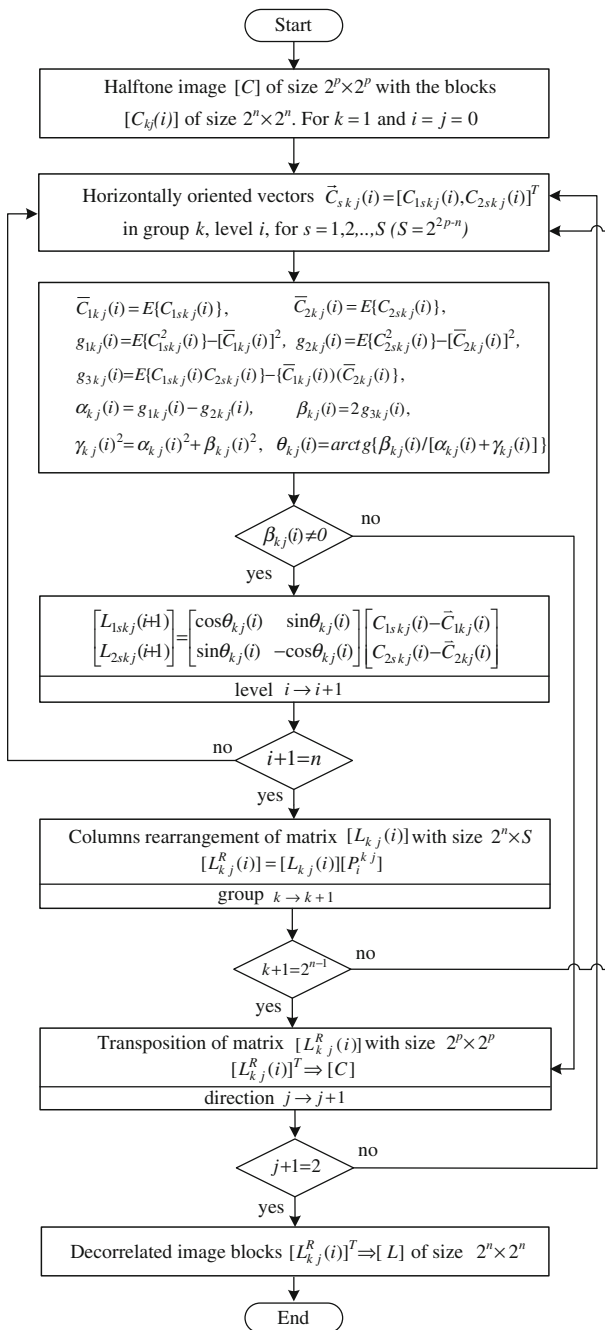


Fig. 4.21 Block diagram of the algorithm for recursive 2D HA-KLT

In the general case, the recursive direct AKLT 2×2 on the rearranged vectors $\vec{L}_{sk}^R(i) = [L_{1sk}^R(i), L_{2sk}^R(i)]^T$ in the group k obtained for the iteration i (transform level) in direction x or y is defined by Eq. 4.75, where the matrix $[\Phi_k(i)]$ in iteration i (for $i = 2, 3, \dots, n$) is defined on the basis of the vectors $\vec{L}_{sk}^R(i) = [L_{1sk}^R(i), L_{2sk}^R(i)]^T$ from the group $k = 1, 2, \dots, 2^{n-1}$.

$$\begin{bmatrix} L_{1sk}(i+1) \\ L_{2sk}(i+1) \end{bmatrix} = \begin{bmatrix} \Phi_{11}^k(i) & \Phi_{21}^k(i) \\ \Phi_{12}^k(i) & \Phi_{22}^k(i) \end{bmatrix} \begin{bmatrix} L_{1sk}^R(i) - \vec{L}_{1k}^R(i) \\ L_{2sk}^R(i) - \vec{L}_{2k}^R(i) \end{bmatrix} \quad \text{for } s = 1, 2, \dots, S \quad (4.75)$$

In the first iteration only ($i = 1$) the corresponding matrix $[\Phi_k^x(1)]$ in direction x is defined on the basis of the vectors $\vec{C}_{sk} = [C_{1sk}, C_{2sk}]^T$, and the matrix $[\Phi_k^y(1)]$ in direction y —on the basis of the vectors $\vec{F}_{sk}^t = [F_{1sk}, F_{2sk}]$. In this case, for $s = 1, 2, \dots, S$, Eq. 4.59 is changed as follows in Eqs. 4.76–4.77.

$$\begin{bmatrix} L_{1sk}^x(1) \\ L_{2sk}^x(1) \end{bmatrix} = \begin{bmatrix} \Phi_{11}^{k_x}(1) & \Phi_{21}^{k_x}(1) \\ \Phi_{12}^{k_x}(1) & \Phi_{22}^{k_x}(1) \end{bmatrix} \begin{bmatrix} (C_{1sk} - \vec{C}_{1k}) \\ (C_{2sk} - \vec{C}_{2k}) \end{bmatrix} \quad \text{— in direction } x \quad (4.76)$$

$$\begin{bmatrix} F_{1sk}^y(1) \\ F_{2sk}^y(1) \end{bmatrix} = \begin{bmatrix} \Phi_{11}^{k_y}(1) & \Phi_{21}^{k_y}(1) \\ \Phi_{12}^{k_y}(1) & \Phi_{22}^{k_y}(1) \end{bmatrix} \begin{bmatrix} (F_{1sk} - \vec{F}_{1k}) \\ (F_{2sk} - \vec{F}_{2k}) \end{bmatrix} \quad \text{— in direction } y \quad (4.77)$$

The block diagram of the algorithm for recursive 2D HA-KLT is shown in Fig. 4.21. The preferred type of the recursive calculation of the algorithm depends on the structure used for the implementation: hardware, software, or firmware.

4.10 Evaluation of the Computational Complexity of 2D HA-KLT

The computational complexity of the 2D HA-KLT algorithm could be evaluated in respect of that, needed for the 2D-KLT with a transform matrix with $N \times N$ elements, for $N = 2^n$. Both algorithms should be compared in respect of the number of operations O (additions and multiplications) needed for the calculation of the covariance matrix $[K_C]$ of size $N \times N$, its eigen values and vectors, and the elements of the transformed matrix $[L]$. Let us estimate the computational complexity of the 2D HA-KLT algorithm as following below:

1. The total number of operations needed for the calculation of all elements of the $[K_C]$ matrix is computed by Eq. 4.78.

$$O_{kov}(N) = (1/2)N(N+1)[N(N-1) + 2(N+2)] \quad (4.78)$$

In particular, for $N/2$ matrices $[K_C]$ of size 2×2 ($N=2$) one can obtain the result from Eq. 4.79.

$$O_{kov}(N) = (N/2)(3N^2 + 5N + 12) \quad (4.79)$$

2. The eigen values of $[K_C]$ of size $N \times N$ could be calculated using the QR decomposition and the Householder transform with $(N-1)$ steps [50]. In this case, the corresponding number of operations O_{val} is defined by Eq. 4.80 [48].

$$O_{val}(N) = (1/6)(N-1)(8N^2 + 17N + 42) \quad (4.80)$$

In case that for the calculation of the components Φ_{jt} of the eigen vector $\vec{\Phi}_t$ of $[K_C]$ the iterative algorithm is used [47], Eq. 4.81 will be received, where $\Phi_{jt}^{(l)}$ and $\Phi_{jt}^{(l+1)}$ are the values of the component (j, t) in the iterations (l) and $(l+1)$, k_{jp} is the elements of the matrix $[K_C]$, δ_{jp} is single operator, and L is the general number of needed operations.

$$\Phi_{jt}^{(l+1)} = \Phi_{jt}^{(l)} - (1/k_{jj}) \left[\sum_{p=1}^N (k_{jp} - \lambda_t \delta_{jp}) \Phi_{pt}^{(l)} \right] \quad (4.81)$$

for $j = 1, 2, \dots, N$ and $l = 0, 1, 2, \dots, L$

From Eq. 4.81 it follows that the number of operations needed for the calculation of the eigen vector $\vec{\Phi}_t$ is $2LN(N+1)$. Then the total number of operations needed to calculate all N eigen vectors is determined by Eq. 4.82.

$$O_{vec}(N) = N[2N(LN + L + 1) - 1] \quad (4.82)$$

In particular, for $L=4$, Eq. 4.83 is obtained.

$$O_{vec}(N) = N[2N(4N + 5) - 1] \quad (4.83)$$

To calculate the eigen vectors of all $N/2$ matrices $[K_C]$ of size 2×2 on the basis of Eqs. 4.49–4.51 are needed $O_{vec}(N) = 9N/2$ operations in total.

3. To apply the direct KLT on each N -component vector obtained for an image of size $N \times N$ are needed $N(2N-1)$ operations in total. As it was explained before, this transform should be executed N times on all rows of the image. The total number of needed operations is defined by Eq. 4.84.

$$O_{KLT}(N) = N^2(2N-1) \quad (4.84)$$

Then for the execution of the direct 2D-KLT on a block of size $N \times N$, the number of operations is calculated by Eq. 4.85.

$$O_{2DKLT}(N) = 2N^2(2N - 1) \quad (4.85)$$

To execute the direct HA-KLT on the N -component vector (for $N = 2^n$) are needed $n = \lg_2 N$ levels for the direct AKLT 2×2 on $N/2$ groups of 2-component vectors. In this case, the total number of operations is provided by Eq. 4.86 and for the 2D HA-KLT it is $O_{2DAKLT}(N) = 6 N \lg_2 N$, respectively.

$$O_{AKLT}(2) = 6n(N/2) = 3N \lg_2 N \quad (4.86)$$

4. The total number of operations $O_1(N)$ for the divisible 2D-KLT is determined by Eq. 4.87.

$$O_1(N) = (1/3)[3N(N^3 - 3) + 2N^2(37N + 42) - 2] \quad (4.87)$$

The computational complexity in this case is $\approx O_1(N^4)$.

5. The total number of operations $O_2(N)$ for the divisible AM 2D-KLT is defined by Eq. 4.88 and the corresponding computational complexity is $\approx O_2(N^3)$.

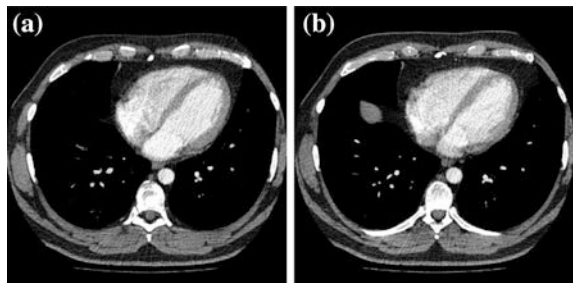
$$O_2(N) = 2[O_{kov}(N) + O_{vec}(2N) + O_{AKLT}(N)] = N(3N^2 + 5N + \lg_2 N + 21) \quad (4.88)$$

6. The reduction of the total number of operations needed for the 2D HA-KLT compared to that of the 2D-KLT could be defined by Eq. 4.89 by using the coefficient η .

$$\eta(N) = \frac{O_1(N)}{O_2(N)} = \frac{3N(N^3 - 3) + 2N^2(37N + 42) - 2}{3N(3N^2 + 5N + \lg_2 N + 21)} \quad (4.89)$$

For a transform matrix of size $N = 4, 8, 16$, $\eta(N) = 5.6; 8.9; 12.4$ is obtained. In the general case, the reduction of the computational complexity of the 2D HA-KLT algorithm compared to that of the 2D-KLT is approximately N times.

Fig. 4.22 CT images from a group of ten images: **a** the first image, **b** the tenth image



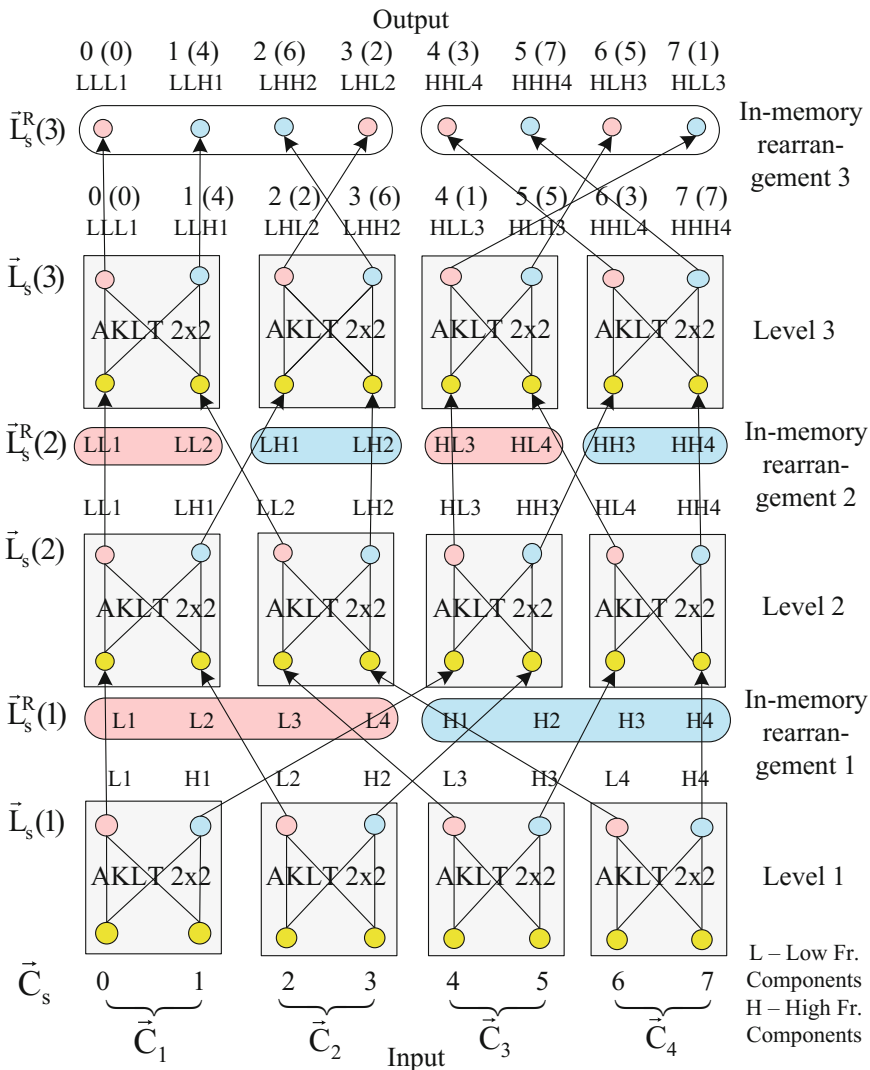


Fig. 4.23 Direct three-level 1D HA-KLT, for the vectors $\vec{C}_s = [C_{1s}, C_{2s}]^T, s = 1, 2, 3, 4$

4.11 Experimental Results for Single Images

The proposed algorithm was tested on a group of ten halftone CT images each of size 512×512 pixels, 16 bpp (depending on the medical device, in practice the images with 8, 10, 12 and 16 bpp are used). The first and the last CT image from this group are shown in Fig. 4.22a, b. There is no specific change in the proposed algorithm performance for all these types of images.

Table 4.4 The covariance ratio and the execution times for three-level 2D HA-KLT, versus the full separable 2D-KLT for blocks of size 8×8 pixels

Image	Level 0		Level 1		Level 2		2D-KLT
	<i>CovR</i>	Execution time, s	<i>CovR</i>	Execution time, s	<i>CovR</i>	Execution time, s	Execution time, s
1	0.28	0.0076	16.25	0.0142	68.07	0.0195	0.1624
2	0.17	0.0066	9.94	0.0122	40.99	0.0177	0.1558
3	0.23	0.0057	12.94	0.0126	55.67	0.0176	0.1654
4	0.31	0.0051	17.42	0.0097	74.98	0.0149	0.1471
5	0.25	0.0047	13.96	0.0095	62.31	0.0140	0.1585
6	0.27	0.0055	15.73	0.0102	66.42	0.0152	0.1529
7	0.19	0.0048	10.97	0.0095	46.55	0.0151	0.1353
8	0.29	0.0057	17.10	0.0100	69.99	0.0150	0.1553
9	0.30	0.0060	17.41	0.0137	72.79	0.0194	0.1496
10	0.22	0.0049	13.01	0.0103	54.09	0.0157	0.1350

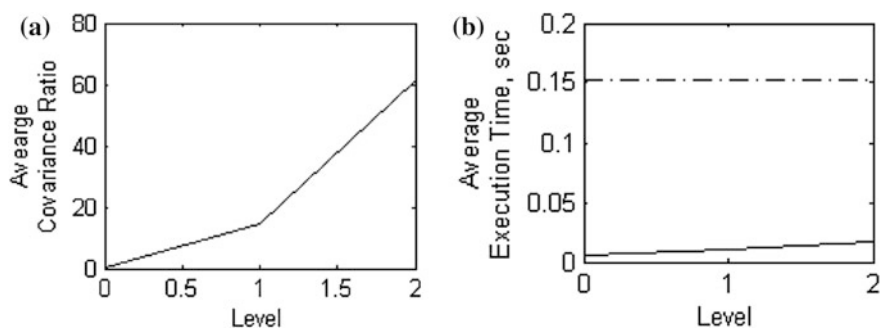


Fig. 4.24 Evaluation of the execution time for the 2D HA-KLT: **a** average covariance ratio of the HA-KLT for levels 0, 1, 2, **b** average execution time for 2D HA-KLT and 2D-KLT

For the experiments, all images were divided into blocks of size 8×8 pixels, and the eight vectors corresponding to a row of pixels in the block of 8 components were calculated. All eight vectors (block by block) were stored in a matrix of 32,768 rows and 8 columns going through the blocks in a progressive scan to the end of the image. Then over the resulting matrix the direct 3-level 2D HA-KLT was applied as shown in Fig. 4.23. Here, with L and H are marked the first (low-frequency) components with high power and the second (high frequency) components with low power obtained by the calculation of the AKLT 2×2 , respectively.

For comparison, the classic 2D-KLT was also performed. The execution times were calculated for each CT image, being processed by both algorithms. The test environment was IBM[®] PC[®] compatible computer with P4 processor, running at

2 GHz with 2 GB of RAM under MS[®] Windows[®] XP SP3 (32 bit) and Matlab R2009A. One more aspect of the experiments was to estimate the decorrelation achieved for the proposed algorithm in each level of the 2D HA-KLT. For this, a new parameter *CovR* was introduced, called Covariance Ratio, which actually represents the ratio between the sum of the squared values of all diagonal elements in the covariance matrix and the sum of the squared values of the non-diagonal elements (Eq 4.90).

$$CovR = \frac{\sum_{i=1}^8 \sum_{j=1}^8 k_{ij(i=j)}^2}{\sum_{i=1}^8 \sum_{j=1}^8 k_{ij(i \neq j)}^2} \quad (4.90)$$

For the classic 2D-KLT algorithm this ratio is obviously infinity for any image. The experimental results in Table 4.4 are given for all 10 test images. The graphic representation of the numerical results from Table 4.4 is shown in Fig. 4.24.

There is a significant difference among the decorrelation degrees from the level 0 up to the level 2 by a factor of a hundred units. It is obvious that, if higher decorrelation is needed and thus—a higher compression, then all three levels should be used. In contrast to the general 2D-KLT, the achieved execution time is lower by a factor of 10, which is in consent with the theoretically derived reduction of the computational complexity by 8.9 times. Some deviation from the predicted values and the experimental results could be noticed due to the rearrangement of the vector components in each level, which were not included in the expressions.

The proposed new algorithm for 2D transform of halftone images has the following basic advantages:

1. The significant decorrelation of the elements of the transformed image blocks of size $2^n \times 2^n$ is achieved.
2. The computational complexity of the 2D HA-KLT for an image block of size $2^n \times 2^n$ compared to that of the conventional 2D-KLT is reduced $\approx 2^n$ times.
3. The 2D HA-KLT algorithm is executed in n consecutive levels, which permits parallel and recursive calculation. This makes the offered algorithm quite suitable for hardware implementation and opens the possibility for real-time processing of video sequences.

The 2D HA-KLT algorithm could be also generalized for inter-block image decorrelation. For this, it is necessary to arrange the elements of same position in each block transformed with 2D HA-KLT in corresponding 2D matrices. After their division into smaller blocks, each could be processed with the 2D HA-KLT again, etc. In result of this recursive processing, it is possible to achieve a decorrelation of the image elements. The block size should cover the maximum correlation range for the image pixels in horizontal and vertical directions. This algorithm could be also used for the achievement of spatial decorrelation of color images. In this case, it is executed separately for each color component matrix.

4.12 Conclusion

In this research, one general approach for decorrelation of groups of images and of the blocks of single images, which are processed by using the new hierarchical transform based on the AKLT, is presented. The new algorithms for processing of single images and groups of related images through HA-KLT could be combined as follows: each group of images is decorrelated through inter-image HA-KLT and then on each of the retained eigen images is applied the intra-frame HA-KLT. As a result, the compression efficiency for the processed group of images is significantly enhanced. One more important application of the HA-KLT is to achieve a reduction of the features space through combining different kernels for nonlinear transform without increasing the pattern recognition error.

References

1. Fieguth P (2011) *Statistical image processing and multidimensional modeling*. Springer, New York, Dordrecht, Heidelberg, London
2. Reed T (2004) *Digital image sequence processing, compression, and analysis*. CRC Press, Boca Raton, London, New York, Washington, DC
3. Shi Y, Sun H (2008) *Image and video compression for multimedia engineering: fundamentals, algorithms, and standards*, 2nd edn. CRC Press, Taylor&Francis Group, LLC, Boca Raton, London, New York, Washington, DC
4. Mukhopadhyay J (2011) *Image and video processing in the compressed domain*. Chapman & Hall/CRC, Taylor & Francis Group, Boca Raton
5. Rao K, Kim D, Hwang J (2010) *Fast Fourier transforms: algorithms and applications*. Springer, Dordrecht, Heidelberg, London, New York
6. Richards J, Jia X (2013) *Remote sensing digital image analysis—an introduction*, 5th edn. Springer, Berlin, Heidelberg
7. Hyvarinen A, Hurri J, Hoyer P (2009) *Natural image statistics, a probabilistic approach to early computational vision*. Springer, Dordrecht, Heidelberg, London, New York
8. Jolliffe I (2002) *Principal component analysis*, 2nd edn. Springer, New York
9. Dony R (2001) Karhunen-Loeve transform. In: Rao K, Yip P (eds) *The transform and data compression handbook*. CRC Press, Boca Raton, London, New York, Washington, DC
10. Fleury M, Downton A, Clark A (1997) Karhunen-Loeve transform—image processing. University of Essex, Wivenhoe Park, UK, pp 172–201
11. Orfanidis S (2007) SVD, PCA, KLT, CCA, and all that. Rutgers University Electrical and Computer Engineering Department, *Optimum Signal Processing*, pp 1–77
12. Miranda A, Borgne Y, Bontempi G (2007) New routes from minimal approximation error to principal components. Kluwer Academic Publishers, pp 1–14
13. Landqvist R, Mohammed A (2006) Comparative performance analysis of three algorithms for principal component analysis. *Radioengineering* 15(4):84–90
14. Bishop C (1998) Bayesian principal component analysis. *Adv Neural Inf Process Syst* 11:382–388
15. Tipping M, Bishop C (1999) Probabilistic Principal Component Analysis. 61. *J Roy Stat Soc Ser B Part 3*:611–622
16. Liwicki S, Tzimiropoulos G, Zafeiriou S, Pantic M (2013) Euler principal component analysis. *Int J Comput Vis* 101(3):498–518

17. Patil U, Mudengudi U (2011) Image fusion using hierarchical PCA. In: International conference on image information processing (ICIIP'2011), Himachal Pradesh, pp 1–6
18. Abadpour A, Kasaei S (2008) Color PCA eigen images and their application to compression and watermarking. *J Image Vis Comput* 26(7):878–890
19. Chadha A, Satam N, Sood R, Bade D (2013) Image steganography using Karhunen-Loève transform and least bit substitution. *Int J Comput Appl* 79(9):31–37
20. Naik G, Kumar D (2011) An overview of independent component analysis and its applications. *Informatica* 35:63–81
21. Strang G (2006) Linear algebra and its applications, 4rd edn. Thomson Brooks/Cole
22. Golub G, Van Loan C (1996) Matrix computations, 3rd edn. John Hopkins University Press, Baltimore, Maryland
23. Householder A (1975) The theory of matrices in numerical analysis. Dover Publications, New York
24. Press W, Teukolsky S, Vetterling W (2001) Numerical recipes in C. The art of scientific computing, 2nd edn. Cambridge University Press, New York
25. Carlen E (2003) The symmetric eigenvalue problem. In: *Calculus++*, Georgia Tech, Atlanta
26. Diamantaras K, Kung S (1996) Principal component neural networks. Wiley, New York
27. Levy A, Lindenbaum M (2000) Sequential Karhunen-Loeve basis extraction and its application to images. *IEEE Trans Image Process* 9(8):1371–1374
28. Li W, Yue H, Cervantes S, Qin S (2000) Recursive PCA for adaptive process monitoring. *J Process Control* 10:471–486
29. Erdogmus D, Rao Y, Peddaneni H, Hegde A, Principe J (2004) Recursive PCA using eigenvector matrix perturbation. *EURASIP J Adv Sig Process* 13:2034–2041
30. Hanafi M, Kohler A, Qannari E (2010) Shedding new light on hierarchical principal component analysis. *J Chemometr* 24(11–12):703–709
31. Amar A, Leshem A, Gastpar M (2010) Recursive implementation of the distributed Karhunen-Loève transform. *IEEE Trans Sig Process* 58(10):5320–5330
32. Thirumalai V (2012) Distributed compressed representation of correlated image sets. Thesis No 5264, Lausanne, EPFL
33. Saghri J, Schroeder S, Tescher A (2009) An adaptive two-stage KLT scheme for spectral decorrelation in hyperspectral bandwidth compression. *SPIE Proc* 7443:72–84
34. Wongsawat Y, Orintara S, Rao K (2006) Integer Sub-Optimal Karhunen Loeve transform for multi-channel lossless EEG compression. In: 14th European signal processing conference (EUSIPCO'2006), Florence, Italy, pp 4–8
35. Blanes I, Sagristà J (2010) Cost and scalability improvements to the Karhunen-Loève transform for remote-sensing image coding. *IEEE Trans Geosci Remote Sens* 48(7):2854–2863
36. Blanes I, Sagristà J, Marcellin M, Rapesta J (2012) Divide-and-conquer strategies for hyperspectral image processing. *IEEE Sig Process Mag* 29(3):71–81
37. Blanes I, Sagristà J (2011) Pairwise orthogonal transform for spectral image coding. *IEEE Trans Geosci Remote Sens* 49(3):961–972
38. Ma Z (2013) Sparse principal component analysis and iterative thresholding. *Ann Stat* 41(2):772–801
39. Jain A (1976) A fast Karhunen-Loeve transform for a class of random processes. *IEEE Trans Commun COM-24*:1023–1029
40. Rokhlin V, Szlam A, Tytgert M (2009) A randomized algorithm for principal component analysis. *SIAM J Matrix Anal Appl* 31(3):1100–1124
41. Waldemar P, Ramstad T (1997) Hybrid KLT-SVD image compression. In: IEEE international conference on acoustics, speech, and signal processing. IEEE Computer Society Press, Los Alamitos, pp 2713–2716
42. Kambhatla N, Haykin S, Dony R (1998) Image compression using KLT, wavelets and an adaptive mixture of principal components model. *J VLSI Sign Proc* 18(3):287–296
43. Bitz I, Barret M, Pham D (2010) On optimal transforms in lossy compression of multicomponent images with JPEG2000. *Signal Process* 90(3):759–773

44. Kountchev R, Kountcheva R (2013) PCA-based adaptive hierarchical transform for correlated image groups. In: IEEE international conference on telecommunications in modern satellite, cable and broadcasting services (TELSIKS'2013), Serbia, pp 323–332
45. Kountchev R (2013) Applications of the hierarchical adaptive PCA for processing of medical CT images. *Egyptian Comput Sci J* 37(3):1–25
46. Kountchev R, Nakamatsu K (2012) One approach for grayscale image decorrelation with adaptive multi-level 2D KLT. In: Graña M, Toro C, Posada J, Howlett R, Jain L (eds) *Advances in knowledge-based and intelligent information and engineering systems*. IOS Press, Netherlands
47. Korn G, Korn T (1968) *Mathematical handbook for scientists and engineers*. Mc Graw-Hill Book Company, New York
48. Kountchev R, Kountcheva R (2013) Decorrelation of multispectral images, based on hierarchical adaptive PCA. *Int J WSEAS Trans Sign Proc* 3(9):120–137
49. Arora S, Barak B (2009) *Computational complexity*. Cambridge University Press, New York
50. Faires J, Burden R (2012) *Numerical methods*, 4th edn. Cengage Learning, Brooks/Cole, Boston, MA
51. Simoncelli E, Olshausen B (2001) Natural image statistics and neural representation. *Ann Rev Neurosci* 24:1193–1216
52. Kountchev R, Kountcheva R (2012) Compression of CT Images with branched inverse pyramidal decomposition. In: 12th WSEAS international conference on signal processing, computational geometry and artificial vision (ISCGAV'2012), Turkey, pp 74–79
53. Hao P, Shi Q (2003) Reversible integer KLT for progressive-to-lossless compression of multiple component images. In: IEEE international conference on image processing ICIP'2003, vol 1. Barcelona, Spain, pp 633–636
54. Liu Y, Bouganis C, Cheung P, Leong P, Motley S (2006) Hardware efficient architectures for eigenvalue computation. In: *Proceedings of design, automation and test in Europe (DATE'2006)*, pp 953–958

Chapter 5

Automatic Estimation for Parameters of Image Projective Transforms Based on Object-Invariant Cores

Vadim Lutsiv

Abstract A lot of efforts were addressed to developing the affine-invariant image descriptions. At the same time, development of projective-invariant sets of features still remains the open problem. An opposite approach to automatic recognition of images with geometric transforms is proposed in this chapter. The images are transformed to object-invariant form, in which all generic-specific features are suppressed while the parameters of geometric transform are still preserved. Then the parameters of geometric transform of image are estimated by means of comparison of its object-invariant description with a template form common for all classes of objects. The estimated geometric transforms can be then compensated, and the image can be recognized by any pattern recognition techniques. The presented theoretical development is also proven by computer simulation, and the reached theoretical results are compared with the ones reached by other authors. Several examples of practical application of developed theory are also presented.

Keywords Image recognition · Affine transform · Projective transform · Transformation parameters · Object-invariant core

5.1 Introduction

The geometric transforms of images are well-known problem in computer vision. The Affine Transform (AT) and Projective Transform (PT) can result from image projection through the spherical (or almost spherical) lenses being the traditional

V. Lutsiv (✉)

Vavilov State Optical Institute, 5-2 Kadetskaya line, St. Petersburg 199034
Russian Federation
e-mail: vluciv@mail.ru

V. Lutsiv

National Research University of Information Technologies, Mechanics and Optics,
49 Kronversky av., St. Petersburg 197101, Russian Federation

parts of photo- and video-cameras. A lot of efforts were addressed to developing the affine-invariant image descriptions. Very simple and weak solutions based on invariance to rotations and scaling (see, e.g. [1, 2]) were initially proposed in the seventieths of the last century. The methods based on translation invariance of Fourier spectrum and peculiarities of polar coordinate system [3–5] (the Fourier-Mellin transform among them) turned out to be much more sensitive and powerful, as well as the methods based on invariant moments [6] enabling an invariance to transformations, belonging to the similarity group. The invariant descriptions of object borders by chain codes [7] and signatures [8] also should be mentioned. The challenging solutions (enabling invariance to translations, rotations, and scaling) were reached with the use of histograms of inner distances [9] and histograms of directions of gradients being the basic tool of the famous methods Scale-Invariant Feature Transform (SIFT) [10], Affine-SIFT (ASIFT) [11], and Speeded-Up Robust Feature (SURF) [12]. However, actually, the invariance to translations and scaling was reached there by searching for key-points through different positions and scales. It is interesting to mention also the paper [13], in which the advantages of SIFT and SURF key-points were combined with the rotation and scale invariance reached by Fourier-Mellin transform. The most successful modern algorithm of Viola and Jones [14] and matching methods based on histograms of oriented gradients [15, 16] also search for solutions through different positions and scales. The powerful algorithm of discriminatively trained part based models [16] also applies latently a type of structural analysis by means of decomposition of image into several structural components. The other structural methods (see, e.g. in [17–19]) estimate the parameters of affine or projective transforms by means of joint analysis of spatial positions of detected contour structural elements, and then the estimations of transforms are used in image matching hypotheses.

However, an opposite approach can be applied for automatic recognition of images with geometric transforms. The images may be transformed to an object-invariant form, in which all generic-specific features are suppressed while the parameters of geometric transform are still preserved. Then, the parameters of geometric transform can be estimated by means of comparison of object-invariant description with a template form common for all classes of objects. Thus, the geometric transform of image can be estimated and compensated, and then the image can be recognized by any pattern recognition techniques.

The author of this chapter succeeded in designing of object-invariant cores that correspond to images of all types of spatially compact (having restricted definitional domain) objects separated from background [20, 21]. The object-invariant core is synthesized by means of truncating the high-frequency harmonic components of spatial spectrum of image. These rejected high-frequency components present the object peculiarities, while the rest (extremely low-frequency) components still contain the information about image spatial transforms. It is shown [20, 21] that such object-invariant core is mathematically described by elliptic paraboloid (quadratic parabola in the 1D image projections). It is also shown in [20, 21] that the spatial spectra of mentioned above object-invariant cores (elliptic paraboloids, 1D parabolas, and harmonic functions) are orthogonal to the rest part of image

spectrum. Thus, if any image is convolved (or subjected to operation of correlation) with the templates of mentioned above narrow-band shapes (paraboloids, parabolas, harmonic functions), such templates will interact only with the corresponding to them object-invariant cores, contained in image. Thus, the parameters of image geometric transform can be estimated from such correlation functions as it is described in detail in the sections below.

For estimation of different parameters of AT, it is presented as a sequence of simpler transformations in Sect. 5.2. It is shown in Sects. 5.3–5.5, how all parameters of AT (except the rotation and specular reflection) can be determined analytically from the properties of the proposed object-invariant core. The parameters of rotation and specular reflection are analytically calculated from the cyclic narrow-band harmonic cores of image projection on the angular coordinate of polar coordinate system as it is presented in Sect. 5.6.

While the 6-parametric AT is entirely linear, the full PT contains additionally a nonlinear part described by two additional parameters. Due to this nonlinearity, the specific parameters of PT can not be determined analytically. A novel iterative optimization procedure is proposed in Sect. 5.7 for estimating all parameters of PT. The idea of this procedure is borrowed from the living vision systems, in which the surface inclination is detected from recession of texture elements (texels) [22]. It is shown in Sect. 5.7, how the missing parameters of PT can be estimated from displacement of object-invariant core of image under a test projective transform. The convergence of such iterative estimation procedure is rigorously proven.

The software simulation results proving the correctness of proposed theoretical development are presented in Sect. 5.8 showing also several examples of practical application of developed theory. The similar theoretic results reached independently by other authors are discussed in Sect. 5.9. The materials of this chapter are finally summarized in Sect. 5.10 of Conclusion.

5.2 Projective Transform as a Sequence of Simpler Geometric Transformations

The plain PT is described by the following formulae (Eq. 5.1) linking the pixel coordinates $\vec{X} = (x, y)^T$ in initial image with coordinates $\vec{X}' = (x', y')^T$ in transformed image, where a_1, \dots, a_6 are the parameters of AT that is a partial case of PT corresponding to zero parameters a_7 and a_8 .

$$\begin{aligned} x' &= (a_1x + a_2y + a_5)/(a_7x + a_8y + 1) \\ y' &= (a_3x + a_4y + a_6)/(a_7x + a_8y + 1) \end{aligned} \quad (5.1)$$

Let us present AT in a form more convenient for analysis.

Theorem 1 Any plain AT can be presented as the following sequence of simple plain geometric transformations:

- An image rotation by angle φ with respect to coordinate origin (Eq. 5.2):

$$x' = x\cos(\varphi) - y\sin(\varphi) \quad y' = x\sin(\varphi) + y\cos(\varphi). \quad (5.2)$$

- A spatially isotropic M -fold scaling of image (Eq. 5.3):

$$x' = Mx \quad y' = My. \quad (5.3)$$

- A spatially anisotropic scaling of image (μ -fold expanding or shrinking of image in the direction Θ and optional specular reflection corresponding to $\mu < 0$) (Eq. 5.4):

$$\begin{aligned} x' &= \mu [x \cos(-\Theta) - y \sin(-\Theta)] \cos(\Theta) - [x \sin(-\Theta) + y \cos(-\Theta)] \sin(\Theta) \\ y' &= [x \sin(-\Theta) + y \cos(-\Theta)] \cos(\Theta) + \mu [x \cos(-\Theta) - y \sin(-\Theta)] \sin(\Theta). \end{aligned} \quad (5.4)$$

- The translations along two coordinate axes (Eq. 5.5):

$$x' = x + a_5 \quad y' = y + a_6. \quad (5.5)$$

The proof of this theorem is not difficult, and the author presented it in [20].

It can be easily seen that the image rotation (Eq. 5.2) and the isotropic scaling (Eq. 5.3) do not change the position of image centroid (center of mass), if it has the zero coordinates. The anisotropic scaling of image along the abscissa axis (it corresponds to $\Theta = 0$ in Eq. 5.4) or along the ordinate axis also do not change the centroid coordinates, if they are zero. Thus, the anisotropic scaling (Eq. 5.4) composed of rotations and scaling along the abscissa axis also does not displace image centroid situated at coordinate origin.

Let us present the AT in a matrix form \mathbf{A} as a sequence of described above simple geometric transforms (Eq. 5.6), where a_1, \dots, a_6 are the parameters of AT, the matrices $\mathbf{A}_1, \mathbf{A}_2$ correspond to translations of image centroid along the abscissa and ordinate axes, the matrix \mathbf{A}_3 corresponds to anisotropic scaling, the matrix \mathbf{A}_4 describes the specular reflection with respect to a straight line passing through coordinate origin, the matrix \mathbf{A}_5 corresponds to isotropic scaling, the matrix \mathbf{A}_6 describes image rotation with respect to coordinate origin.

$$\mathbf{A} = \begin{pmatrix} a_1 & a_2 & a_5 \\ a_3 & a_4 & a_6 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 \mathbf{A}_4 \mathbf{A}_5 \mathbf{A}_6 \quad (5.6)$$

The sequential order of simple transforms presented in Eq. 5.6 can be changed. However, this is just this order that enables step by step determining the parameters

of these transforms and compensating them beginning from \mathbf{A}_1 and finishing by \mathbf{A}_6 because in the case of such compensation order, each next operation of compensation does not disturb the results of previous operations of compensation in this sequence. The transforms \mathbf{A}_1 and \mathbf{A}_2 are mutually orthogonal (i.e. compensation of centroid displacement along the ordinate axis does not disturb the compensation of displacement along the abscissa axis, and vice versa). Thus, these two transforms may be swapped in Eq. 5.6. As it was shown above, the transforms \mathbf{A}_3 – \mathbf{A}_6 do not displace image centroid from coordinate origin. Thus, the zero abscissa and ordinate of centroid are worth to be considered as its template position, and compensation of centroid displacement from coordinate origin should be the first step of compensation of AT. Similarly, the transforms \mathbf{A}_5 and \mathbf{A}_6 of isotropic scaling and rotation may be swapped. The transform \mathbf{A}_3 of anisotropic scaling changes both the radial and azimuthal polar coordinates of points. As it can be seen from Eq. 5.4, it changes the ratios of radial coordinates of the points having differing azimuthal coordinates. At the same time, the transforms \mathbf{A}_5 and \mathbf{A}_6 do not change the ratios of radial coordinates of points, thus the compensation of \mathbf{A}_3 should precede the compensations of \mathbf{A}_5 and \mathbf{A}_6 . The transform \mathbf{A}_4 of specular reflection may be swapped with isotropic scaling \mathbf{A}_5 . The transform \mathbf{A}_4 may be determined and compensated simultaneously with the compensation of transform \mathbf{A}_3 (in fact, transform \mathbf{A}_4 is a part of transform \mathbf{A}_3) as well as with the compensation of rotation \mathbf{A}_6 .

5.3 Determining the Parameters of Image Translation

As it was mentioned in Sect. 5.1, the parameters of geometric transforms can be estimated by inspecting the properties of object-invariant image core proposed in [20, 21]. For determining the image translations, let us prove the following lemma.

Lemma 1 *Let us consider a correlation function of image with template elliptic paraboloid, the correlation function is calculated by the parameter of their mutual shift. If the correlation function reaches its extremum, then the coordinates of image centroid coincide with the vertex coordinates of elliptic paraboloid at the point of extremum.*

Proof Let the longer axis of elliptic section of elliptic paraboloid be parallel to abscissa axis. This assumption does not restrict the generality of task because the cases of any arbitrary orientation of elliptic paraboloid can be boiled down to this orientation by simple rotation of coordinate system. According to the proposed assumption, let us describe the paraboloid by Eq. 5.7, where a and b are the paraboloid vertex coordinates along the abscissa and ordinate axes, respectively, C is an arbitrary constant, M is a ratio of lengths of axes of elliptic section of paraboloid.

$$F(x, y) = [M(x - a)]^2 + (y - b)^2 + C \quad (5.7)$$

□

Let $T(x, y)$ be a brightness of pixel having coordinates $\{x, y\}$. The correlation function of brightness $T(x, y)$ and paraboloid (Eq. 5.7) by the parameter $\tau = -a$ of shift along the abscissa axis is monotone convex (concave) [20], thus it may have a single extremum. This correlation function is represented by Eq. 5.8, where Ω is the definitional domain of function $T(x, y)$ describing the brightness of image points.

$$f(\tau) = \iint_{\Omega} T(x, y)F(x + \tau, y)dxdy \quad (5.8)$$

Here and further, the complex conjugation will be skipped in operation of correlation because the both operands of correlation have real values. Then, taking in account Eq. 5.7, the extremum of Eq. 5.8 can be found from the following expression:

$$\frac{\partial}{\partial \tau} f(\tau) = \iint_{\Omega} T(x, y) \frac{\partial}{\partial \tau} F(x + \tau, y)dxdy = \iint_{\Omega} T(x, y)2M^2(x - a)dxdy = 0.$$

Therefore, keeping in mind the well known formulae (Eq. 5.9) for coordinates x_c and y_c of centroid, it can be concluded that the image centroid has abscissa $x = a$, i.e. the abscissa of centroid coincides with the abscissa of vertex of paraboloid, if the correlation function (Eq. 5.8) reached its extremum.

$$x_c = \frac{\iint_{\Omega} x \cdot T(x, y)dxdy}{\iint_{\Omega} T(x, y)dxdy} \quad y_c = \frac{\iint_{\Omega} y \cdot T(x, y)dxdy}{\iint_{\Omega} T(x, y)dxdy} \quad (5.9)$$

In a similar way, by differentiation of correlation function by the parameter of shift of paraboloid (Eq. 5.7) along the ordinate axis, it could be shown that the ordinate of centroid coincides with the ordinate of vertex of paraboloid, if the correlation function reached its extremum. Thus, the Lemma 1 is proven.

The object-invariant core (Eq. 5.7) used as one of operands of correlation function can be referred to as a weighting function of centroid detector or simply as a centroid detector, and in the task of detection it is worth to set $M = 1$.

As it was mentioned above, the object-invariant parabolic core of image has a spectral range entirely separated from the rest spatial spectrum of image, and according to decomposition (Eq. 5.6), the affine transforms of image cause the rotations, an isotropic scaling, an anisotropic scaling of image spectrum, and the linear changes in the phases of spectral components. This is also true for the spectral components corresponding to the object-invariant parabolic core of image. Thus, the object-invariant image core is also subjected to AT. Let us consider the paraboloid of revolution having zero vertex coordinates as object-invariant core of image in its template state. Then, any AT applied to image will result in expanding (shrinking) of the core in respective direction (the core will become the elliptic

paraboloid) and in displacement of its vertex. The paraboloid of revolution will remain the paraboloid of revolution under any isotropic scaling, rotation, and specular reflection. Thus, it can be easily seen that the direction of axis of elliptic section of elliptic paraboloid contained in image will correspond to the parameter Θ of direction of anisotropic scaling of image while the ratio of lengths of axes of elliptic section will correspond to the coefficient μ of anisotropic scaling as it is shown in Eq. 5.4. It will be shown in the next section that if an image is transformed into such state that the elliptic paraboloid contained in it became the paraboloid of revolution with zero vertex coordinates, then such transformation really compensates the transforms \mathbf{A}_1 , \mathbf{A}_2 , and \mathbf{A}_3 of image translations and anisotropic scaling. It will enable the analytical derivation of formulae for determining the parameters of anisotropic scaling. Thus, the elliptic paraboloid (Eq. 5.7) (the object-invariant core of image) remains the elliptic paraboloid under any AT while the parameters of transformed core bear the information on the parameters of AT, to which an image was subjected.

5.4 Determining the Direction and Value of Anisotropic Scaling of Image

The next simple transform to be estimated and compensated according to Eq. 5.6 is the anisotropic image scaling \mathbf{A}_3 . This transform has two parameters: the direction Θ and ratio μ of scaling. In the plain polar coordinate system, this transform can be described by Eqs. 5.10–5.11, where the polar coordinates r_1 and φ_1 of points of transformed image are expressed in the terms of coordinates r and φ of initial image.

$$r_1^2 = r^2 [\mu^2 \cos^2(\varphi - \Theta) + \sin^2(\varphi - \Theta)] \quad (5.10)$$

$$\varphi_1 = \arccos \left[\mu \cos(\varphi - \Theta) / \sqrt{\mu^2 \cos^2(\varphi - \Theta) + \sin^2(\varphi - \Theta)} \right] + \Theta \quad (5.11)$$

Let the paraboloid of revolution with zero vertex coordinates be the object-invariant core of image in its template state. Then, as it was discussed in the previous section, the paraboloid of revolution becomes an elliptic paraboloid as a result of transform \mathbf{A}_3 . According to the technique applied in Sect. 5.3, let us determine the parameters Θ and μ of anisotropic scaling with respect to template state of image on the base of analysis of correlation function of image with a weighting function of the shape of elliptic paraboloid. Such weighting function presented in a plain polar coordinate system with the coordinates r and φ will be described by Eq. 5.12, where ψ is the direction of one of axes of elliptic section of paraboloid, M is the elongation ratio of this section, and C_1 is an arbitrary constant that will be further skipped.

$$F(r, \varphi, \psi, M) = [Mr \cos(\varphi - \psi)]^2 + [r \sin(\varphi - \psi)]^2 + C_1 \quad (5.12)$$

The operands of correlation function (the image under analysis and the weighting function of the form of elliptic paraboloid) should be described in a common coordinate system, thus the following mathematical manipulations will be accomplished. The Eq. 5.12 describes a paraboloid of revolution M -fold expanded (or shrunk) in the direction ψ . Thus, when the correlation function is calculated, the brightness of image points may be “weighted” not by the values of template weighting function (Eq. 5.12) but by the values of paraboloid of revolution, the points coordinates of which started corresponding (as a result of mentioned above shrinking) to the points coordinates of transformed (distorted) image under analysis. In such case, the squared radius r_2^2 of the points of paraboloid of revolution will be expressed in terms of coordinates of transformed image $T(r_1, \varphi_1)$ by the following formula:

$$r_2^2 = r_1^2 [M^2 \cos^2(\varphi_1 - \psi) + \sin^2(\varphi_1 - \psi)].$$

The last formula can be transformed to Eq. 5.13 keeping in mind Eqs. 5.10 and 5.11:

$$\begin{aligned} r_2^2 &= r^2 [\mu^2 \cos^2(\varphi - \Theta) + \sin^2(\varphi - \Theta)] [M^2 \cos^2(\varphi_1 - \psi) + \sin^2(\varphi_1 - \psi)] \\ &= r^2 [\mu^2 \cos^2(\varphi - \Theta) + \sin^2(\varphi - \Theta)] \left\{ M^2 [\cos(\Theta - \psi) \frac{\mu \cos(\varphi - \Theta)}{\sqrt{\mu^2 \cos^2(\varphi - \Theta) + \sin^2(\varphi - \Theta)}} \right. \\ &\quad \left. - \sin(\Theta - \psi) \frac{\sin(\varphi - \Theta)}{\sqrt{\mu^2 \cos^2(\varphi - \Theta) + \sin^2(\varphi - \Theta)}} \right]^2 \\ &\quad + [\sin(\Theta - \psi) \frac{\mu \cos(\varphi - \Theta)}{\sqrt{\mu^2 \cos^2(\varphi - \Theta) + \sin^2(\varphi - \Theta)}} \\ &\quad + \cos(\Theta - \psi) \frac{\sin(\varphi - \Theta)}{\sqrt{\mu^2 \cos^2(\varphi - \Theta) + \sin^2(\varphi - \Theta)}}]^2 \left. \right\} \\ &= r^2 [M^2 \mu^2 \cos^2(\psi - \Theta) \cos^2(\varphi - \Theta) + M^2 \sin^2(\psi - \Theta) \sin^2(\varphi - \Theta) \\ &\quad + 2M^2 \mu \cos(\psi - \Theta) \sin(\psi - \Theta) \cos(\varphi - \Theta) \sin(\varphi - \Theta) + \sin^2(\psi - \Theta) \mu^2 \cos^2(\varphi - \Theta) \\ &\quad + \cos^2(\psi - \Theta) \sin^2(\varphi - \Theta) - 2 \sin(\psi - \Theta) \mu \cos(\varphi - \Theta) \sin(\varphi - \Theta) \cos(\psi - \Theta)]. \end{aligned} \quad (5.13)$$

The Eq. 5.13 can now be transformed to Eq. 5.14:

$$\begin{aligned}
r_2^2 &= r^2 \left\{ \frac{M^2}{4} \mu^2 [1 + \cos(2\psi - 2\Theta)] [1 + \cos(2\varphi - 2\Theta)] + \frac{M^2}{4} [1 - \cos(2\psi - 2\Theta)] \right. \\
&\quad \times [1 - \cos(2\varphi - 2\Theta)] + \frac{M^2}{2} \mu \sin(2\psi - 2\Theta) \sin(2\varphi - 2\Theta) + \frac{\mu^2}{4} [1 - \cos(2\psi - 2\Theta)] \\
&\quad \times [1 + \cos(2\varphi - 2\Theta)] + \frac{1}{4} [1 + \cos(2\psi - 2\Theta)] [1 - \cos(2\varphi - 2\Theta)] \\
&\quad \left. - \frac{\mu}{2} \sin(2\psi - 2\Theta) \sin(2\varphi - 2\Theta) \right\} \\
&= \frac{r^2}{4} \{ M^2 \mu^2 (1 + \cos[2(\psi - \Theta)] + \cos[2(\varphi - \Theta)] + \cos[2(\psi - \Theta)] \cos[2(\varphi - \Theta)]) \\
&\quad + M^2 (1 - \cos[2(\psi - \Theta)] - \cos[2(\varphi - \Theta)] + \cos[2(\psi - \Theta)] \cos[2(\varphi - \Theta)]) \\
&\quad + 2M^2 \mu \sin[2(\psi - \Theta)] \sin[2(\varphi - \Theta)] + \mu^2 (1 - \cos[2(\psi - \Theta)] \\
&\quad + \cos[2(\varphi - \Theta)] - \cos[2(\varphi - \Theta)] \cos[2(\psi - \Theta)]) + (1 + \cos[2(\psi - \Theta)] - \cos[2(\varphi - \Theta)] \\
&\quad - \cos[2(\psi - \Theta)] \cos[2(\varphi - \Theta)]) - 2\mu \sin[2(\psi - \Theta)] \sin[2(\varphi - \Theta)] \}. \tag{5.14}
\end{aligned}$$

The following replacements were applied for passing from Eqs. 5.13 to 5.14:

$$\begin{aligned}
2\sin(\alpha)\cos(\alpha) &= \sin(2\alpha) \\
\cos^2(\alpha) &= [2\cos^2(\alpha) - 1] / 2 + 1/2 = [\cos(2\alpha) + 1] / 2 \\
\sin^2(\alpha) &= 1 - \cos^2(\alpha) = [1 - \cos(2\alpha)] / 2.
\end{aligned}$$

Let us substitute into Eq. 5.14 the following formulae:

$$\begin{aligned}
\sin(\alpha)\sin(\beta) &= [\cos(\alpha - \beta) - \cos(\alpha + \beta)] / 2 \\
\cos(\alpha)\cos(\beta) &= [\cos(\alpha - \beta) + \cos(\alpha + \beta)] / 2.
\end{aligned}$$

Then, Eq. 5.14 is transformed to Eq. 5.15:

$$\begin{aligned}
r_2^2 &= \frac{r^2}{4} \{ \cos[2(\psi - \Theta)] (M^2 \mu^2 - M^2 - \mu^2 + 1) + \cos[2(\varphi - \Theta)] \\
&\quad \times [M^2 \mu^2 - M^2 + \mu^2 + 1] + (M^2 \mu^2 + M^2 + \mu^2 + 1) + \frac{1}{2} \langle \cos[2(\psi - \varphi)] \\
&\quad + \cos[2(\psi + \varphi - 2\Theta)] \rangle (M^2 \mu^2 + M^2 - \mu^2 - 1) + \langle \cos[2(\psi - \varphi)] \\
&\quad - \cos[2(\psi + \varphi - 2\Theta)] \rangle (M^2 \mu - \mu) \} \\
&= \frac{r^2}{8} \{ 2 \cos[2(\psi - \Theta)] (\mu^2 - 1) (M^2 - 1) + 2 \cos[2(\varphi - \Theta)] \\
&\quad \times (\mu^2 - 1) (M^2 + 1) + 2(\mu^2 + 1) (M^2 + 1) + \cos[2(\psi - \varphi)] (\mu + 1)^2 (M^2 - 1) \\
&\quad + \cos[2(\psi + \varphi - 2\Theta)] (\mu - 1)^2 (M^2 - 1) \}. \tag{5.15}
\end{aligned}$$

Let us consider the correlation function of transformed image and weighting function shown in Eq. 5.12, where Ω is the definitional domain of image, and the coordinates r_1 and φ_1 are defined in Eqs. 5.10 and 5.11:

$$f(\psi - \Theta) = \iint_{\Omega} T(r_1, \varphi_1) F(r, \varphi, \psi, M) dr d\varphi. \quad (5.16)$$

Let us keep in mind that $F(r_2) = r^2$, and let us use in Eq. 5.16 $T(r, \varphi)$ instead of $T(r_1, \varphi_1)$ because the weighting function F was recalculated to the coordinate system of not transformed (not distorted) image. Then, Eq. 5.16 is presented in the form shown in Eq. 5.17:

$$\begin{aligned} f(\psi - \Theta) = & \iint_{\Omega} T(r, \varphi) r^2 \left\{ \frac{M^2 - 1}{8} \cos[2(\varphi - \psi)] (\mu + 1)^2 \right. \\ & + \frac{M^2 - 1}{8} \cos[2(2\Theta - \psi - \varphi)] (\mu - 1)^2 \\ & + \frac{(\mu^2 - 1)(M^2 + 1)}{4} \cos[2(\varphi - \Theta)] \\ & + \cos[2(\Theta - \psi)] \left[\frac{M^2 - 1}{4} (\mu^2 - 1) \right] \\ & \left. + \frac{(M^2 + 1)}{4} (\mu^2 + 1) \right\} dr d\varphi. \end{aligned} \quad (5.17)$$

Five summands in the curly brackets in Eq. 5.17 can be considered as five independent definite integrals. The fifth summand describes a constant term of correlation function. The third summand does not depend on ψ , thus it also does not depend on $\psi - \Theta$. Therefore, only the first, second and fourth summands in Eq. 5.17 are informative because only they specify nontrivial mode of correlation function $f(\psi - \Theta)$. It can be easily seen that these summands contain only second harmonic component with respect to angle ψ . The direction of anisotropic scaling can be defined accurate within $n\pi$ (where n is arbitrary integer value), which also corresponds to second harmonic component. Thus, it may be supposed that the parameters of harmonic summands of $f(\psi - \Theta)$ in Eq. 5.17 are unambiguously connected with the direction Θ of anisotropic scaling of image $T(r, \varphi)$. The parameters of harmonic summands of $f(\psi - \Theta)$ may be also connected with the angular position ψ of template weighting function shown in Eq. 5.12, which also is defined accurate within $n\pi$. And the parameters of these harmonic summands also can depend on the ratio μ of anisotropic scaling and on the parameter M of template weighting function. Let us assume that using the anisotropic scaling with parameters Θ and μ , the image can be transformed to such form that the correlation function (Eq. 5.17) of this image with the template weighting function of the shape of elliptic paraboloid would not contain any variable components. Let us consider this state of image as its template form with respect to the transform of anisotropic scaling. Then, Θ and $1/\mu$ (or $\Theta + \pi/2$ and μ) will be the parameters of distortion \mathbf{A}_3 of image under analysis (see in Eq. 5.6). For checking the correctness of this assumption, let us prove the following theorem:

Theorem 2 *Let us consider an arbitrary image having zero centroid coordinates (the parabolic core of which has zero vertex coordinates), and the points of this image have nonnegative (or nonpositive) brightness. Let us consider also the weighting function of the shape of elliptic paraboloid having zero coordinates of vertex and unlimited definitional domain. Then, there exists an anisotropic scaling transform that brings the image in such state that the correlation function of the image with the elliptic paraboloid by the parameter of their mutual rotation with respect to coordinate origin is constant. The direction Θ and ratio μ of such anisotropic scaling can be calculated uniquely and are defined by Eqs. 5.18–5.19, where $T(r, \varphi)$ describes the brightness of image points in polar coordinate system, Ω is the definitional domain of image, $\{r, \varphi\}$ are the radius and azimuth of image point.*

$$\mu = \pm \sqrt{\frac{D - \sqrt{C^2 + B^2}}{D + \sqrt{C^2 + B^2}}} \quad (5.18)$$

$$\Theta = 0.5 \tan^{-1}(C/B) + \pi n \quad (5.19)$$

$$B = \iint_{\Omega} T(r, \varphi) r^2 \cos(2\varphi) dr d\varphi$$

$$C = \iint_{\Omega} T(r, \varphi) r^2 \sin(2\varphi) dr d\varphi$$

$$D = \iint_{\Omega} T(r, \varphi) r^2 dr d\varphi$$

The proof of this theorem is rather cumbersome, thus it is not shown here, however it can be found in [20]. Thus, according to Theorem 2, the parameters of anisotropic scaling \mathbf{A}_3 are also determined in analytical form, and \mathbf{A}_3 can be compensated. Now it is possible pass to determining the parameters of transformations \mathbf{A}_4 , \mathbf{A}_5 , и \mathbf{A}_6 of specular reflection, isotropic scaling, and rotation.

5.5 Determining the Parameters of Isotropic Scaling of Image

The transforms \mathbf{A}_5 and $\mathbf{A}_6\mathbf{A}_4$ are the normal divisors of the group of similarities that is a subgroup of the group of plain AT. The transform \mathbf{A}_5 of isotropic scaling does not depend on angular coordinates of image points and does not influence the results of transforms \mathbf{A}_6 and \mathbf{A}_4 , thus the parameter M of transform

$$\mathbf{A}_5 = \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix}$$

can be determined based on image projection upon a subspace invariant to rotation and specular reflection (upon the radius of polar coordinate system [23]). The 1D projections of images upon the radius also have an object-invariant core—the quadratic parabola (corresponding to cylindrical paraboloid being a partial case of Eq. 5.7). The M -fold isotropic scaling of image causes the M -fold scaling of position of each point of its projection upon the radius, thus the coordinate of centroid of projection also is M -fold scaled. The position of centroid can be found using the technique presented above in the Sect. 5.3—by searching for extremum of correlation function of image projection and template weighting function. In this case, the weighting function is a function of a single argument, it is a quadratic parabola. The Eq. 5.9 calculating the Cartesian coordinates of centroid can be presented as a function of single argument. Then, the coefficient M of isotropic scaling of image can be determined using Eq. 5.20, where $T(r, \varphi)$ is the brightness of point of image having zero centroid coordinates in the polar coordinate system, Ω is the definitional domain of image in this coordinate system, $\{r, \varphi\}$ are the radius and azimuth of image point in this coordinate system.

$$M = \frac{\iint_{\Omega} r \cdot T(r, \varphi) dr d\varphi / \iint_{\Omega} T(r, \varphi) dr d\varphi}{G} \quad (5.20)$$

In this case, the coefficient of isotropic scaling is determined by calculating the centroid displacement ratio with respect to its template position G for image projection upon the radius of polar coordinate system.

Thus, the coefficient M of isotropic scaling \mathbf{A}_5 also is determined in analytical form, and it is possible to pass to determining the parameters of rotation \mathbf{A}_6 and specular reflection \mathbf{A}_4 .

5.6 Determining the Parameters of Rotation and Specular Reflection

The rotation \mathbf{A}_6 of image with respect to coordinate origin and the specular reflection \mathbf{A}_4 with respect to a straight line passing through the coordinate origin in Cartesian coordinate system do not change the radial coordinates of image points in polar coordinate system. Thus, the parameters of \mathbf{A}_6 and \mathbf{A}_4 also can be determined using an image projection upon a subspace invariant to radial coordinate changes—upon a circle, i.e. upon the azimuthal coordinate of polar coordinate system. However, the azimuthal coordinate varies cyclically, thus the 1D object-invariant parabolic template functions having unlimited definitional domain cannot be applied in this case. The narrow-band cyclic weighting functions and cyclic object-invariant image cores should be analyzed. The harmonic components of Fourier spectrum calculated for azimuthal projection of image are the good examples of such cyclic cores.

In particular, the angular position of image projection upon a circle is unambiguously defined by the phase of first component of Fourier spectrum of this projection. If the first harmonic component is insufficiently strong, the angular position of projection is unambiguously defined by the phases of any two spectral components having aliquant numbers. This property of projection of image upon a circle is a base of several well known methods of determining of image rotation, see e.g. in [24]. Besides, the analysis of several such harmonic components enables also detection of image specular reflection with respect to straight line passing through coordinate origin [24]. It can be easily understood (see, e.g. in [20]) that this technique of determining the parameters of rotation and specular reflection also corresponds to the image analysis technique based on inspection of correlation function of pattern under analysis with object-invariant templates (just this technique was applied in previous sections). Really, the Fourier transform can be considered as result of analysis of correlation function of template sinusoids with a pattern being inspected.

As an example of practical implementation of this technique, let us consider a method of determining the parameters of rotation and specular reflection on the base of analysis of three neighboring Fourier harmonic components that was proposed in [20]. Let the observed phase φ_k of k th spectral component of image projection on a circle be described by Eq. 5.21, where i_k is the number of not observable full cycles of k th harmonic component ($i_k = 0, 1, \dots, k - 1$), ε is the parameter of specular reflection ($\varepsilon = -1$, if the specular reflection really occurred, otherwise $\varepsilon = 1$), ψ_k is the phase of k th harmonic component in the projection of image having template orientation, φ is the unknown rotation angle of image being recognized.

$$\varphi_k + 2\pi i_k = \varepsilon\psi_k + k\varphi \quad (5.21)$$

The similar formulae (Eqs. 5.22–5.23) can be given for two spectral components having neighboring numbers with conditions (Eqs. 5.24–5.25).

$$\varphi_n + 2\pi i_n = \varepsilon\psi_n + n\varphi \quad (5.22)$$

$$\varphi_m + 2\pi i_m = \varepsilon\psi_m + m\varphi \quad (5.23)$$

$$k - m = 1 \quad (5.24)$$

$$k - n = -1 \quad (5.25)$$

Here and in the further steps of analysis, it is supposed (without loss of generality) that the reflection is accomplished with respect to a straight line having zero inclination angle. Let us subtract Eqs. 5.22 and 5.23 from Eq. 5.21. It will result in the simultaneous linear equations that can be easily solved with respect to φ and ε as it is shown in Eqs. 5.26–5.27.

$$\varphi = \varphi_k - \varphi_m - \varepsilon(\psi_k - \psi_m) \quad (5.26)$$

$$\varepsilon = (2\varphi_k - \varphi_n - \varphi_m)/(2\psi_k - \psi_n - \psi_m) \quad (5.27)$$

The following substitutions were applied for getting the solution: $k - m = 1$, $k - n = -1$, $2\pi(i_k - i_n) = 0$, $2\pi(i_k - i_m) = 0$. The last two substitutions are correct because it may be considered that the rotation angle $\varphi \in [0; 2\pi)$, thus the difference of numbers of not observable full cycles is zero for the harmonic components having neighboring numbers.

One can encounter the cases, when strong enough harmonic components, satisfying the demands shown in Eqs. 5.24–5.25 do not exist in the spectrum of image projection upon the circle. Some different solutions are possible in such cases [24], however the method of determining the parameters of rotation and reflection described above worked satisfactory in numerous practical experiments.

Thus, the described in this section method of determining the parameters of rotation and reflection also proceeds within the frameworks of general technique proposed in this chapter. It is based on analysis of narrow-band object-invariant cores contained in images. A method of estimating the parameters of rotation and reflection based on image core of parabolic form also was proposed. The interested readers can find this information in [25].

5.7 Iterative Compensation and Estimation of Full Projective Transform

Let us pass to analysis of the rest part of PT. This part has not yet been considered, and it corresponds to denominators of Eq. 5.1. The plain PT is described by the mathematical expressions Eqs. 5.28–5.29 connecting the coordinate vectors $\vec{X} = (x, y)^T$ of initial images with coordinate vectors $\vec{X}' = (x', y')^T$ of transformed images, where a_1, \dots, a_6 are the parameters of AT described in Eq. 5.1, while the parameters a_7 and a_8 describe the nonlinear part of PT.

$$x' = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1} = \frac{a_1x + a_2y + a_3}{\vec{X} \cdot \vec{P} + 1} \quad (5.28)$$

$$y' = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1} = \frac{a_4x + a_5y + a_6}{\vec{X} \cdot \vec{P} + 1} \quad (5.29)$$

It can be seen in Eqs. 5.28–5.29 that the denominator of formulae of PT can be expressed in terms of scalar product $\vec{X} \cdot \vec{P}$ of the vector of Cartesian coordinates of image point and vector $\vec{P} = (a_7, a_8)^T$ of parameters. Here and in the text presented below, the transform corresponding to denominators of Eqs. 5.28 and 5.29 will be

mentioned as Nonlinear part of Projective Transform (NPT) because this is a part of transform that differs the full PT from linear AT.

The PT is nonlinear with respect to parameters of its denominator, thus the vector \vec{P} of its parameters could not be determined in analytical form as it was done for the parameters of AT. And moreover, the NPT causes the displacements of image centroid and disturbs the results of image normalization with respect to other parts of AT, while the AT substantially influences the results of NPT, if the last one follows the AT. Thus, it turned out to be impossible to separate the procedures of determining and compensation of AT and NPT (as it was done above for the simple parts of AT). The problems of determining the parameters of AT and NPT should be solved in a complex way keeping in mind their interaction.

The problem is solved based on iterative optimization of solution in the space of parameters a_7 and a_8 . The solution method proposed in this chapter exploits the specific property of NPT: in contrast to AT, the PT causes rearrangement of image points with respect to image centroid. It can be easily shown analytically in the following way. Let us present Eq. 5.28 in the form shown in Eq. 5.30, where α corresponds to the numerator of Eq. 5.28, $|\cdot|$ means absolute value, $\beta = \vec{X} \cdot \vec{P}$.

$$x' = \alpha / (1 + \text{sign}(\beta) \cdot |\beta|) \quad (5.30)$$

Let (for definiteness) the vector \vec{P} be antiparallel to abscissa axis, i.e. its scalar product with the unit vector of abscissa axis is negative. Let the image centroid has zero coordinates. If $\beta \neq 0$, the points $x' = \alpha / (1 - |\beta|)$ are situated to the right from the ordinate axis and the points $x' = \alpha / (1 + |\beta|)$ are situated to the left from the ordinate axis. Using simple mathematical manipulations, it is easy to show that $\alpha / (1 + |\beta|) - \alpha / 1 < 0$, $\alpha / (1 - |\beta|) - \alpha / 1 > 0$, and $|\alpha / (1 + |\beta|) - \alpha / 1| < |\alpha / (1 - |\beta|) - \alpha / 1|$. Therefore, the negative abscissas of image points will decrease in absolute value, and the positive abscissas of points will increase in their absolute values as a result of NPT corresponding to assumed direction of \vec{P} . This will cause a displacement of centroid to the domain of positive abscissas. The changes of positive abscissas are larger in absolute values than the changes of negative abscissas. The larger are the positive abscissas of points, the faster is their movement resulting from this NPT, and the stronger is “rarefying” of their coordinates. At the same time, the larger are the absolute values of negative abscissas, the stronger is “concentration” of absolute values of coordinates. Thus, when the centroid displacement resulting from such NPT is compensated, there will become more image points to the left of ordinate axis (this part of image becomes more “massive”), but these points will have in average smaller absolute values of abscissas than the abscissas of points to the right of ordinate axis. Such reasoning would be also correct for any other direction of vector \vec{P} of parameters of NPT because all such cases may be reduced to the considered partial situation by means of rotation of coordinate system.

Let us consider an arbitrary image having zero centroid coordinates. Let us divide the image into two semi-images by an arbitrary straight line passing through coordinate origin. The semi-images can be considered as two halves of dipole having the

masses M_1 and M_2 located at centroids of semi-images being at the distances L_1 and L_2 from coordinate origin. The dipole is balanced with respect to centroid of the whole image, however it is asymmetric, i.e. $M_1L_1 = M_2L_2$, but $M_1 \neq M_2$ and $L_1 \neq L_2$. Let $L_2 > L_1$, and let the ratio L_2/L_1 be maximum in the direction Θ . Then, the image can be presented as a dipole oriented along the direction Θ , and this dipole may be described with a vector \vec{L} having the length L_2/L_1 and directed along the longer arm L_2 of dipole. Let the case corresponding to Eq. 5.31 be the template state of image (with respect to NPT)

$$M_1 = M_2 \quad L_1 = L_2. \quad (5.31)$$

In such state, the dipole is symmetric and balanced with respect to image centroid for any direction of straight line dividing the image into two semi-images, i.e. $|\vec{L}| = 1$ for any direction Θ of dipole arms. A circle centered at coordinate origin can be an example of template image.

As it was shown above, if any NPT having the vector \vec{P} of parameters $a_7 \neq 0$ and/or $a_8 \neq 0$ was applied to image of circle, then it rearranges the circle points with respect to its centroid, thus the condition presented in Eq. 5.31 will be violated, and the centroid of image will be displaced. The rotation of vector \vec{P} will cause the rotation of direction of centroid displacement, but the value of displacement will remain constant. Thus, if some time in the past, an image of circle centered with respect to coordinate origin was subjected to NPT with parameters \vec{P} , and its centroid was brought again to coordinate origin, then the present image of transformed circle will correspond to asymmetric dipole for which $|\vec{L}| > 1$. Now, let us apply to this transformed image of circle an additional “testing NPT” with rotating vector \vec{P}_t of parameters. In such case, different directions of \vec{P}_t will cause different absolute values of displacement of centroid of image. The “velocity” of displacement will be maximum for some direction $\Phi = \Phi_{\max}$ of vector \vec{P}_t , and in the case of image of circle, Φ_{\max} corresponds to the direction of vector \vec{P} of parameters of NPT applied to the circle in the past. For understanding of this statement, remember that the circle is absolutely symmetric, thus the image of circle subjected to NPT may be considered as dipole of the same length for any direction of vector \vec{P} . Now, let us prove the following lemma.

Lemma 2 *Let us consider a case, when some NPT having the vector \vec{P}_t of parameters is applied to an asymmetric dipole with zero coordinates of centroid (the dipole is balanced with respect to coordinate origin), and the direction of \vec{P}_t is antiparallel to the direction Θ of dipole. Then, the displacement of centroid of dipole subjected to such NPT will be larger in absolute value than the displacement corresponding to the case, when the direction of \vec{P}_t is parallel to the direction Θ of dipole.*

The proof of Lemma 2 is presented in Appendix 1.

It can be seen from the proof of Lemma 2 that the more is the coefficient N of asymmetry of dipole,—the more is the difference of centroid displacements corresponding to oppositely directed vectors \vec{P}_i of NPT parameters.

The statements presented above are correct for image of circle, however the real images have more complicated spatial distributions of brightness of pixels that are asymmetric in general case. For showing the correctness of results of Lemma 2 for arbitrary real pictures, let us assume that there was an arbitrary image having initially a template state satisfying the condition presented in Eq. 5.31, and this image was subjected to some NPT. Then the centroid of this image was brought to coordinate origin, thus this image became an asymmetric dipole balanced with respect to coordinate origin and oriented along the abscissa axis. The right arm of dipole has the length L_2 , and it is N times longer than the left arm ($M_1 = NM_2$, $L_2 = NL_1$, $N > 1$). It would be logical to try to prove a supposition that such arbitrary image could be presented as a set of balanced asymmetric dipoles oriented along the abscissa axis. A particular case of such proof was considered in [26], where each arm of each dipole was composed of a single pixel or of a single subpixel, and where for each dipole, the conditions $M_1 = NM_2$, $L_2 = NL_1$ were satisfied for a constant (through the whole set of dipoles) $N > 1$. It should be explained that, in [26] as well as in the proofs presented below, a pixel is sometimes presented (for the needs of proving) as a kind of material body having the mass equal numerically to pixel brightness (energy). This body is considered as aggregation of subpixels each having the same coordinates equal to coordinates of this pixel, while the sum of individual masses of these subpixels is the mass of the whole body (equal to energy of pixel). Thus, in the text below the parts of such aggregation corresponding to “subpixels energy” are referred to as “subpixels” or “subpixels”. However, the restricted partial case of proof considered in [26] is not good for arbitrary images. In more general case, an arbitrary image can be presented as a set of balanced dipoles having constant ratio N of arm lengths, and it can be done in two following ways:

- The right arm of dipole is composed of a single pixel or of a subpixel situated to the right of ordinate axis (at the right) and having abscissa L and mass (brightness) M . The left arm of dipole is composed of a single pixel or of a subpixel or it is composed of several pixels or subpixels situated to the left of ordinate axis (at the left). The centroid of this multitude of pixels or subpixels has the abscissa l while the mass m of this multitude is localized at the point of centroid.
- The left arm of dipole is composed of a single pixel or of a subpixel situated to the left of ordinate axis (at the left) and having abscissa l and mass (brightness) m . The right arm of dipole is composed of a single pixel or of a subpixel or it is composed of several pixels or subpixels situated to the right of ordinate axis (at the right). The centroid of this multitude of pixels or subpixels has the abscissa L while the mass M of this multitude is localized at the point of centroid.

The Algorithm 1 presented below shows a possible way of such decomposition for the image under consideration.

Algorithm 1

1. Choosing a pixel or subpixel (pixel—for simplicity of further consideration) not considered and not marked yet with special flag at the left. The pixel has the moment $\mu_l = m \cdot l$ with respect to ordinate axis, where m and l are the mass (brightness) and abscissa of this pixel, respectively. If such pixel was not found, passing to the item 9, otherwise passing to the item 2.
2. Searching for a not considered yet pixel at the right (let the pixel found has the abscissa L , mass M , and moment $\mu_r = M \cdot L$ with respect to ordinate axis, where $M \geq m/N$, $L = -N \cdot l$). If such pixel was found, passing to the item 3, otherwise passing to the item 5.
3. Equipointing the pixel taken at the left according to item 1 (it has the moment $\mu_l = m \cdot l$) by a pixel or subpixel found at the right according to item 2 (it has the moment $\mu_r^a = -m \cdot l = -\mu_l$).
4. If $M = m/N$, the pixel chosen at the left according to item 1 and pixel chosen at the right according to item 2 entirely equipoint each other and are excluded from further consideration. If $M > m/N$, a part of mass $M_1 = m/N$ of the pixel chosen at the right is excluded from further consideration together with the pixel chosen at the left. The non-equipointed part of mass $M_2 = M - M_1$ of the pixel chosen at the right is considered as a pixel remaining for further consideration. The entire image remains balanced with respect to ordinate axis and may be considered as a new dipole having the same ratio $m_c/M_c = N$ of masses and same ratio of arm lengths ($L_c = -N \cdot l_c$). Passing to the item 1.
5. If no unconsidered yet pixels having abscissa $L = -N \cdot l$ were found according to item 2 because there were no such pixels with non-zero mass in image, then it means that the pixel chosen at the left according to item 1 or its subpixel can not be equipointed immediately by a single pixel or by a subpixel at the right having abscissa $L = -N \cdot l$. For resolving this problem, marking all pixels having abscissa l with a special flag and passing to the item 1, otherwise, passing to the item 6.
6. Choosing an unconsidered yet pixel having abscissa $L = -N \cdot l$ at the right (let this pixel have the mass M and moment $\mu_r = M \cdot L$ with respect to ordinate axis). Such pixel will be found according to item 5. Passing to the item 7.
7. Equipointing the pixel taken at the right according to item 6 and having the moment $\mu_r = M \cdot L$ by a subpixel found at the left according to item 1 having the moment $\mu_l^a = -\mu_r = -M \cdot L$.
8. The pixel chosen at the right according to item 6 is excluded from further consideration. The subpixel chosen at the left according to item 1 and having the mass $m_1 = M \cdot N$ is excluded from further consideration, and only a non-equipointed subpixel of this pixel having the mass $m_2 = m - m_1$ is further considered. The whole image remains balanced with respect to ordinate axis and may be considered as a dipole having the mass ratio $m_c/M_c = N$ and arm lengths ratio $|L_c/l_c| = N$. Passing to the item 1.

9. If no unconsidered pixels and no pixels not marked with special flag remain in the image, then passing to the item 29. Otherwise, the multitude of marked pixels non-equiposed yet in the items 1–8 composes asymmetric dipole balanced with respect to ordinate axis. The pixels having positive abscissas compose the right arm of dipole. The length L_c of right arm corresponds to centroid of the multitude of pixels having positive abscissas and aggregate mass M_c . The pixels having negative abscissas compose the left arm of dipole. The length $-l_c$ of left arm corresponds to centroid of the multitude of pixels having negative abscissas and aggregate mass m_c . $L_c = -N \cdot l_c$, $M_c = m_c/N$. For continuing the procedure of mutual equiposing of remaining pixels with respect to ordinate axis, passing to the item 10.
10. Choosing an unconsidered yet pixel at the left having minimum in absolute value negative abscissa. Let it has abscissa l , mass m , and moment $\mu_l^0 = m \cdot l$ with respect to ordinate axis. If such pixel was not found, it means that all pixels had been already considered and mutually equiposed; then passing to the item 29. Otherwise, there should also remain some pixels at the right that could equipose the pixel found at the left.
11. If $l \neq l_c$ ($|N \cdot l| \neq L_c$), then passing to the item 12. If $l = l_c$ ($|N \cdot l| = L_c$), then the abscissa of pixel chosen at the left according to item 10 is equal to the abscissa l_c of centroid of multitude of pixels situated to the left of ordinate axis. Thus, according to the item 9, the chosen pixel can be equiposed by the multitude of all unconsidered pixels or subpixels situated at the right, the mass of equiposing subpixel of i th pixel at the right is $M_i^a = M_i \cdot m/m_c$, where M_i is the mass of i th pixel at the right. The centroid of the multitude of pixels at the right has the abscissa $L_c = -N \cdot l_c = -N \cdot l$. After such equiposing, the pixel chosen at the left according to item 10 and having the mass m and abscissa l is excluded from further consideration. The pixels or subpixels at the right having the masses M_i^a are excluded from further consideration. The remaining subpixels at the right having the masses $M_i^b = M_i - M_i^a$ are left for further consideration. The whole image remains balanced with respect to ordinate axis and may be considered as a dipole having the ratio $m_c/M_c = N$ of changed masses and arm lengths ratio $|L_c/l_c| = N$. Passing to the item 10.
12. Searching for an unconsidered pixel at the right having minimum abscissa $L_1 < |N \cdot l|$. Let the pixel found have the mass M_1 , moment $\mu_{r1}^0 = M_1 \cdot L_1$ with respect to ordinate axis, and moment $\mu_{r1}^L = M_1 \cdot (L_1 - L)$ with respect to the straight line parallel to ordinate axis and crossing the abscissa axis at the point $L = |N \cdot l|$. If such pixel was not found, then it means that a pixel at the right having abscissa $L_1 > |N \cdot l|$ exists, that could be used for equiposing the pixel chosen at the left according to item 10 (the case $L_1 = |N \cdot l|$ was considered in the items 1–8). Passing to the item 20 for analyzing such situation, otherwise passing to the item 13.
13. Searching for a not considered yet pixel at the right having abscissa $L_2 > |N \cdot l|$, let it have the mass M_2 , moment $\mu_{r2}^0 = M_2 \cdot L_2$ with respect to ordinate axis, and moment $\mu_{r2}^L = M_2 \cdot (L_2 - L)$ with respect to the straight line parallel to ordinate axis and crossing the abscissa axis at the point $L = |N \cdot l|$. Such pixel should be

found according to the following reasons. Let the multitude of unconsidered pixels at the left have the abscissa l_c of centroid and aggregate mass m_c . This multitude is equipoised by the multitude of not considered yet pixels at the right having the centroid abscissa $L_c = -m_c \cdot l_c / M_c = -N \cdot l_c$, where M_c is the aggregate mass of pixels at the right. The pixel chosen at the left according to item 10 has the minimum in absolute value negative abscissa l , $|l| < |l_c|$ (the case $l = l_c$ was considered in the item 11), and a pixel having abscissa $L_1 < |N \cdot l| < |N \cdot l_c| = L_c$ was chosen according to item 12 for equipoising it. Thus a pixel having abscissa $L_2 > L_c = |N \cdot l_c| > |N \cdot l|$ exists in the multitude of unconsidered pixels at the right.

14. If $|\mu_{r2}^L| > |\mu_{r1}^L|$, the following sequence of operations is accomplished: choosing the subpixel having the mass $M_2^a = M_1 \cdot (L - L_1) / (L_2 - L)$ belonging to the pixel having the mass M_2 found according to item 13, while the subpixel having the mass $M_2^b = M_2 - M_2^a$ belonging to this pixel is left for further consideration; choosing as a whole the pixel having the mass $M_1^a = M_1$ found according to item 12; passing to the item 17.
15. If $|\mu_{r2}^L| < |\mu_{r1}^L|$, the following sequence of operations is accomplished: choosing the subpixel having the mass $M_1^a = M_2 \cdot (L_2 - L) / (L - L_1)$ belonging to the pixel having the mass M_1 found according to item 12, while the subpixel having the mass $M_1^b = M_1 - M_1^a$ belonging to this pixel is left for further consideration; choosing as a whole the pixel having the mass $M_2^a = M_2$ found according to item 13; passing to the item 17.
16. If $|\mu_{r2}^L| = |\mu_{r1}^L|$, choosing as a whole the both pixels having the masses $M_1^a = M_1$ and $M_2^a = M_2$ found according to items 12 and 13, respectively, and passing to the item 17.
17. If the condition $M_1^a \cdot L_1 + M_2^a \cdot L_2 = -m \cdot l$ is satisfied for the pixel having the mass m chosen according to item 10 and for the pixels or subpixels having the masses M_1^a , M_2^a chosen according to items 14, 15, 16, then the following is accomplished: these pixels or subpixels are considered as mutually equipoised and excluded from further consideration (the whole image remains balanced with respect to ordinate axis and may be considered as a dipole having the ratio $m_c / M_c = N$ of changed masses and ratio $|L_c / l_c| = N$ of changed lengths of arms); passing to the item 10.
18. If the condition $M_1^a \cdot L_1 + M_2^a \cdot L_2 < -m \cdot l$ is satisfied for the pixel having the mass m chosen according to item 10 and for the pixels or subpixels having the masses M_1^a , M_2^a chosen according to items 14, 15, 16, then the following is accomplished: the subpixel of mass m at the left having the mass $m^a = (M_1^a \cdot L_1 + M_2^a \cdot L_2) / |l|$ is equipoised by the pixels or subpixels having the masses M_1^a , M_2^a at the right; the mutually equipoised in this way pixels and subpixels are excluded from further consideration (the whole image remains balanced with respect to ordinate axis and may be considered as a dipole having the ratio $m_c / M_c = N$ of changed masses and ratio $|L_c / l_c| = N$ of changed lengths of arms); the subpixel of mass $m^b = m - m^a$ of the pixel at the left remains for further consideration; passing to the item 10.

19. If the condition $M_1^a \cdot L_1 + M_2^a \cdot L_2 > -m \cdot l$ is satisfied for the pixel having the mass m chosen according to item 10 and for the pixels or subpixels having the masses M_1^a, M_2^a chosen according to items 14, 15, 16, then the following is accomplished: the pixel at the left having the mass m is equipoised by the subpixels at the right having the masses $M_1^c = k \cdot M_1^a$ and $M_2^c = k \cdot M_2^a$, where $k = -m \cdot l / (M_1^a \cdot L_1 + M_2^a \cdot L_2)$; the equipoised pixel at the left having the mass m is excluded from further consideration; the subpixels at the right having the masses M_1^c and M_2^c are excluded from further consideration (the whole image remains balanced with respect to ordinate axis and may be considered as a dipole having the ratio $m_c/M_c = N$ of changed masses and ratio $|L_c/l_c| = N$ of changed lengths of arms); the subpixels at the right having the masses $M_1^d = M_1^a - M_1^c$ and $M_2^d = M_2^a - M_2^c$ remain for further consideration (the subpixels at the right having the masses M_1^b and M_2^b that were left for further consideration according to items 14 and 15 also should not be forgotten); passing to the item 10.
20. Searching for a not considered pixel at the right having minimum abscissa $L_2 > -N \cdot l$. Such pixel exists because no pixels having abscissas satisfying the condition $L_1 \leq -N \cdot l$ were found according to item 12, while at least one pixel should be found at the right for equipoising the pixel chosen at the left according to item 10. Let the pixel found at the right have the mass M_2 and moment $\mu_r^0 = L_2 \cdot M_2$ with respect to ordinate axis. The pixel having abscissa l chosen at the left according to item 10 will have the moment $\mu_l^{L_2} = (l + L_2/N) \cdot m$ with respect to the straight line parallel to ordinate axis and crossing the abscissa axis at the point $-L_2/N$.
21. Let the multitude of unconsidered yet pixels at the left have the centroid abscissa l_c and aggregate mass m_c . This multitude at the left is equipoised by the multitude of unconsidered pixels at the right having the centroid abscissa $L_c = -m_c \cdot l_c / M_c = -N \cdot l_c$, where M_c is the aggregate mass of multitude of pixels at the right. If $L_2 \neq |N \cdot l_c|$, passing to the item 22. Otherwise, $L_2 = |N \cdot l_c|$, which means that the abscissa of pixel found at the right according to item 20 is N -fold larger in absolute value than the abscissa of centroid of the whole multitude of not considered yet pixels at the left. As the multitude of pixels at the left is equipoised by the multitude of pixels at the right, the abscissa L_2 is equal to the centroid abscissa of unconsidered pixels at the right, i.e. $L_2 = L_c$. Thus, for equipoising the pixel having abscissa L_2 that was found at the right according to item 20, the subpixels of every (i th) pixel are chosen at the left having the masses $m_i^a = m_i \cdot M_2/M_c$, where m_i is the mass of i th pixel at the left, M_c is the aggregate mass of all unconsidered yet pixels at the right. After such juxtaposing, the pixel chosen at the right according to item 20 having the mass M_2 and abscissa L_2 is considered to be equipoised and is excluded from further consideration. The subpixels at the left having the masses m_i^a are equipoised and excluded from further consideration. The whole image remains balanced with respect to ordinate axis and may be considered as a dipole having the ratio $m_c/M_c = N$ of changed masses and ratio $|L_c/l_c| = N$ of

- changed lengths of arms. The subpixels at the left having the masses $m_i^b = m_i - m_i^a$ remain for further consideration. Passing to the item 10.
22. Searching for an unconsidered pixel at the left having abscissa $l_2 < -L_2/N$ ($|l_2| > L_2/N$). Let the pixel found have the mass m_2 , and let its moment with respect to ordinate axis be $\mu_{l_2}^0 = m_2 \cdot l_2$, and its moment with respect to a straight line parallel to ordinate axis and crossing the abscissa axis at the point $-L_2/N$ be $\mu_{l_2}^{L_2} = m_2 \cdot (l_2 + L_2/N)$. Such pixel should exist according to the following reason. Let the multitude of unconsidered pixels at the left have the centroid abscissa l_c and aggregate mass m_c . This multitude at the left is equipoised by the multitude of unconsidered pixels at the right having the centroid abscissa $L_c = -m_c \cdot l_c / M_c = -N \cdot l_c$, where M_c is the aggregate mass of unconsidered pixels at the right. The pixel chosen at the right according to item 20 has the minimum (among all pixels at the right) positive abscissa $L_2 < L_c = -N \cdot l_c$ (the case of $L_2 = L_c = -N \cdot l_c$ was considered in the item 21), and a pixel having the mass m and abscissa l ($|l| < L_2/N < L_c/N = |l_c|$) was chosen according to item 10 for equipoising it. Thus, a pixel having abscissa $|l_2| > |l_c| = L_c/N > L_2/N$ (i.e. $l_2 < l_c = -L_c/N < -L_2/N$) should exist in the multitude of unconsidered pixels at the left.
 23. If $|\mu_{l_2}^{L_2}| > |\mu_l^{L_2}|$, then accomplishing the following sequence of operations: choosing a subpixel of mass $m_2^g = m \cdot |l + L_2/N|/|l_2 + L_2/N|$ of the pixel having the mass m_2 found according to item 22, while the subpixel having the mass $m_2^b = m_2 - m_2^g$ of this pixel is left for further consideration; taking as a whole the pixel of mass $m^a = m$ chosen according to item 10; passing to the item 26.
 24. If $|\mu_{l_2}^{L_2}| < |\mu_l^{L_2}|$, then accomplishing the following sequence of operations: taking the subpixel having the mass $m^a = m_2 \cdot |l_2 + L_2/N|/|l + L_2/N|$ of the pixel of mass m chosen according to item 10, while the subpixel having the mass $m^b = m - m^a$ of this pixel is left for further consideration; taking as a whole the pixel of mass $m_2^g = m_2$ found according to item 22; passing to the item 26.
 25. If $|\mu_{l_2}^{L_2}| = |\mu_l^{L_2}|$, then accomplishing the following sequence of operations: taking as a whole the both pixels having the masses $m^a = m$ and $m_2^g = m_2$ found according to items 10 и 22; passing to the item 26.
 26. If the condition $m^a \cdot l + m_2^g \cdot l_2 = -M_2 \cdot L_2$ is satisfied for the pixel of the mass M_2 found according to item 20 and for the pixels or subpixels having the masses m^a and m_2^g chosen at the left according to items 23, 24, 25, then the following sequence of operations is accomplished: these pixels and subpixels are considered as mutually equipoised and removed from further consideration (the whole image remains balanced with respect to ordinate axis and may be considered as a dipole having the ratio $m_c/M_c = N$ of changed masses and ratio $|L_c/l_c| = N$ of changed lengths of arms); passing to the item 10.
 27. If the condition $M_2 \cdot L_2 > |m^a \cdot l + m_2^g \cdot l_2|$ is satisfied for the pixel of the mass M_2 found according to item 20 and for the pixels or subpixels having the masses m^a and m_2^g chosen at the left according to items 23, 24, 25, then the following sequence of operations is accomplished: equipoising the subpixel of mass $M_2^g = |m^a \cdot l + m_2^g \cdot l_2|/L_2$ of the pixel of mass M_2 at the right by the

pixels or subpixels at the left having the masses m^a and m_2^a and excluding these mutually equipoised pixels and subpixels from further consideration (the whole image remains balanced with respect to ordinate axis and may be considered as a dipole having the ratio $m_c/M_c = N$ of changed masses and ratio $|L_c/l_c| = N$ of changed lengths of arms); the subpixel of the mass $M_2^b = M_2 - M_2^a$ at the right is left for further consideration; passing to the item 10.

28. If the condition $M_2 \cdot L_2 < |m^a \cdot l + m_2^a \cdot l_2|$ is satisfied for the pixel of the mass M_2 found according to item 20 and for the pixels or subpixels having the masses m^a and m_2^a chosen at the left according to items 23, 24, 25, then the following sequence of operations is accomplished: equipoising the pixel having the mass M_2 at the right by the subpixels at the left having the masses $m^c = k \cdot m^a$ and $m_2^c = k \cdot m_2^a$, where $k = M_2 \cdot L_2 / |m^a \cdot l + m_2^a \cdot l_2|$; the mutually equipoised pixel of mass M_2 at the right and subpixels having the masses m^c and m_2^c at the left are excluded from further consideration (the whole image remains balanced with respect to ordinate axis and may be considered as a dipole having the ratio $m_c/M_c = N$ of changed masses and ratio $|L_c/l_c| = N$ of changed lengths of arms); the subpixels having the masses $m^d = m^a - m^c$ and $m_2^d = m_2^a - m_2^c$ at the left remain for further consideration (also the subpixels at the left having the masses m^b and m_2^b left remain for further consideration according to items 23 and 24); passing to the item 10.
29. The end of the algorithm. Every pixel or subpixel at the left of ordinate axis was equipoised by a single pixel or by several pixels or subpixels at the right of ordinate axis. Every non-equipoised pixel or subpixel at the right of ordinate axis was equipoised by a single pixel or several pixels or subpixels at the left of ordinate axis.

The Algorithm 1 described above enables presenting any image having zero centroid coordinates as a multitude of simple dipoles balanced with respect to one and the same straight line passing through the coordinate origin, and all these dipoles have the same ratio N of arm lengths. However, one of the arms of such dipoles may not be composed of a single pixel or subpixel, but of several pixels or subpixels having differing coordinates. The presented above proof of Lemma 2 is not valid in such case because it stipulated that the mass of each arm of dipole was concentrated in a single pixel. The Lemma 2 concluded that there is a difference of displacements of centroid of asymmetric dipole resulting from oppositely directed NPT. Let us show, what conditions should be satisfied for keeping valid the conclusions of Lemma 2 for the following more complex cases of dipole composition considered in Algorithm 1:

- The right arm of dipole is composed of a single pixel or of a subpixel situated to the right of ordinate axis (at the right) and having abscissa L and mass (brightness) M . The left arm of dipole is composed of a single pixel or of a subpixel or it is composed of several pixels or subpixels situated to the left of ordinate axis (at the left), and the centroid of this multitude of pixels or

subpixels at the left has the abscissa l while the mass m of this multitude is localized at the point of centroid.

- The left arm of dipole is composed of a single pixel or of a subpixel situated to the left of ordinate axis (at the left) and having abscissa l and mass (brightness) m . The right arm of dipole is composed of a single pixel or of a subpixel or it is composed of several pixels or subpixels situated to the right of ordinate axis (at the right), and the centroid of this multitude of pixels or subpixels at the right has the abscissa L while the mass M of this multitude is localized at the point of centroid.

Keeping in mind these two dipole configurations, two versions of conditions will be considered that should be satisfied for correctness of the following statement: if an NPT described with vector \vec{P}_t of parameters is applied to asymmetric dipole, then the absolute value of displacement of centroid of asymmetric dipole in the case, when the direction of \vec{P}_t is antiparallel to the direction of dipole is larger, than in the case, when the direction of \vec{P}_t is parallel to direction of dipole (where direction of dipole is the direction of its longer arm). This consideration will be accomplished in the form of proofs of two lemmas presented below. For convenience of description, a particular case will be considered corresponding to dipole lying on abscissa axis, which could be extended to any general case by simple conversion of coordinate system.

Lemma 3 *Let a dipole lying on abscissa axis exist and be balanced with respect to ordinate axis. Let the left (shorter) arm of this dipole correspond to a single pixel or subpixel lying to the left of ordinate axis (at the left) and having the absolute value l of abscissa and mass (brightness) m . Let the right arm of dipole be N -fold longer than the left arm ($N > 1$) and correspond to a multitude of two pixels or subpixels at the right having the differing masses M_1 and M_2 and abscissas L_1 and L_2 ($L_2 > L > L_1$), while the centroid of these pixels at the right has the abscissa $L = N \cdot l$ and aggregate mass $M = m/N$ concentrated at centroid. Then, there exists an arbitrarily small in absolute value vector \vec{P}_t of parameters of NPT formed according to Eqs. 5.28–5.29, oriented along the abscissa axis and having the following properties: if \vec{P}_t is parallel to abscissa axis, then the displacement of dipole centroid caused by such NPT is smaller in absolute value than the centroid displacement caused by NPT corresponding to vector \vec{P}_t antiparallel to abscissa axis.*

The proof of Lemma 3 is presented in Appendix 2.

Lemma 4 *Let a dipole lying on abscissa axis exist and be balanced with respect to ordinate axis. Let the left (shorter) arm of this dipole correspond to a multitude of two pixels or subpixels situated to the left of ordinate axis (at the left) and having the masses (brightness) M_1 and M_2 and the abscissas differing in the absolute values L_1 and L_2 ($L_2 > L > L_1$). The centroid of multitude of pixels at the left has the abscissa L and mass $M = M_1 + M_2$ concentrated at the point of centroid. Let the right arm of dipole be $1/N$ times longer than the left arm ($0 < N < 1$). The right arm corresponds*

to a single pixel or subpixel situated to the right of ordinate axis (at the right) and having the abscissa $l = L/N$ and mass $m = M \cdot N$. Then, an arbitrarily small in absolute value vector \vec{P}_t of parameters of NPT formed according to Eqs. 5.28–5.29, oriented along the abscissa axis and having the following properties, exists: if \vec{P}_t is parallel to abscissa axis (is directed along the longer arm of dipole, is parallel to dipole), then the displacement of dipole centroid caused by such NPT is smaller in absolute value than the centroid displacement caused by NPT corresponding to vector \vec{P}_t antiparallel to abscissa axis (antiparallel to dipole).

The proof of Lemma 4 is presented in Appendix 3.

Remark The Lemmas 3 and 4 consider the dipoles, one arm of which is composed of a single pixel or its subpixel, while the other arm is composed of two pixels or their subpixels. However, a quite different case is considered in the items 11 and 21 of Algorithm 1: a single pixel F or its subpixel situated in one semi plane with respect to ordinate axis is balanced with a multitude $f = \{f_i, i = 1 \dots n\}$ of subpixels situated in the opposite semi plane, and the size of this multitude is larger than 2 in general case. However, it can be easily shown (see e.g. in [26]) that the multitude f can be presented as a set of pairs $\{f'_k, f'_l, k = 1 \dots n, l = 1 \dots n, k \neq l\}$ of subpixels or their sub-parts balanced along the abscissa axis with respect to the point of centroid of this multitude. Then, each pair $\{f'_k, f'_l\}$ situated in one semi plane can be balanced (with respect to the ordinate axis) with some subpixel F'_{kl} of pixel F or of subpixel F situated in the other semi plane. Thus, the Lemmas 3 and 4 will also provide the correctness of the items 11 and 21 of Algorithm 1.

It was proven in the Lemmas 2, 3, 4 that if some testing NPT was applied to a symmetric dipole deformed initially by some other NPT, then the dipole centroid displacement caused by testing NPT was larger in the case, when the testing and initial NPTs were parallel, as compared with the case, when these two transforms were antiparallel. The proving was accomplished for the case, when the “masses” of each of arms of dipole were concentrated in one or two image points. Here and in the further discussion, the direction of vector \vec{P} of parameters of the NPT is also referred to as the direction of this NPT. According to Eqs. 5.28–5.29, the dipole displacements orthogonal to direction of NPT do not influence the presented above results. It should be only provided that the dipole centroid would not be displaced with respect to coordinate origin along the direction of dipole.

Let us consider an arbitrary 2D image having zero coordinates of centroid, and let us present this image as a multitude of balanced asymmetric simple dipoles. Let these dipoles have the same ratios of arm lengths in the direction of vector \vec{P}_t of parameters of testing NPT, and let the centroids of dipoles be not displaced with respect to coordinate origin in the direction parallel or antiparallel to \vec{P}_t as it was shown in the Algorithm 1. Then, the proofs of Lemmas 2, 3, 4 will be true for every simple dipole composing the image and for the whole image, if this image has restricted definitional domain and zero abscissa of centroid. In the general case of

arbitrary 2D image, the direction of asymmetric balanced dipole corresponding to it may not coincide with the direction of abscissa axis. However, according to the Lemma 2, the more is the asymmetry of dipole in the direction of testing NPT,—the more will be the difference of dipole centroid displacement velocities in the directions parallel and antiparallel to testing NPT. It is correct for any dipoles considered in the Lemmas 2, 3, 4, thus it will be correct for the whole images composed of such dipoles. Therefore, by means of rotating the vector \vec{P}_t of parameters of testing NPT, the direction of \vec{P}_t corresponding to maximum difference of displacement velocities can be found, and this direction will coincide with the direction of image dipole having maximum coefficient N of asymmetry.

At the beginning of this section, it was demonstrated using Eqs. 5.28–5.29 that any NPT results in rearrangement of image points with respect to the point of its centroid. If an initial image corresponded to symmetric balanced dipole, i.e. if the condition presented in Eq. 5.31 was satisfied, then this condition would be violated as a result of any NPT applied to image. Then, the transformed image would correspond to an asymmetric dipole the direction of which (the direction corresponding to the maximum value of coefficient N of its asymmetry) would not in general case coincide with the direction of vector \vec{P} of parameters of this NPT. However, it can be concluded from Eqs. 5.28–5.29 that the direction of displacement of pixels as well as the direction of maximum image “sparseness” occurring as a result of NPT always have nonpositive projection on the vector \vec{P} of parameters of this NPT. This enables formulating the Lemma 5, the proof of which would create a base for proving the convergence of iterative procedure of compensation of the NPT described below with the Algorithm 2.

Lemma 5 *Let us consider an image having restricted definitional domain. This image corresponded initially to a symmetric dipole, but it started corresponding to an asymmetric dipole as a result of some NPT (having vector of parameters \vec{P}), and the dipole is now balanced with respect to the origin of coordinates. Then, such NPT (described with vector of parameters \vec{P}_t of nonzero length) exists that will partially compensate the initial NPT (having parameters \vec{P}) of this image. This vector \vec{P}_t corresponding to compensating transform is parallel to the dipole of image and is directed towards its longer arm.*

The proof of this lemma is presented below in the Appendix 4. It is also shown there that the procedure of compensation of NPT of image by sequential application of compensating transforms having such vectors \vec{P}_t of parameters converges. Besides, it is explained in Appendix 4 that for enhancing of convergence, it is worth to determine in each iteration and compensate the constituent part A_3 of AP of image. The iterative procedure of compensation of NPT is formalized below in the Algorithm 2.

Algorithm 2

1. Determining and compensating the shifts \mathbf{A}_1 and \mathbf{A}_2 of image centroid with respect to the origin of coordinates and the anisotropic scaling \mathbf{A}_3 of image as it was described above in the Sects. 5.3 and 5.4.
2. Choosing the direction of vector \vec{P}_t of parameters of small compensating NPT (it is such direction of NPT that image centroid displacement ΔC_1 caused by NPT corresponding to \vec{P}_t differs maximally in its module from the displacement ΔC_2 corresponding to oppositely directed vector of NPT parameters). Estimating the values of ΔC_1 and ΔC_2 , corresponding to chosen vector \vec{P}_t .
3. Finishing the execution of algorithm, if $||\Delta C_1| - |\Delta C_2|| < < |\Delta C_1|$ and $||\Delta C_1| - |\Delta C_2|| < < |\Delta C_2|$. Otherwise, accomplishing a partially compensating NPT corresponding to the chosen vector \vec{P}_t of parameters and passing to the point 1 of the algorithm (the designation $|\cdot|$ corresponds here to absolute value).

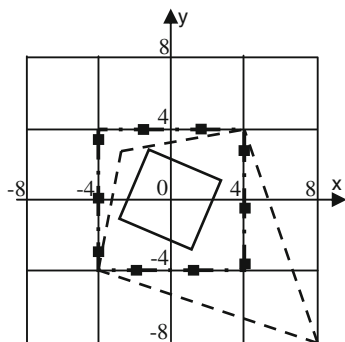
After finishing of iterative compensating Algorithm 2, it is possible to calculate the parameters of initial PT of inspected image by means of sequential summation of vectors of compensating transforms applied at each iteration, as it is also shown in Appendix 4.

It is not difficult to see that the parameters a_7 and a_8 of PT were also estimated on the base of analysis of object-independent kernel of image (see in Eq. 5.7).

5.8 Experimental Results

The described above Algorithm 2 was implemented in a software model and successfully tested for compensation of the NPT of diversified images. Two examples of normalization of images with respect to NPT and conjugated with it anisotropic scaling are shown in Figs. 5.1–5.2. It can be seen in Fig. 5.1 that a distorted by NPT square is brought to its template (symmetric) state as a result of such compensation. Figure 5.2 shows an arbitrary figure that had initially a normalized form and then

Fig. 5.1 Compensation of projective transform: *chain line with squares*—initial image, *dashed line*—distorted image, *solid line*—result of compensation of the NPT



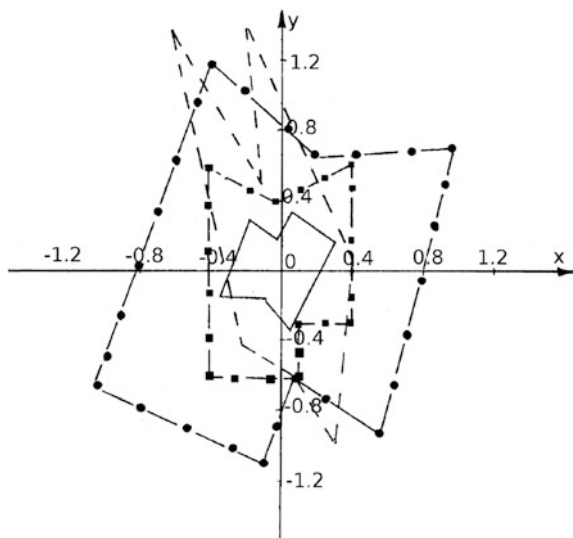


Fig. 5.2 Compensation of projective transform: *chain line with squares*—initial image, *dashed line*—distorted image, *solid line*—result of compensation of the NPT, *chain line with circles*—result of compensation of other NPT (a distorted image is not shown for this case)

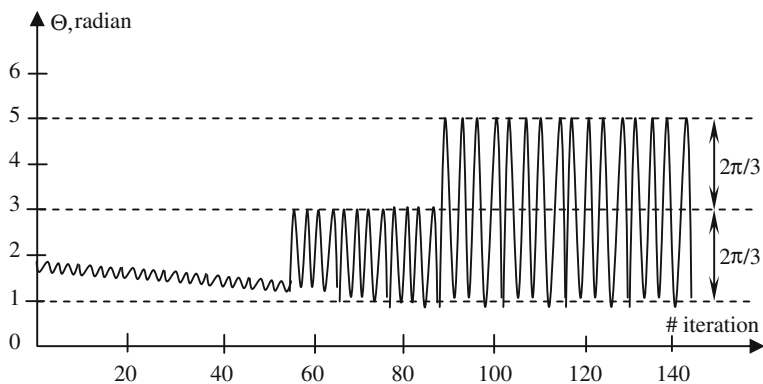


Fig. 5.3 Directions of maximum difference of displacements of image centroid at different iterations of compensation of its NPT

was distorted by two differing NPT. As a result of proposed iterative compensation, the distorted figure is brought in the both cases to the same shape accurate within isotropic scaling and rotation with respect to coordinate origin.

An example process for convergence of iterative algorithm of NPT compensation is presented in Fig. 5.3. Looking at this figure, it is interesting to note that there was a type of stable symmetric bifurcation and trifurcation in the directions of

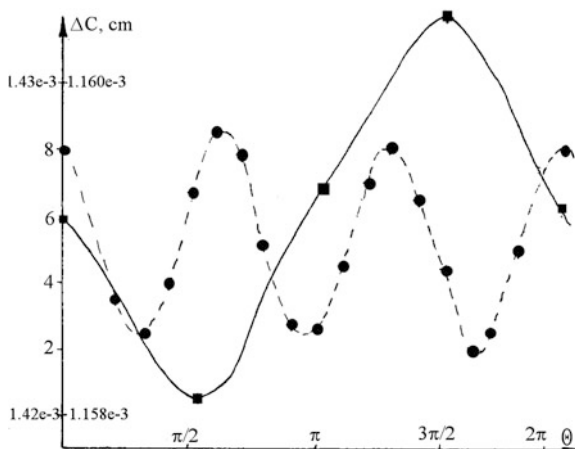


Fig. 5.4 Values of displacement of image centroid for different directions of vector of parameters of testing NPT: *solid line*—at initial iterations of compensation of NPT (*left scale ruler*); *dashed line*—at final iterations of compensation of NPT (*right scale ruler*)



Fig. 5.5 Examples of objects automatically recognized at manufacturing plants using the procedures of estimation and compensation of projective transforms proposed above

remaining error vector of NPT compensation: the remaining error vector accomplishes a kind of circulation around a target point. A graph of functional dependency of the centroid displacement values for a real nonnormalized image on the direction of testing NPT is shown in Fig. 5.4. It can be seen in Fig. 5.4 that initially (for long vectors of parameters of noncompensated NPT) this functional dependency has one period per full rotation cycle of vector of transformation parameters, and there are points of maximum and minimum displacement corresponding to opposite directions of vector of parameters. This case corresponds to the solid line of graph and to the left scale ruler at ordinate axis in Fig. 5.4. The graph depicted in Fig. 5.4 with dashed line corresponds to final iterations of compensation. The scale of this graph along the ordinate axis is described by the right ruler. A real trifurcation is seen in this case: the centroid displacement has three maxima separated by $2\pi/3$ per one rotation cycle of vector of transformation parameters. This second

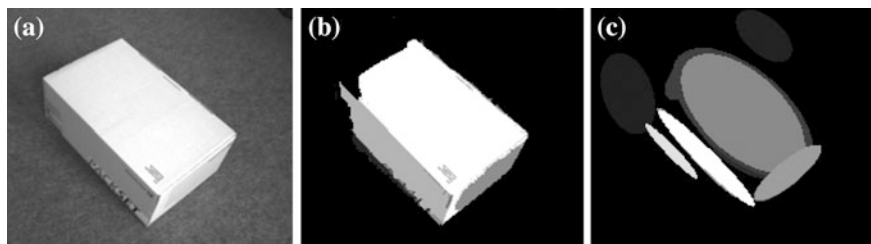


Fig. 5.6 Attention zones simulated using the proposed object-invariant image cores: **a** an image of 3D scene, **b** a locally incorrect result of image segmentation using traditional tools, **c** the attention zones built based on brightness features

graph shows much smaller magnitude of variation of centroid displacement corresponding to negligibly small residual uncompensated NPT. The described above experimental results proof the efficiency of proposed transformation estimation technique.

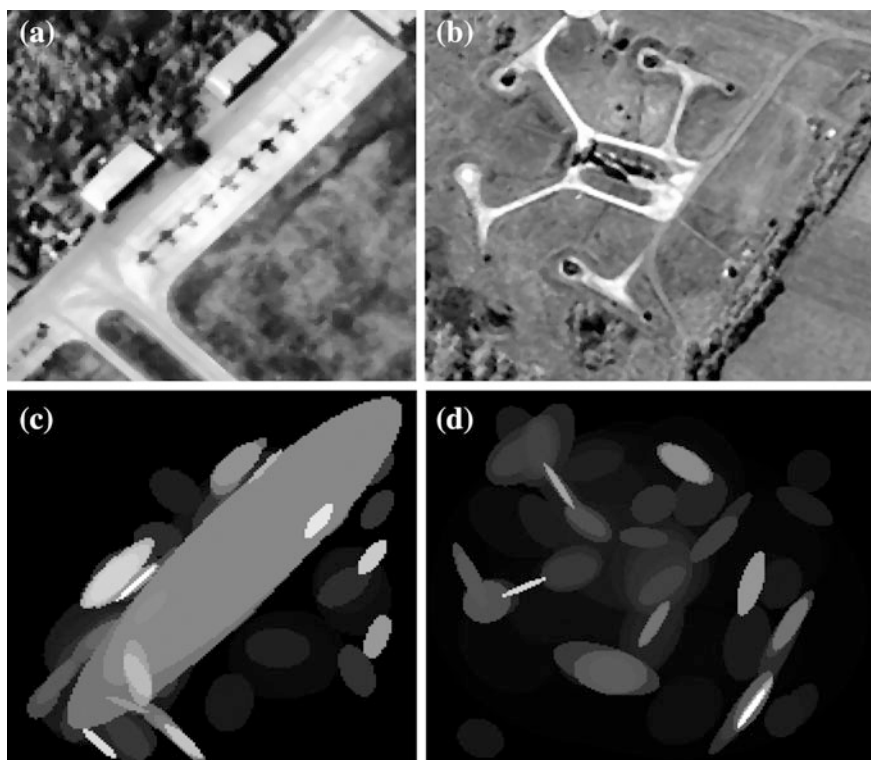


Fig. 5.7 Attention zones simulated using the proposed object-invariant image cores: **a, b** a pair of aerial photographs, **c, d** the attention zones built for these aerial photographs based on brightness features

Finally, it is interesting to note that a similar “method” of image analysis is applied in human vision, in which the changes of surface inclination and related to them NPT are determined from local velocity of texels recession [22].

As it could be forecasted, the proposed technique of image normalization turned out to be rather efficient just in the tasks of automatic recognition of images subjected to projective transforms. However, special observation conditions should be provided in such cases: the visible images of objects should be considered as almost plane, and the task of separation of object from background should be solved sufficiently reliably. Such conditions can be rather efficiently ensured at manufacturing plants [20, 25]. A pair of images of such entirely plain and non-plain objects that can be efficiently dealt with is shown in Fig. 5.5. The problem of separation of object from background using the technique of attention zones is discussed below.

The technique of object-invariant cores turned out to be so universal that it was successfully applied for texture cells description and texture recognition [27–30], for simulation of attention zones in human vision [31, 32], and even for detection of contour structural elements [25, 30], but these could be the themes of other research. The examples of elliptic attention zones detected and described using the proposed above parabolic object-invariant weighting functions are shown in Fig. 5.6 (for image of 3D object) and Fig. 5.7 (for aerial photographs). The depicted sizes of attention zones are specially decreased for avoiding their mutual screening. It can be seen in Fig. 5.6 that application of technique of such attention zones solves the problem of separation of undesirably fused adjoining image regions having similar values of brightness, color, or texture features. The technique based on attention zones was applied in [25, 30] for more efficient synthesis of multilevel hierarchical image structural descriptions, and the problem of separation of object from background in real images was also rather successfully solved there using the attention zones technique.

5.9 Similar Results Reached by Other Authors

It is interesting to note that the formulae for determining of parameters of anisotropic scaling could also be derived in other ways. In particular, Bradski gives in [33] a formula for determining the directions of axes of ellipse approximating a separated from background image. Bradski gives also the formulae for the length and width of approximated in such way image and refers to Freeman et al. [34] for more details of development of these formulae. The details of development of the formula for direction of ellipse axes are presented by Jähne in [35] (it is the 6th edition of his famous book on digital image processing, but the similar information could be also found earlier, e.g. in the 4th edition). The mentioned formulae for orientation, length, and width of approximating ellipse were developed using the technique of central moments. If the formulae for calculation of moments would be substituted into the formulae proposed in [33] by Bradski, then the Eqs. 5.18–5.19 developed in this chapter for the value and direction of anisotropic scaling could be

easily obtained by simple mathematical manipulations. Thus, the other researchers also independently succeeded in getting the formulae similar to those developed in this chapter for determining some parameters of the AT.

A fundamentally different method of estimating the parameters of anisotropic scaling of local image areas proposed by Baumberg [36] also should be mentioned. In contrast to the methods of determining the parameters of image shape presented in [33–35] and in this chapter, Baumberg estimates the parameters of anisotropic scaling of image based on the anisotropy of texture. Baumberg considers that image is normalized with respect to spatially anisotropic scaling, if its texture has statistical parameters isotropic with respect to direction in image plane. This approach could be accepted. However, it is not difficult to present real examples, in which the texture has isotropic parameters while an object having such texture (e.g. an elliptic object) is elongated in some direction. Thus, the method of estimating the parameters of anisotropic scaling proposed in [36] could be supposed to have restricted area of use.

5.10 Conclusion

The materials presented above demonstrate convincingly how the images of compact objects separated from background can be transformed to an object-invariant form, in which all generic-specific features are suppressed while the parameters of geometric transform are still preserved. Then, the parameters of projective transform can be estimated by means of comparison of such object-invariant description with a template form common for all classes of objects. Thus, the geometric transforms of images can be estimated based on object-invariant image cores and compensated, and then the image can be recognized by any pattern recognition techniques.

Acknowledgments This work was partially financially supported by the Government of Russian Federation, Grant 074-U01.

Appendices

Appendix 1. Proof of Lemma 2

Let us consider (for definiteness) the case, when the vector \vec{L} of dipole coincides in its direction with the abscissa axis, then the vector \vec{P} of NPT parameters applied in the past has the direction exactly opposite to direction of abscissa axis. Consideration of this particular case will not violate the generality of consideration because any other cases may be reduced to this one by rotation of coordinate system. Let M_1

be the mass (brightness) of left arm of dipole, let L_1 be the absolute value of abscissa of the left arm, let M_2 be the mass of right arm, and let L_2 be the absolute value of its abscissa. Then, $M_1 = N M_2$, $L_2 = N L_1$, $N > 1$. Let us apply to this dipole an additional “testing NPT” having the vector \vec{P}_t of parameters parallel to \vec{P} (antiparallel to abscissa axis). Let us designate $P_t = \|\vec{P}_t\|$. Then, as a result of the additional NPT, the lengths of dipole arms will get the transformed values L'_1 and L'_2 , that, according to Eqs. 5.28–5.29, are calculated using Eq. 5.32.

$$L'_1 = L_1/(1 + L_1 P_t) \quad L'_2 = L_2/(1 - L_2 P_t) \quad (5.32)$$

The increments ΔL_1 and ΔL_2 of coordinates of the dipole halves are then calculated by Eqs. 5.33–5.34.

$$\Delta L_1 = L_1 - L_1/(1 + L_1 P_t) = L_1^2 P_t/(1 + L_1 P_t) \quad (5.33)$$

$$\Delta L_2 = L_2/(1 - L_2 P_t) - L_2 = L_2^2 P_t/(1 - L_2 P_t) \quad (5.34)$$

Keeping in mind Eqs. 5.33–5.34, the absolute value ΔC_1 of displacement of abscissa of dipole centroid can be described with Eq. 5.35 that was derived using the substitutions $M_1 = N M_2$ and $L_2 = N L_1$.

$$\Delta C_1 = \frac{\Delta L_1 M_1 + \Delta L_2 M_2}{M_1 + M_2} = \frac{N M_2 L_1^2 P_t (1 + N)}{(1 + L_1 P_t)(1 - N L_1 P_t)(M_1 + M_2)} \quad (5.35)$$

The Eq. 5.36 showing the absolute value ΔC_2 of displacement of abscissa of centroid corresponding to opposite direction of vector \vec{P}_t was derived in similar way.

$$\Delta C_2 = \frac{N M_2 L_1^2 P_t (1 + N)}{(1 - L_1 P_t)(1 + N L_1 P_t)(M_1 + M_2)} \quad (5.36)$$

Let us compare in Eq. 5.37 the absolute values ΔC_1 and ΔC_2 of displacements of centroid corresponding to oppositely directed vectors \vec{P}_t (the symbol \sim means the inequality of unknown sign).

$$\frac{N M_2 L_1^2 P_t (1 + N)}{(1 + L_1 P_t)(1 - N L_1 P_t)(M_1 + M_2)} \sim \frac{N M_2 L_1^2 P_t (1 + N)}{(1 - L_1 P_t)(1 + N L_1 P_t)(M_1 + M_2)} \quad (5.37)$$

Let us assume $1 - N L_1 P_t > 0$ and $1 - L_1 P_t > 0$, which is needed for ensuring that the points of dipole would not move to infinity. Then, not violating the sign of inequality, Eq. 5.37 is reduced to Eq. 5.38, which gives $N \sim 1$ after simple standard mathematical manipulation.

$$(1-L_1P_t)(1+NL_1P_t) \sim (1+L_1P_t)(1-NL_1P_t) \quad (5.38)$$

However, it was assumed above that $N > 1$, thus the displacement of dipole centroid in the case, when the direction of \vec{P}_t is antiparallel to the direction of dipole (parallel to direction of \vec{P}), is larger in absolute value than the displacement corresponding to the case, when the direction of \vec{P}_t is parallel to direction of dipole (when the testing NPT partially compensates the initial NPT). Thus, Lemma 2 is proven.

Appendix 2. Proof of Lemma 3

As it was described in the condition of Lemma 3, there exists a dipole lying on abscissa axis, and it is balanced with respect to ordinate axis; the left (shorter) arm of this dipole corresponds to a single pixel or subpixel lying to the left of ordinate axis (at the left) and having the absolute value l of abscissa and mass (brightness) m . The right arm of dipole is N -fold longer than the left arm ($N > 1$) and corresponds to a multitude of two pixels or subpixels at the right having the differing masses M_1 and M_2 and abscissas L_1 and L_2 ($L_2 > L > L_1$), while the centroid of these pixels at the right has the abscissa $L = N \cdot l$ and aggregate mass $M = m/N$ concentrated at coordinate of centroid. Now, let some NPT be described with a vector \vec{P}_t of parameters that is antiparallel to abscissa axis (antiparallel to dipole), and let's designate $\|\vec{P}_t\| = P$, then the abscissas of pixels composing the dipole will be incremented as a result of such NPT by Eqs. 5.39–5.41.

$$\Delta l = l - l/(1 + l \cdot P) \quad (5.39)$$

$$\Delta L_1 = L_1/(1 - L_1 \cdot P) - L_1 \quad (5.40)$$

$$\Delta L_2 = L_2/(1 - L_2 \cdot P) - L_2 \quad (5.41)$$

The increment of abscissa C_1 of dipole centroid caused by such NPT is described by Eq. 5.42.

$$\Delta C_1 = (\Delta l \cdot m + \Delta L_1 \cdot M_1 + \Delta L_2 \cdot M_2)/(m + M_1 + M_2) \quad (5.42)$$

If the vector \vec{P}_t is parallel to abscissa axis (parallel to dipole), then the absolute values of increments of abscissas of dipole pixels resulting from such NPT will be described by Eqs. 5.43–5.45.

$$\Delta l' = l/(1 - l \cdot P) - l \quad (5.43)$$

$$\Delta L_1' = L_1 - L_1/(1 + L_1 \cdot P) \quad (5.44)$$

$$\Delta L'_2 = L_2 - L_2 / (1 + L_2 \cdot P) \quad (5.45)$$

The absolute value of increment of abscissa C_2 of dipole centroid caused by such NPT will be described by Eq. 5.46.

$$\Delta C_2 = (\Delta l' \cdot m + \Delta L'_1 \cdot M_1 + \Delta L'_2 \cdot M_2) / (m + M_1 + M_2) \quad (5.46)$$

For proving this lemma, it is necessary to compare the absolute values ΔC_1 and ΔC_2 of increments of abscissa of dipole centroid described by Eqs. 5.42 and 5.46. These formulae have equal nonnegative denominators, thus it would be sufficient to consider the relation of numerators $\Delta \hat{C}_1$ and $\Delta \hat{C}_2$ of Eqs. 5.42 and 5.46. According to Eqs. 5.39–5.42, one can receive Eq. 5.47.

$$\begin{aligned} \Delta \hat{C}_1 = & m \cdot l - m \cdot l / (1 + l \cdot P) + M_1 \cdot L_1 / (1 - L_1 \cdot P) \\ & - M_1 \cdot L_1 + M_2 \cdot L_2 / (1 - L_2 \cdot P) - M_2 \cdot L_2 \end{aligned} \quad (5.47)$$

Transforming Eq. 5.47 to common denominator and collecting the terms in the numerator and denominator of fraction will result in Eq. 5.48.

$$\Delta \hat{C}_1 = \frac{l^2 + l \cdot L_1 + l \cdot L_2 - l \cdot L_1 \cdot L_2 \cdot P - \frac{1}{N} \cdot L_1 \cdot L_2 - \frac{1}{N} \cdot l \cdot L_1 \cdot L_2 \cdot P}{(1 + l \cdot P - L_1 \cdot P - l \cdot L_1 \cdot P^2 - L_2 \cdot P - l \cdot L_2 \cdot P^2 + L_1 \cdot L_2 \cdot P^2 + l \cdot L_1 \cdot L_2 \cdot P^3) / (P \cdot m)} \quad (5.48)$$

The replacements shown in Eqs. 5.49–5.51 were applied in deducing of Eq. 5.48.

$$L_1 \cdot M_1 + L_2 \cdot M_2 = L \cdot M \quad (5.49)$$

$$L \cdot M = l \cdot m \quad (5.50)$$

$$M_1 + M_2 = M = m / N \quad (5.51)$$

According to Eqs. 5.43–5.46, Eq. 5.52 can be obtained.

$$\begin{aligned} \Delta \hat{C}_2 = & m \cdot l / (1 - l \cdot P) - m \cdot l + M_1 \cdot L_1 \\ & - M_1 \cdot L_1 / (1 + L_1 \cdot P) + M_2 \cdot L_2 - M_2 \cdot L_2 / (1 + L_2 \cdot P) \end{aligned} \quad (5.52)$$

Transforming Eq. 5.52 to common denominator and collecting the terms in the numerator and denominator of fraction will result in Eq. 5.53.

$$\Delta \hat{C}_2 = \frac{l^2 + l \cdot L_1 + l \cdot L_2 + l \cdot L_1 \cdot L_2 \cdot P - \frac{1}{N} \cdot L_1 \cdot L_2 + \frac{1}{N} \cdot l \cdot L_1 \cdot L_2 \cdot P}{(1 - l \cdot P + L_1 \cdot P - l \cdot L_1 \cdot P^2 + L_2 \cdot P - l \cdot L_2 \cdot P^2 + L_1 \cdot L_2 \cdot P^2 - l \cdot L_1 \cdot L_2 \cdot P^3) / (P \cdot m)} \quad (5.53)$$

The replacements presented in Eqs. 5.49–5.51 also were applied in deducing of Eq. 5.53. Let us analyze the inequality shown in Eq. 5.54.

$$\Delta\hat{C}_1 \sim \Delta\hat{C}_2 \quad (5.54)$$

Here, the symbol \sim of unknown inequality connects the absolute values of centroid displacements calculated according to Eqs. 5.39–5.46 for the nonlinear projective transforms corresponding to vectors \vec{P}_i of parameters antiparallel (at the left) and parallel (at the right) to abscissa axis respectively. Let us replace $\Delta\hat{C}_1$ and $\Delta\hat{C}_2$ in Eq. 5.54 with their values shown in Eqs. 5.48 and 5.53. Then, the both sides of resulting inequality are divided by positive factor $P \cdot m$ and transformed to common denominator by means of multiplying them by the denominators of Eqs. 5.48 and 5.53. The denominators of fractions in Eqs. 5.39–5.41, 5.43–5.45, 5.47, and 5.52 are positive (they may not approach zero, which would result in infinity in transformed values), thus the denominators of Eqs. 5.48 and 5.53 are also positive. Thus, the mentioned above transformation to common denominator followed by omitting of equal denominators of the both sides of inequality will not change the sign of inequality shown in Eq. 5.54 and will transform it to the form mentioned below:

$$\begin{aligned} & l \cdot L_1 \cdot L_2 \cdot P - l^3 \cdot P + l \cdot L_1^2 \cdot P - \frac{1}{N} L_1^2 \cdot L_2 \cdot P + \frac{1}{N} l^2 \cdot L_1^2 \cdot L_2 \cdot P^3 \\ & + l \cdot L_2^2 \cdot P - \frac{1}{N} L_1 \cdot L_2^2 \cdot P + \frac{1}{N} l^2 \cdot L_1 \cdot L_2^2 \cdot P^3 - l \cdot L_1^2 \cdot L_2^2 \cdot P^3 - l^3 \cdot L_1 \cdot L_2 \cdot P^3 \\ & \sim - l \cdot L_1 \cdot L_2 \cdot P + l^3 \cdot P - l \cdot L_1^2 \cdot P + \frac{1}{N} L_1^2 \cdot L_2 \cdot P - \frac{1}{N} l^2 \cdot L_1^2 \cdot L_2 \cdot P^3 \\ & - l \cdot L_2^2 \cdot P + \frac{1}{N} L_1 \cdot L_2^2 \cdot P - \frac{1}{N} l^2 \cdot L_1 \cdot L_2^2 \cdot P^3 + l \cdot L_1^2 \cdot L_2^2 \cdot P^3 + l^3 \cdot L_1 \cdot L_2 \cdot P^3. \end{aligned}$$

Finally, collecting the terms and dividing by positive P will result in Eq. 5.55.

$$\begin{aligned} & l \cdot L_1 \cdot L_2 + l \cdot L_1^2 + l \cdot L_2^2 + \frac{1}{N} l^2 \cdot L_1 \cdot L_2 \cdot P^2 + \frac{1}{N} l^2 \cdot L_1^2 \cdot L_2 \cdot P^2 \\ & \sim l^3 + \frac{1}{N} L_1^2 \cdot L_2 + \frac{1}{N} L_1 \cdot L_2^2 + l \cdot L_1^2 \cdot L_2 \cdot P^2 + l^3 \cdot L_1 \cdot L_2 \cdot P^2 \end{aligned} \quad (5.55)$$

Let us choose $P \ll 1$. This would enable neglecting the terms containing P^2 in Eq. 5.55, which corresponds in general to the described in this material strategy of iterative compensation of NPT by means of applying the testing-compensating projective transforms having small in absolute value vectors of parameters. Then Eq. 5.55 will be transformed to Eq. 5.56.

$$l \cdot L_1 \cdot L_2 + l \cdot L_1^2 + l \cdot L_2^2 \sim l^3 + \frac{1}{N} L_1^2 \cdot L_2 + \frac{1}{N} L_1 \cdot L_2^2 \quad (5.56)$$

First, let us consider the partial case shown in Eq. 5.57 also satisfying the conditions of Lemma 3.

$$L_2 > L = N \cdot l > L_1 \geq l \quad (5.57)$$

Let us fix $l = 1$. This will not violate the generality of proof because it is possible to pass to arbitrary permitted values of l by simple scaling of coordinate system. This results in Eq. 5.58.

$$L_2 > L = N \cdot 1 > L_1 \geq 1 \quad (5.58)$$

Thus, Eq. 5.56 will be transformed to Eq. 5.59.

$$L_1 \cdot L_2 + L_1^2 + L_2^2 \sim 1 + \frac{1}{N} L_1^2 \cdot L_2 + \frac{1}{N} L_1 \cdot L_2^2 \quad (5.59)$$

Let us compare the respective terms of the left and right parts of inequality presented in Eq. 5.59:

$$L_1 \cdot L_2 > (1/N) L_1^2 \cdot L_2 = (L_1/N) L_1 \cdot L_2 \quad (5.60)$$

because $L_1/N < 1$ according to Eq. 5.58;

$$L_1^2 \geq 1 \quad (5.61)$$

because $L_1 \geq 1$ according to Eq. 5.58;

$$L_2^2 > (1/N) \cdot L_1 \cdot L_2^2 = (L_1/N) \cdot L_2^2 \quad (5.62)$$

because $L_1/N < 1$ according to Eq. 5.58.

Thus, according to Eqs. 5.60–5.62 and the assumption presented in Eq. 5.57, such value $P \ll 1$ for norm of vector \vec{P}_t of NPT parameters exists that the left part of Eq. 5.56 would be larger than its right part. In other words, according to Eq. 5.54 the centroid displacement corresponding to vector \vec{P}_t antiparallel to abscissa axis will be larger in absolute value than the displacement corresponding to vector \vec{P}_t parallel to abscissa axis. Thus the Lemma 3 is proven for the particular case specified in Eq. 5.57.

Let us consider now the alternative case, when the assumption presented in Eq. 5.57 is not true. This corresponds to Eq. 5.63.

$$L_1 < l \quad (5.63)$$

Let us fix $L_1 = 1$, which will give Eq. 5.64.

$$l > 1 \quad (5.64)$$

This will not violate the generality of proof because it is possible to pass to arbitrary permitted values of L_1 by simple scaling of coordinate system. Then, Eq. 5.56 will get the form of Eq. 5.65.

$$l \cdot L_2 + l + l \cdot L_2^2 \sim l^3 + \frac{1}{N}L_2 + \frac{1}{N}L_2^2 \quad (5.65)$$

Collecting the terms in Eq. 5.65 and dividing the both parts of Eq. 5.65 by positive $(l - 1/N)$ will give Eq. 5.66. The term $(l - 1/N)$ is positive because $l > 1$ according to Eq. 5.64 and $N > 1$ according to conditions of lemma.

$$L_2^2 + L_2 + (l - l^3)/(l - 1/N) \sim 0 \quad (5.66)$$

The last term in the left part of Eq. 5.66 can be divided into two fractions, which will give Eq. 5.67.

$$L_2^2 + L_2 + 1 - (l^3 - 1/N)/(l - 1/N) \sim 0 \quad (5.67)$$

Let's transform the last term of sum in the left part of Eq. 5.67 using the rule $x^3 - a^3 = (x - a) \cdot (x^2 + ax + a^2)$ in the following way:

$$-\frac{l^3 - 1/N}{l - 1/N} = -\frac{l^3 - 1/N^3}{l - 1/N} - \frac{1/N^3 - 1/N}{l - 1/N} = -\left(l^2 + l/N + 1/N^2\right) - \frac{1 - N^2}{l \cdot N^3 - N^2}.$$

Finally, this will result in the following Inequality (Eq. 5.68).

$$[L_2^2 + L_2 + 1] - \left(l^2 + l/N + 1/N^2\right) + \frac{N^2 - 1}{l \cdot N^3 - N^2} \sim 0 \quad (5.68)$$

According to conditions of lemma, $N > 1$ and $L_2 > L = l \cdot N > l$, while $l > 1$ according to Eq. 5.64. Thus the last fraction in the left part of Eq. 5.68 is positive, and the contents of round brackets and square brackets in the left part of Eq. 5.68 are positive as well. Thus, Eq. 5.68 can be recorded in the form:

$$[L_2^2 + L_2 + 1] - \left(l^2 + l/N + 1/N^2\right) + \frac{N^2 - 1}{l \cdot N^3 - N^2} > 0.$$

Keeping in mind Eq. 5.54, it allows claiming (also in the case of alternative assumption presented in Eq. 5.63) that the centroid displacement related to vector \vec{P}_t antiparallel to abscissa axis is larger in absolute value than the displacement related to \vec{P}_t parallel to abscissa axis. Therefore, Lemma 3 is proven in general.

Appendix 3. Proof of Lemma 4

It is easy to see that the conditions of this lemma correspond to the conditions of Lemma 3 proven above, but the following was changed:

1. The dipole arm composed of two pixels is situated now in the domain of negative abscissas, and the arm composed of a single pixel is situated in the domain of positive abscissas.
2. The dipole arm composed of two pixels became shorter than the dipole arm composed of a single pixel because the value of N became smaller than 1.

It follows from these changes that the dipole orientation with respect to the direction of vector \vec{P}_t (of parameters of NPT) considered in Lemma 3 is not changed. Thus, if the vector \vec{P}_t is antiparallel to abscissa axis, then such NPT will decrease the absolute values of abscissas of the pair of pixels situated to the left of ordinate axis, and it will increase the abscissa of pixel situated to the right of ordinate axis as it was considered in the materials related to Lemma 3. The absolute value ΔC_2 of displacement of dipole centroid described by Eqs. 5.43–5.46 in the materials of Lemma 3 corresponds to this NPT. Thus, it should be proven that the absolute value ΔC_1 of centroid displacement related to oppositely directed vector \vec{P}_t of NPT parameters would be smaller than ΔC_2 . The value of ΔC_1 is described by Eqs. 5.39–5.42 in the materials of Lemma 3. In other words, it should be proven that Eq. 5.69 is correct.

$$\Delta C_1 < \Delta C_2 \quad (5.69)$$

Thus, according to the materials of proof of Lemma 3, it should be proven here that $\Delta \hat{C}_1 < \Delta \hat{C}_2$, because the values $\Delta \hat{C}_1$ and $\Delta \hat{C}_2$ were derived from ΔC_1 and ΔC_2 in Eqs. 5.48 and 5.53 using multiplication by positive values of denominators of respective fractions. Therefore, the sign of inequality presented in Eq. 5.69 should not be changed.

In the proof of Lemma 3, the inequality presented in Eq. 5.54 ($\Delta \hat{C}_1 \sim \Delta \hat{C}_2$) was transformed to the form shown in Eq. 5.56, and no multiplication or dividing by negative values was applied in the course of such transformation, thus the sign of inequality also was not changed. For convenience of further analysis, the inequality presented in Eq. 5.56 is repeated below in Eq. 5.70. Let us consider Eq. 5.70 within the frameworks of this proof: the symbol \sim should be replaced here by the symbol $<$.

$$l \cdot L_1 \cdot L_2 + l \cdot L_1^2 + l \cdot L_2^2 \sim l^3 + \frac{1}{N} L_1^2 \cdot L_2 + \frac{1}{N} L_1 \cdot L_2^2 \quad (5.70)$$

According to conditions of lemma being proven, Eqs. 5.71–5.72 should be correct.

$$L_2 > L > L_1 \quad (5.71)$$

$$l = L/N > L \quad N < 1 \quad (5.72)$$

Thus, L_1 has the smallest in absolute value abscissa as compared with the abscissas of other pixels composing dipole. Let us fix $L_1 = 1$. This will not violate generality of proof because it is possible to pass to other values of L_1 by simple scaling of coordinate system. Then Eq. 5.70 will get the following form shown in Eq. 5.73.

$$l \cdot L_2 + l + l \cdot L_2^2 \sim l^3 + \frac{1}{N}L_2 + \frac{1}{N}L_2^2 \quad (5.73)$$

Let us divide the both parts of Eq. 5.73 by the positive lL_2^2 , which will result in Eq. 5.74.

$$\frac{1}{L_2} + \frac{1}{L_2^2} + 1 \sim \frac{l^2}{L_2^2} + \frac{1}{N \cdot L_2 \cdot l} + \frac{1}{N \cdot l} \quad (5.74)$$

Comparison of respective terms of the left and right parts of Eq. 5.74 will give Eqs. 5.75–5.77:

$$1/L_2 < 1/(N \cdot l) \quad (5.75)$$

because $N \cdot l = L < L_2$ according to Eqs. 5.71 and 5.72;

$$1/L_2^2 < 1/(N \cdot L_2 \cdot l) \quad (5.76)$$

because $N \cdot l = L < L_2$ according to Eqs. 5.71 and 5.72;

$$1 \leq l^2/L_2^2, \quad (5.77)$$

if

$$L_2 \leq l. \quad (5.78)$$

According to Eqs. 5.75–5.77, the respective terms of right part of Eq. 5.74 are larger than the respective parts of its left part, thus Eq. 5.74 can now be recorded in the following form:

$$\frac{1}{L_2} + \frac{1}{L_2^2} + 1 < \frac{l^2}{L_2^2} + \frac{1}{N \cdot L_2 \cdot l} + \frac{1}{N \cdot l}.$$

This means that $\Delta\hat{C}_1 < \Delta\hat{C}_2$ in Eq. 5.54, and $\Delta C_1 < \Delta C_2$, where ΔC_1 and ΔC_2 are calculated according to Eqs. 5.42 and 5.46.

Thus, the demanded inequality presented in Eq. 5.69 is proven. Therefore, there exists an arbitrarily small in absolute value vector \vec{P}_t of NPT parameters formed according to Eqs. 5.28 and 5.29, oriented along the abscissa axis, and having the following properties: if \vec{P}_t is parallel to abscissa axis, then the displacement of dipole centroid caused by such NPT is smaller in absolute value than the centroid displacement caused by the NPT corresponding to vector \vec{P}_t antiparallel to abscissa axis. Therefore, Lemma 4 is proven.

Remark The correctness of Lemmas 3 and 4 guarantees, that using the Algorithm 1, the image of any object having restricted definitional domain and separated from background can be presented as a set of dipoles balanced with respect to image centroid and having the same ratios of arm lengths. This would ensure the convergence of iterative procedure for NPT compensation of image of such object described above in Algorithm 2. However, Lemma 4 was proven using the assumption (presented in Eq. 5.78) stipulating that the module of abscissa of none of pixels composing the shorter arm of dipole may exceed the module of abscissa of the pixel corresponding to longer arm of dipole. For solving this problem, the use of such assumption could be introduced into respective part of Algorithm 1 by choosing for mutual balancing only those pixels, for which $L_2 \leq l$. Such pairs of pixels could be easily found, if the ratio N of lengths of dipole arms would substantially exceed 1, i.e. if the length of vector of parameters of still not compensated NPT would be much larger than 1. The cases of violation of assumption presented in Eq. 5.78, in which $L_2 > l$ could be considered as outliers corresponding to pixels situated outside the borders of compact image object having restricted definitional domain. However, when the value of N approaches 1 (i.e. in the case of almost entire compensation of the NPT), the assumption presented in Eq. 5.78 becomes awkward for ever larger and larger numbers of pixels. This results in increase of errors of estimation of direction of vector \vec{P}_t of parameters of partially compensating NPT applied in the iterative compensating procedure presented in Algorithm 2. Finally, for some small lengths of vector \vec{P} of parameters of still non-compensated NPT, the further convergence of iterative compensating procedure becomes impossible. It was also confirmed by the results of computer simulation described above in Sect. 5.8. At some steps of iterative compensation, the length of vector \vec{P} of parameters of still non-compensated NPT stopped shortening. However, in this case, the remaining length of \vec{P} and the applied lengths of compensating vector \vec{P}_t had already become commensurable. The software modeling described above showed that the value of non-compensated NPT was so small that the displacements of image centroid corresponding to such lengths of \vec{P} and \vec{P}_t were three orders smaller than the values of coordinates of image pixels (three orders smaller than geometric sizes of image of object). In such case, the absolute difference of displacement of centroid caused by partially compensating NPT (described by \vec{P}_t) and displacement caused by the NPT with oppositely directed vector of parameters was six orders smaller than image size. It suggests that the real errors of estimation of parameters and compensation of NPT caused by use of assumption presented in Eq. 5.78 can be negligibly small.

Appendix 4. Proof of Lemma 5

First, let us consider how should be built a vector of parameters of total PT being the result of a series of sequentially executed projective transforms. Let $\vec{P}_1, \vec{P}_2, \vec{P}_3$ be the vectors of parameters of three NPT, and let \mathbf{A} and \mathbf{B} be the matrices of two AT. Then, the sense of Eqs. 5.28–5.29 can be expressed by a single Eq. 5.79.

$$\vec{X}' = (\mathbf{A}\vec{X}) / (\vec{X} \cdot \vec{P} + 1) \tag{5.79}$$

If the affine component of PT is zero, the transform described by Eq. 5.79 is reduced to NPT described in Eq. 5.80.

$$\vec{X}' = \vec{X} / (\vec{X} \cdot \vec{P} + 1) \tag{5.80}$$

A composite PT resulting from sequential execution of first NPT (described by vector \vec{P}_1 of parameters), the AT (described by matrix \mathbf{A}), and second NPT (described by vector \vec{P}_2 of parameters) can be expressed by Eq. 5.81.

$$\vec{X}' = [(\mathbf{A}\vec{X}) / (\vec{X} \cdot \vec{P}_1 + 1)] / [(\mathbf{A}\vec{X}) / (\vec{X} \cdot \vec{P}_1 + 1) \cdot \vec{P}_2 + 1] = (\mathbf{B}\vec{X}) / (\vec{X} \cdot \vec{P}_3 + 1) \tag{5.81}$$

Simple transformation of Eq. 5.81 shows that the matrix \mathbf{B} of affine component of resulting composite PT is equal to the matrix \mathbf{A} of initial AT, while the vector \vec{P}_3 of parameters of resulting composite NPT can be calculated by Eq. 5.82.

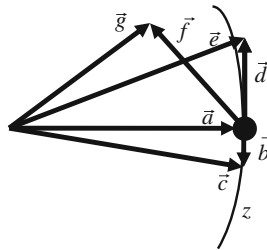


Fig. 5.8 Different variants of summation of vectors \vec{P} and \vec{P}_i of parameters of initial and compensating NPT: \vec{a} is a vector of parameters of initial NPT, z is a circle of radius $||\vec{a}||$ circumscribed by rotating vector \vec{a} , $\vec{b}, \vec{c}, \vec{f}$ are different positions of vector of parameters of NPT compensating (infinitely short vector \vec{b} orthogonal to vector \vec{a} and lying on the circle z ; long vector \vec{d} orthogonal to vector \vec{a} and thus not laying on the circle z), $\vec{c}, \vec{e}, \vec{g}$ are vectors of composite NPT resulting from summation of vector \vec{a} with vectors $\vec{b}, \vec{d}, \vec{f}$ of parameters of compensating NPT

$$\vec{P}_3 = \mathbf{A}\vec{P}_2 + \vec{P}_1 \quad (5.82)$$

If the matrix \mathbf{A} of affine transform would be neglected, it would be seen from Eq. 5.82 that the vector of parameters of resulting composite NPT is a sum of vectors of parameters of two sequentially executed NPT.

A result of execution of compensating NPT is shown in Fig. 5.8 for the cases, when the vector \vec{P}_t of parameters of compensating NPT has nonpositive projection upon the vector \vec{P} of parameters of initial NPT. It can be seen that the vector of parameters of resulting composite NPT becomes shorter, if the vector of parameters of compensating NPT lies inside the circle z . The closer is the vector \vec{P}_t to normal of vector \vec{P} , the shorter should be \vec{P}_t for ensuring correctness of inequality $\|\vec{P} + \vec{P}_t\| < \|\vec{P}\|$. If the vectors \vec{P} и \vec{P}_t are orthogonal, the inequality $\|\vec{P} + \vec{P}_t\| < \|\vec{P}\|$ is satisfied when $\|\vec{P}_t\| \rightarrow 0$.

As it can be concluded from the Eqs. 5.28–5.29, it is theoretically possible that the direction of dipole could be almost orthogonal to vector \vec{P} . It could happen, if the image size along vector \vec{P} would be much smaller then the size in the direction orthogonal to \vec{P} . However, the vectors \vec{P}_t and \vec{P} are never orthogonal, and the projection of \vec{P}_t upon \vec{P} is nonpositive. Therefore, a sufficiently short (but not infinitely short) vector \vec{P}_t can be found, for which $\|\vec{P} + \vec{P}_t\| < \|\vec{P}\|$. Thus, if a compensating NPT having the vector of parameters \vec{P}_t chosen in the way described above was applied, the resulting composite NPT would have vector of parameters shorter than vector \vec{P} corresponding to initial NPT, which means that the initial NPT would be partially compensated. Therefore, Lemma 5 is proven.

At the same time, if \vec{P}_t would be almost orthogonal to \vec{P} , the velocity of decrease of $\|\vec{P} + \vec{P}_t\|$ and the convergence of the whole iterative algorithm of NPT compensation would be very low. For avoiding such situations, it is recommended to compensate substantial elongation of object image in any direction at each iteration of NPT compensation. Only then, the direction of compensating vector \vec{P}_t may be estimated. Such compensation of elongation does not contradict the rules of developed above procedure of PT compensation because the compensation of elongation is an integral part of AT compensation described above in Sects. 5.3–5.6. Besides, the direction of dipole and related to it direction of vector \vec{P}_t are estimated after such compensation of AP, thus the compensation of AP will not disturb the correctness of estimating the direction of vector \vec{P}_t of parameters of compensating NPT. Execution of compensating NPT results in redistribution of pixels with respect to image centroid, which makes the dipole of image more symmetric. On the contrary, the compensating AT does not result in redistribution of pixels with respect to centroid, thus it cannot result in increase of asymmetry of dipole. Therefore, the AT executed at each iteration of Algorithm 2 may be

neglected as it was proposed above in the course of deducing of Eq. 5.82. The partial compensation of initial NPT at each iteration ensures the convergence of iterative compensation of NPT of image described by Algorithm 2 in Sect. 5.7.

References

1. Shapiro LG, Stockman GC (2001) Computer vision. Prentice Hall, Upper Saddle River
2. Haralick RM (1974) A measure of circularity of digital figures. *IEEE Trans Syst Man Cybern SMC-4*:394–396
3. Casasent DP, Barnard E (1990) Adaptive clustering optical neural net. *Appl Opt* 29 (17):2603–2615
4. Casasent D, Psaltis D (1978) Deformation-invariant, space-variant optical pattern recognition. *Prog Opt XVI*:291–356
5. Qin-sheng C, Defrise M, Deconinck F (1994) Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition. *IEEE Trans Pattern Anal Mach Intell* 16(12):1156–1167
6. Maitra S (1979) Moment invariants. *Proc IEEE* 67(4):697–699
7. Freeman H (1974) Computer processing of line drawing images. *Comput Surv* 6:57–97
8. Gonzalez RC, Woods RE (2002) Digital image processing. Prentice Hall, Upper Saddle River
9. Ling H, Jacobs D (2007) Shape classification using the inner distance. *IEEE Trans Pattern Anal Mach Intell* 29(2):286–299
10. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
11. Morel JM, Yu G (2009) ASIFT: a new framework for fully affine invariant image comparison. *SIAM J Imaging Sci* 2(2):438–469
12. Murillo AS, Guerrero JJ, Sagüés C (2007) SURF features for efficient robot localization with omnidirectional images. In: *IEEE international conference on robotics and automation*, pp 3901–3907
13. Averkin AV, Potapov AS, Lutsev VR (2010) Construction of systems of local invariant image indicators based on the Fourier-Mellin transform. *J Opt Technol* 77(1):28–32
14. Viola P, Jones MJ (2001) Rapid object detection using a boosted cascade of simple features. In: *Proceedings of IEEE computer vision and pattern recognition conference*, pp I-501–I-518
15. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *IEEE conference on computer vision and pattern recognition*, pp 886–893
16. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part based models. *IEEE Trans Pattern Anal Mach Intell* 32 (9):1627–1645
17. Lutsiv VR, Malyshev IA, Pepelka VA, Potapov AS (2002) The target independent algorithms for description and structural matching of aerospace photographs. *SPIE Proc* 4741:351–362
18. Lutsiv VR, Malyshev IA, Potapov AS (2003) Hierarchical structural matching algorithms for registration of aerospace images. *SPIE Proc* 5238:164–175
19. Lutsiv V, Potapov A, Novikova T, Lapina N (2005) Hierarchical 3D structural matching in the aerospace photographs and indoor scenes. *SPIE Proc* 5807:455–466
20. Lutsiv V (1985) Methods and tools of control of industrial equipment on the base of video-information. Ph.D. thesis, Institute of Aircraft Instrumentation (in Russian)
21. Lutsiv V, Malyshev I (2013) Image structural analysis in the tasks of automatic navigation of unmanned vehicles and inspection of earth surface. *Proc SPIE* 8897. doi:[10.1117/12.2028840](https://doi.org/10.1117/12.2028840)

22. Marr D (1982) *Vision: a computational investigation into the human representation and processing of visual information*. W.H. Freeman and Co., New York
23. Erosch IL (1984) The tasks of pattern recognition in robot systems. In: *Problems and prospects of optical image processing methods*. Physic-technical Institute of Soviet Academy of Sciences (in Russian), pp 75–78
24. Erosch IL (1981) Application of Krestenson transform for determination of object position parameters from plain projections. *Eng Cybern* 3:46–52 (in Russian)
25. Lutsiv V (2012) An object-independent approach to image structural analysis. DPhil thesis, Saint Petersburg University of Aerospace Instrumentation, Russia (in Russian)
26. Lutsiv V (2009) Method of iteratively compensating projective image distortions. *J Opt Technol* 76(7):417–422
27. Lutsiv VR, Dolinov DS, Zhrebko AK, Novikova TA (1997) Using artificial neural networks in image processing problems. *J Opt Technol* 64(2):112–118
28. Lutsiv V, Malyshev I, Novikova T (2004) Hierarchical approaches to analysis of natural textures. *SPIE Proc* 5426:144–154
29. Lutsiv VR (2007) The application of generalized reference functions in natural and artificial visual systems. *J Opt Technol* 74(11):759–763
30. Lutsiv VR (2008) Object-independent approach to the structural analysis of images. *J Opt Technol* 75(11):708–714
31. Lutsiv VR (2007) Modelling the attention zones in problems involving the automatic decomposition and structural analysis of images. *J Opt Technol* 74(4):274–281
32. Lutsiv VR, Novikova TA (2008) Modeling attention zones on the basis of an analysis of local features of the image texture. *J Opt Technol* 75(7):449–456
33. Bradski R (1998) Computer vision face tracking for use in perceptual user interface. *Intel Technology J Q2'98*:706–740
34. Freeman WT, Tanaka K, Ohta J, Kyuma K (1996) Computer vision for computer games. In: *2nd International conference on automatic face and gesture recognition*, pp 100–105
35. Jähne B (2005) *Digital image processing*. Springer, Berlin-Heidelberg
36. Baumberg A (2000) Reliable feature matching across widely separated views. In: *IEEE conference on computer vision and pattern recognition*, vol 1, pp 774–781

Chapter 6

A Way of Energy Analysis for Image and Video Sequence Processing

Andrei Bogoslovsky and Irina Zhigulina

Abstract Usually the object movements are determined by the analysis of interframe difference in video signals. It is the simplest universal method. However, it does not provide the intelligent processing, especially in the case of extremely low luminance. The interframe differences of energy and phase-energy spectrums are considered as an alternative way. The phase-energy spectrum is a product of partial derivatives in spatial phase-frequency spectrum over their spatial frequencies. It provides the detailed information about motion in finite frames. Moreover, the edges in an image have a significant role. The modeling of interframe differences of frequency responses is based on the analysis of pixels locating near the “moving” boundaries. This increases a probability of movement’s detection. A distortion of moving object’s shape, movement’s characteristics, and a quantity of moving objects are defined from the analysis of types of interframe differences. The interframe differences of frequency responses always lead to the best results than the differences of video signals in spatial domain. The changes of the energetic indexes in static images determine the efficiency function as a dependence of output and input energies of 2D filter. This function is defined on a whole set of impulse responses of a filter. The efficiency function is a positively certain quadratic form with certain coefficients. These coefficients are obtained as a result of energy spectrum decomposition of input frame into 2D Fourier series over the cosines. The analysis of stationary points and also their efficiency function allow to synthesize the optimum and the quasi-optimum 2D filters. The proposed way of energy analysis provides some novel possibilities, for example, a detection of visual objects with extremely small contrast.

Keywords Video sequences · Interframe differences · Phase-energy spectrum · Energy spectrum · Optimal image processing

A. Bogoslovsky · I. Zhigulina (✉)
Tambov State Technical University, 106 Sovetskaya st, Tambov 392000, Russian Federation
e-mail: irazhigulina@gmail.com

A. Bogoslovsky
e-mail: p-digim@mail.ru

6.1 Introduction

A biological sense of vision including a human vision is characterized by most impressive results. A vision system processes and perceives a huge array of real-time information [1–7]. However, the system's dimensions, the mass and energy consumption are such that its owner as a mobile being can solve a great number of other problems. A human vision system is not revealed yet to a full degree for a time being [8, 9]. Its operational efficiency is ensured both by training and by the fact that a brain is a part of body, i.e. the brain also uses other sensory organs while viewing and perceiving the real life scenes [10]. One of the determining factors is the operation arrangement of a visual system, and, first of all, the hierarchical pattern of processing. This pattern is such that an initial processing is performed on eye retina, and the ready product of such processing is received by a visual cortex. A hierarchical pattern of processing may be considered as a general computational resource for optimization.

Unfortunately, computer vision of modern control, navigation and safety systems is yet long way off by its performance from a biological sense of vision. Probably, one of such reasons is a hierarchy unavailability of computer vision. A preliminary data processing stage is not singled out from the general computer vision system [11–15]. However, if the preliminary data processing system is available, it usually solves the auxiliary problems serving for image quality improvement, that is, noise impact reduction, spatial profiteering, etc. At the same time, an eye retina identifies additionally the basic spatial and temporal features of visual objects. Such identifying has to perform a great number of arithmetical and logical operations with nonnegative signals by means of multiple layers of different cell types, these layers being mutually associated by many spatial connections. Thus, a preliminary data processing in the biological sense of vision is intelligent one, and an eye retina is considered to be a brain outside part [16, 17]. Apart from hierarchy and intelligence, the biological sense of vision has a feature common for all living systems: any process is repeatedly duplicated by various methods [18–20].

Video sequences can be represented as a set of static images (frames), which are processed by a quasioptimal spatial filtering. On the other hand, these sequences provide addition information about frame-to-frame differences. These algorithms can be applied as a preliminary processing of images. Moreover, they can shed light on a retina operation conception [21–25].

An analysis of the most important universal feature, i.e. signal energy, lies at the basis of all such algorithms. During filtering of static images, the energy ratio of 2D output-input signals is maximized, and for definition of movement and location characteristics, either the energy spectra differences of two frames' lines and columns, either their phase-energetic spectra differences are analyzed [26–30].

Due to small-scale shifts between frames, a movement existence in the image can be judged by the energy spectrum. Sometimes information of phases from an energy spectrum has not been examined [31–33]. However, this statement is correct only in case, when image finiteness in spatial domain does not taken into account.

The problem statement is located in Sect. 6.2. The frequency response is discussed in Sect. 6.3 while as the frequency responses difference model is represented in Sect. 6.4. The following sections provide the special cases (Sect. 6.5), the Frequency Responses of InterFrame Differences (FR-IFD) application (Sect. 6.6), the image energy at discrete spatial filter output (Sect. 6.3), and input signal matrix (Sect. 6.8). Section 6.9 addresses the average squared error minimization. The quasioptimal spatial filtering is studied in Sect. 6.10. Section 6.11 contains the canonical form of output energy functional. Conclusion is drawn in Sect. 6.12.

6.2 Problem Statement

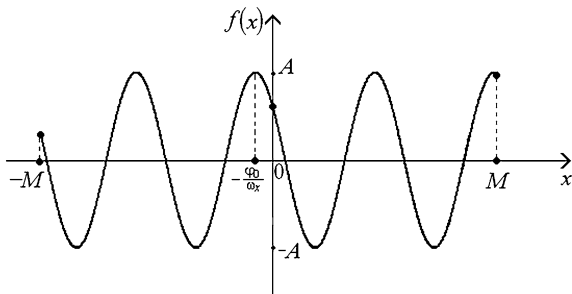
Let us introduce a function $f(x) = A \cos(\omega_x x + \varphi_0)$ presented in Fig. 6.1. Let us consider energy E of finite by a spatial coordinate of cosine curve determined by Eq. 6.1, where $\text{sinc } 2\omega_x M = \frac{\sin 2\omega_x M}{2\omega_x M}$.

$$E = \int_{-M}^M f^2(x) dx = A^2 M (1 + \text{sinc } 2\omega_x M \cdot \cos 2\varphi_0) \tag{6.1}$$

From Eq. 6.1, it follows that energy of a cosine curve’s finite section depends essentially on initial phase φ_0 . Figure 6.2 shows the $\frac{E}{A^2 M}$ normed energy dependence on $2\omega_x M$ normed spatial frequency at various values of φ_0 initial phase. It can be seen that this dependence most intensely becomes apparent at lower frequencies. However, this dependence is available at any and all frequencies except the following ones $2\omega_x M = k\pi, k = \pm 1, \pm 2, \dots$. The energy of a cosine curve’s finite section remains constant, if $\varphi_0 = \pm \frac{\pi}{4} + \frac{\pi k}{2}$.

A lens system of video camera as well as an eye pupil is a window, through which an image finite by spatial coordinates is formed. If an object being observed is moving, then during its movement through an image field every spatial cosine curve, that forms object’s image spectrum, gains an additional phase incursion [35, 36]. If two different frames are viewed, then such an incursion can be interpreted as a change of all spectral components’ initial phases. Consequently, by measuring

Fig. 6.1 Finite section of spatial cosine curve



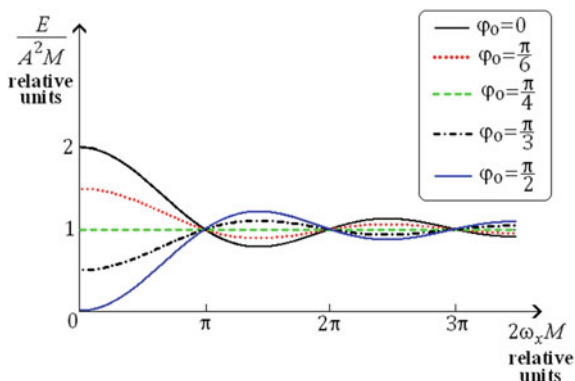


Fig. 6.2 The normed energy dependence from the normed spatial frequency at various values of initial phase

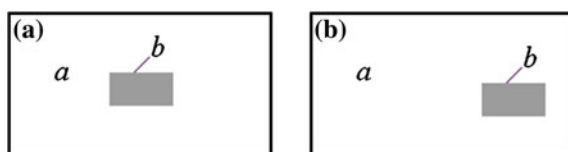


Fig. 6.3 Scheme of object motion: **a** a moving object in the first frame, **b** a moving object in the second frame

energy difference of two frames or, to be more accurate, the difference of energy spectra according to spatial frequencies forming these spectra, the moving objects can be detected as well as their motion characteristics are determined. This can be performed, even if image energy of object movement remains constant as, for example, in Fig. 6.3, where the background intensity a and the object brightness b remain invariable as well as object dimensions.

6.3 Frequency Response

In references dealing with images processing, it is customary to place the origin of coordinates in a frame upper left corner as shown in Fig. 6.4a. Due to traditional TV scanning, a frame processing is usually performed by spatial signal processing with respect to time, the signal being formed with scanning “from left to right” and from “top to bottom”. However, the spatial coordinates are not causal in contrast with temporal coordinate. Aside from that nowadays examples of signal spatial processing without time base are known (e.g. charge-coupled devices operating in an undirected charge transfer mode [34]). Therefore, in this research a more natural system of coordinates was used as shown in Fig. 6.4b. The origin of coordinates is placed in a frame center, x and y are integral-valued coordinates, $x \in [-M; M]$, $y \in [-N; N]$.

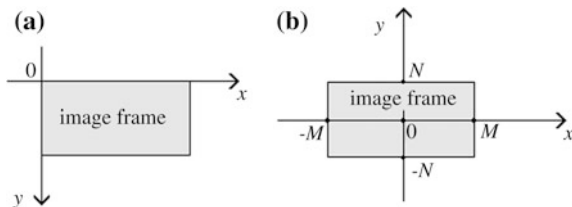


Fig. 6.4 The coordinate systems: **a** usually used in the image processing, **b** the natural coordinate system applied for 2D spatial signals

The position measurement of moving object can be divided (factored). Thus, for example, only lines are enough to be considered.

Let only one moving object be in an image. Let us describe a $f(m)$ video signal of any picture line by an applicative model. Then in the first frame (see Fig. 6.5) Eq. 6.2 will be executed, where a_k and b_k are signals describing a static object and a moving object, respectively.

$$f_1(m) = \sum_{k=-M}^{l-1} a_k \delta(k-m) + \sum_{k=l}^r b_k \delta(k-m) + \sum_{k=r+1}^M a_k \delta(k-m) \tag{6.2}$$

$$\delta(k-m) = \begin{cases} 1 & k = m \\ 0 & k \neq m \end{cases}$$

In the next frame, Eq. 6.3 will be fulfilled, where i is a number of pixels, on which the object has moved.

$$f_2(m) = \sum_{k=-M}^{l-1+i} a_k \delta(k-m) + \sum_{k=l+i}^{r+i} b_k \delta(k-m) + \sum_{k=r+1+i}^M a_k \delta(k-m) \tag{6.3}$$

From this point of view, let us always consider that $i > 0$ and a motion direction should be marked with a «+» sign before i that means a motion to the right, and a «-» sign means a motion to the left.

Let us write down a line energy spectrum $S(\varphi_x)$ as Eq. 6.4 [34], where $\varphi_x \in [-\pi, \pi]$ is a normed spatial frequency, $S(p)$ is a line energy characteristic.

$$S(\varphi_x) = \sum_{p=0}^{2M} S(p) \cos p\varphi_x \tag{6.4}$$

Apart from the energy spectrum, let us apply a function determined by Eq. 6.5, where $\Phi(\varphi_x)$ is a line image phase spectrum.

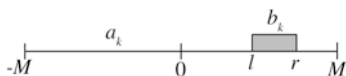


Fig. 6.5 Picture line

$$I(\varphi_x) = \frac{d\Phi(\varphi_x)}{d\varphi_x} S(\varphi_x) \quad (6.5)$$

A function $I(\varphi_x)$ contains information both of line image phase features and of its energy. Therefore, let us call it a phase-energetic line spectrum.

The next Eq. 6.6 is also correct, where $I(p)$ is a phase-energetic line feature.

$$I(\varphi_x) = \sum_{p=0}^{2M} I(p) \cos p\varphi_x \quad (6.6)$$

Let us also call $S(p)$ and $I(p)$ as the line frequency responses.

For analysis of video sequence, the formulae records at $p = 0$ and $p \neq 0$ are not necessary to be distinguished. Therefore, in future let us apply the modified frequency responses and determine them by Eqs. 6.7–6.8, where γ_k is a video discrete signal value in a pixel with number k .

$$S_m(p) = \sum_{k=-M}^{M-p} \gamma_k \gamma_{k+p} \quad (6.7)$$

$$I_m(p) = \sum_{k=-M}^{M-p} (2k+p) \gamma_k \gamma_{k+p} \quad (6.8)$$

It is necessary to analyze the FR-IFDs for motion estimation in an image, which are calculated by Eqs. 6.9–6.10, where a single character stroke «'» corresponds to a previous frame and a double character stroke «''» corresponds to a next one.

$$\Delta S_m(p) = S_m''(p) - S_m'(p) \quad (6.9)$$

$$\Delta I_m(p) = I_m''(p) - I_m'(p) \quad (6.10)$$

6.4 Frequency Responses Difference Model

Let us discuss the possibility for estimation of motion direction in an image by using frequency responses. For definiteness, let an object be located on the right of the image center and it has moved between frames to the center as it is schematically shown in Fig. 6.6.

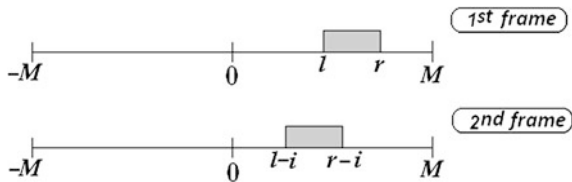


Fig. 6.6 The same line of two frames containing a moving object

An object contrast respect to a background may be characterized by $\Delta_L = b_l - a_{l-1}$ and $\Delta_R = b_r - a_{r+1}$ values or by $\Delta = b - a$ value, where b and a are averaged on some neighboring pixels of the object and the background, respectively.

Let us consider that video signals of background pixels and mutually corresponding object pixels between frames have changed insufficiently and also that $i < r - l$. Let us also consider that the condition (Eq. 6.11) is executed.

$$p > r - l + i \tag{6.11}$$

The analysis shows that in this case a size of changing p from 0 to $2M$ naturally is divided into 10 sub-ranges. Let us consider them in the order of p increasing. Figure 6.7 shows the plots of $\Delta I_M(p)$ and $\Delta S_M(p)$ (a and b are constants).

Consider the following cases.

1. If $p = 0$, then a_{r-n} and b_{l-1-n} values naturally correspond to the second frame in Eqs. 6.12–6.13, respectively.

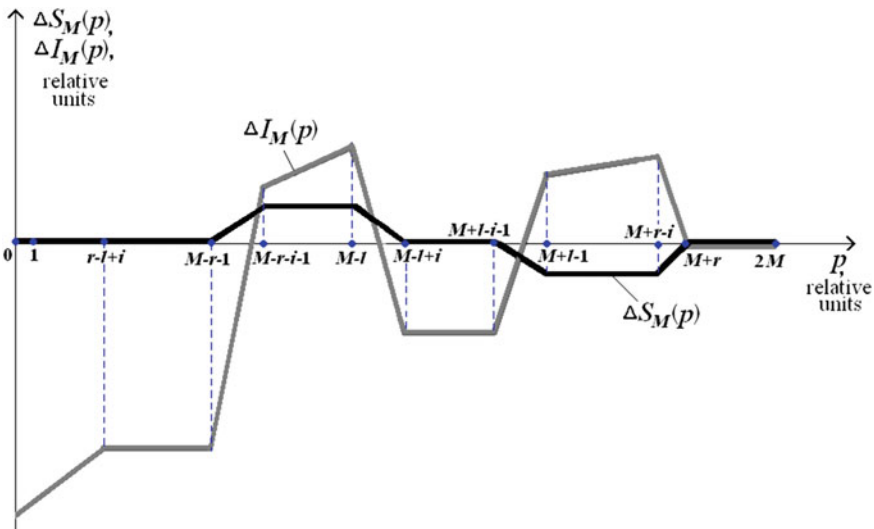


Fig. 6.7 The plots of frame differences frequency responses corresponding to Fig. 6.6

$$\Delta I_M(0) = 2 \sum_{n=0}^{i-1} \Delta_R(n-r) (a_{r-n} + b_{r-n}) + \Delta_L(l-1-n) (a_{l-1-n} + b_{l-1-n}) \quad (6.12)$$

$$\Delta S_M(0) = -\Delta_R \sum_{n=0}^{i-1} (a_{r-n} + b_{r-n}) + \Delta_L \sum_{n=0}^{i-1} (a_{l-1-n} + b_{l-1-n}) \quad (6.13)$$

At constant luminance, an object moving through background is described by expressions $\Delta S_M(0) = 0$ and $\Delta I_M(0) = -2i \cdot \Delta \cdot (a + b) \cdot (r - l + 1) \neq 0$, i.e. the value is proportional to movement i , Δ object contrast and increased respect to an object length by one pixel.

For any p , a frequency responses difference can be described as Eqs. 6.14–6.15, where parameters I_k and S_k are calculated by Eq. 6.16.

$$\Delta I_M(p) = \Delta_L \cdot (I_3 + I_4) - \Delta_R \cdot (I_1 + I_2) \quad (6.14)$$

$$\Delta S_M(p) = \Delta_L \cdot (S_3 + S_4) - \Delta_R \cdot (S_1 + S_2) \quad (6.15)$$

$$I_k = \sum_{n=0}^{u_k} \left(2v_k + (-1)^k p \right) a_{v_k} \quad S_k = \sum_{n=0}^{u_k} a_{v_k} \quad k = \overline{1, 4} \quad (6.16)$$

2. With regard to next phases $1 \leq p \leq M - r - 1$ at $1 \leq p \leq r - l + i$ here and in next three sub-ranges special cases, the additional analysis is required. At $r - l + i < p \leq M - r - 1$, Eqs. 6.14–6.15 are correct, where variables accept values from Eq. 6.17.

$$\begin{cases} u_1 = u_2 = u_3 = u_4 = i - 1 \\ v_1 = r + p - n \\ v_2 = r - p - n \\ v_3 = l - 1 + p - n \\ v_4 = l - 1 - p - n \end{cases} \quad (6.17)$$

As the limits of sums in Eqs. 6.14–6.15 are equal, and in expressions for $I_1 - I_4$ a phase number p enters with the opposite signs, values $\Delta S_M(p)$ and $\Delta I_M(p)$ show small dependence on p . Also, among other factors, when a and b are constants (see Fig. 6.7), $\Delta S_M(p) = 0$ and $\Delta I_M(p) = -4i \cdot a \cdot \Delta \cdot (r - l + 1)$.

3. A near pulse front of the FR-IFD is formed at $M - r \leq p \leq M - r + i - 1$. To find the FR-IFD, it is necessary to assume that $u_1 = M - p - r + i - 1$ and $v_1 = M - n$, remaining u_k and v_k values are the same as in the previous case. Edge duration is marked by i pixels, i.e. it equals the object movement.
4. A near pulse peak of the FR-IFD is observed at $M - r + i \leq p \leq M - l$. In this connection, it should be assumed in Eqs. 6.14–6.15 that $I_1 = S_1 = 0$, u_k and v_k values are determined by Eq. 6.17. As limits of summing in Eq. 6.16 are equal,

$\Delta S_M(p)$ will have small dependence on a value p , and $\Delta I_M(p)$ will increase together with p increase.

5. At $M - l + 1 \leq p \leq M - l + i$, a near pulse edge of the FR-IFD may be estimated. In this connection $I_1 = S_1 = 0$, $u_3 = M - p - l + i$, $v_3 = M - n$. A pulse fall time coincides with a pulse rise time.
6. A pulse-to-pulse area is located at $M - l + i + 1 \leq p \leq M + l - i - 1$. In Eqs. 6.14–6.15, it is necessary to consider that $I_1 = I_3 = 0$ and $S_1 = S_3 = 0$. Also for u_2 , u_4 , v_2 , v_4 values, Eq. 6.17 is correct.
7. A remote pulse front of the FR-IFD corresponds to $M + l - i \leq p \leq M + l - 1$. A dissimilarity from the previous case consists only in the expression for $u_4 = M + l - 1 - p$.
8. A remote pulse peak of the FR-IFD is characterized by $M + l \leq p \leq M + r - i$ phases numbers. In Eqs. 6.14–6.15 for the given and next sub-ranges, only I_2 and S_2 values are remained. In this case, $u_2 = i - 1$.
9. A remote pulse edge of the FR-IFD is described by $M + r - i + 1 \leq p \leq M + r$, $u_2 = M + r - p$ phases numbers.
10. At $p \geq p_{\max} = M + r + 1$, $\Delta S_M = \Delta I_M = 0$ is executed.

The essential singularity of the proposed model is that it links video signal values corresponding to a stationary background with features inherent in a moving object. At the same time, the number of video signal values of the background is a relatively few; it varies from 1 to $4i$ depending on a sub-range. In this connection, it can be possible “to consider” various sites of dependences by changing some values of video signal. For example, a change of value in pixel numbered as $(-M)$ brings the influence on a pulse front and remote pulse edge of the FR-IFD.

Figure 6.8 shows other possible variants of moving object location.

Fig. 6.8 Other object possible movements:
a motion to the *right*, $M > 0$,
b motion to the *right*, $M < 0$,
c motion to the *left*, $M < 0$

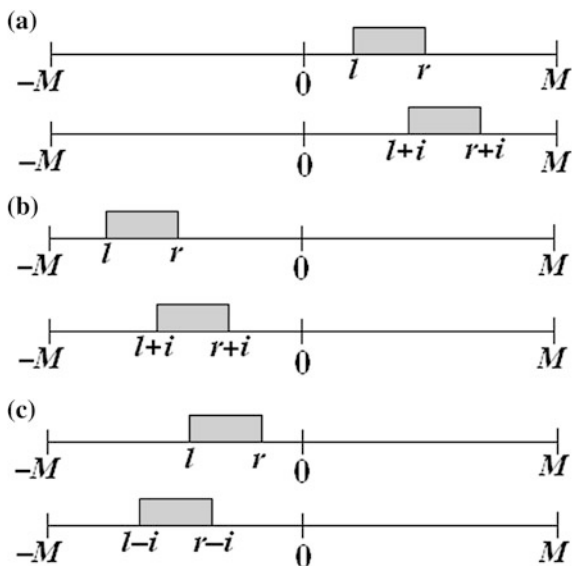


Table 6.1 Rules for obtaining models in other movement cases

Figure number	Substitutions in the model corresponding to Fig. 6.6
Figure 6.8a	$r \rightarrow r + i, \quad l \rightarrow l + i, \quad \Delta \rightarrow -\Delta$
Figure 6.8b	$r \leftrightarrow -l, \quad (l \rightarrow -r), \quad \Delta_R \leftrightarrow \Delta_L, \quad \gamma_k \rightarrow \gamma_{-k}$
Figure 6.8c	$r - i \leftrightarrow -l, \quad (l - i \rightarrow -r), \quad \Delta_R \leftrightarrow \Delta_L, \quad \gamma_k \rightarrow \gamma_{-k}, \quad \Delta \rightarrow -\Delta$

The respective models can be obtained from the considered model by a way for applying of exchange rules situated in Table 6.1.

For example, in the case of situation, represented in Fig. 6.8a, $\Delta S_M(p)$ takes the form $\Delta S_M(p) = -\Delta_L \sum_{n=0}^{M-l-p} a_{l+p+n}$, and a remote pulse edge will be observed under the assumption that $M - l - i + 1 \leq p \leq M - l$.

6.5 Special Cases

Such cases arise at violation of condition from Eq. 6.11, i.e. when $p \leq r - l + i$. As experience shows, the special cases are essential for the FR-IFD applying. Such cases can be realized in sub-ranges of 2–5 by p changing in the spatial and frequency domains of the close pulse and close phase forming, respectively. In this case, the FR-IFD forms the maximal number of video signal values.

The following special cases are possible:

1. $p \leq i$ (It is realized at $r \geq M - i - 1$ in the second sub-range, also at $r \geq M - i$ in the third sub-range).
2. $p \leq r - l - i$ (at $r \geq M - (r - l) + i - 1$ in the second sub-range, at $r \geq M - (r - l) + i$ in the third sub-range, at $r \geq M - (r - l) + 2i$ in the fourth sub-range).
3. $p \leq r - l$ (at $r \geq M - (r - l) - 1$ in the second sub-range, at $r \geq M - (r - l)$ in the third sub-range, at $r \geq M - (r - l) + i$ in the fourth sub-range).
4. $p \leq r - l + i$ (at $r \geq M - (r - l) - i - 1$ in the second sub-range, at $r \geq M - (r - l) - i$ in the third sub-range, at $r \geq M - (r - l)$ in the fourth sub-range, at $r \geq M - i + 1$ in the fifth sub-range).

In any case, the FR-IFDs are determined by Eqs. 6.18–6.19, where I_1, I_4, S_1, S_4 are estimated by Eq. 6.16, values $I_2^{a(b)}, I_3^{a(b)}, S_2^{a(b)}, S_3^{a(b)}$ are also determined by Eq. 6.16, if a is a superscript. If b is a superscript, then the Impulse Response (IR) counts of video signal a corresponding to the background should be exchanged for IR counts of video signal b corresponding to the object (Eq. 6.16).

$$\Delta I_M(p) = I_1 + I_2^{a(b)} + I_3^{a(b)} + I_4 + I'_{1(2)} \quad (6.18)$$

$$\Delta S_M(p) = S_1 + S_2^{a(b)} + S_3^{a(b)} + S_4 + S'_{1(2)} \quad (6.19)$$

Variables I'_1, I'_2, S'_1 , and S'_2 are calculated in a following way:

$$\begin{aligned}
 I'_1 &= \sum_{n=0}^{i-1-p} (2r-p-2n) \left[(a_{r-p-n} a_{r-n})'' - (b_{r-p-n} b_{r-n})' \right] \\
 &\quad - \sum_{n=0}^{i-1-p} (2l-2-p-2n) \left[(a_{l-1-p-n} a_{l-1-n})'' - (b_{l-1-p-n} b_{l-1-n})' \right] \\
 I'_2(z_1; z_2) &= \sum_{n=0}^{z_1} (2z_2+p) \left[(b_{z_2} a_{z_2+p})'' - (a_{z_2} b_{z_2+p})' \right] \\
 S'_1 &= \sum_{n=0}^{i-1-p} \left[(a_{r-p-n} a_{r-n})'' - (b_{r-p-n} b_{r-n})' \right] - \sum_{n=0}^{i-1-p} \left[(a_{l-1-p-n} a_{l-1-n})'' - (b_{l-1-p-n} b_{l-1-n})' \right] \\
 S'_2(z_1; z_2) &= \sum_{n=0}^{z_1} \left[(b_{z_2} a_{z_2+p})'' - (a_{z_2} b_{z_2+p})' \right].
 \end{aligned}$$

Case 1 can be realized only as 2nd and 3rd sub-ranges (accordingly items 2.1 and 3.1 in Table 6.2). In this connection:

$$\begin{aligned}
 I'_{1(2)} = I'_1 \quad S'_{1(2)} = S'_1 \quad I_2^{a(b)} = I_2^b \quad I_3^{a(b)} = I_3^b \quad S_2^{a(b)} = S_2^b \quad S_3^{a(b)} = S_3^b \\
 u_2 = u_3 = u_4 = p-1 \quad v_2 = r-i-n \quad v_3 = l-1+p-n \quad v_4 = l-i-1-n.
 \end{aligned}$$

All special cases for the phase-energetic frequency response are shown in Table 6.2.

6.6 FR-IFD Application

The FR-IFDs are very sensitive to changing of video signal values. The numerical estimates can be obtained on the basis of the given above model. Figure 6.9 shows the changing of energy responses' interframe differences at nullification of pixels under -19 and 0 numbers, and Fig. 6.10 shows changing of phase-energy responses' frame-to-frame differences at nullification of pixels under -1 and 1 numbers. Nullification is performed only in the second frame. Numerical values of parameters are: $M = 27$, $l = 6$, $r = 14$, $i = 3$, $a = 123$, $b = 133$.

Let us presume that a value changing of video signal occurred in the second frame in a pixel numbered: $\gamma''_{s1} = \gamma''_s + \Delta\gamma_s$. Then Eqs. 6.20–6.21 can be obtained using Eqs. 6.7–6.10.

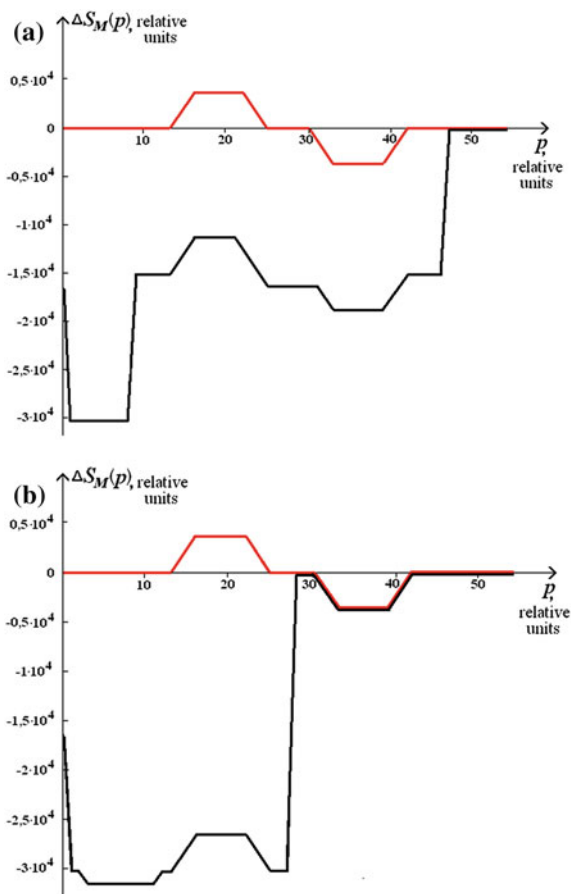
$$\Delta S_{M1}(p) = \Delta S_M(p) + \Delta\gamma_s \cdot \left(\gamma''_{s-p} + \gamma''_{s+p} \right) \quad (6.20)$$

$$\Delta I_{M1}(p) = \Delta I_M(p) + 2s\Delta\gamma_s \cdot \left(\gamma''_{s-p} + \gamma''_{s+p} \right) - p\Delta\gamma_s \cdot \left(\gamma''_{s-p} - \gamma''_{s+p} \right) \quad (6.21)$$

Table 6.2 Special cases

Special cases	Sub-range	FR-IFD	u_1	u_2	u_3	u_4	γ_1	γ_2	γ_3	γ_4	
1	$p \leq i$	$I_1 + I_2' + I_3' + I_4 + I_1'$	$p - 1$	$p - 1$			$\gamma + p - n$	$\gamma - i - n$	$l - 1 + p - n$	$l - 1 + p - n$	
	3.1		$M - r - 1$				$M - n$				
2	$p \leq r - l - i$	$I_1 + I_2' + I_3' + I_4$	$i - 1$	$i - 1$			$\gamma + p - n$	$\gamma - p - n$		$l - 1 - p - n$	
			3.3				$M - p - r + i - 1$				$M - n$
			4.2				$-$				$-$
			4.3				$i - 1$				$\gamma + p - n$
3	$p \leq r - l$	$I_1 + I_2' + I_3' + I_4 + I_3(p - r + l + i - 2; i - 1 - n)$	$M - p - r + i - 1$	$l - r - p$			$M - n$		$r - i - n$		
			4.3				$-$				$-$
			4.3				$i - 1$				$\gamma + p - n$
4	$p \leq r - l + i$	$I_1 + I_2 + I_3 + I_4 + I_1'(r - l + i - p; r - p - n)$	$i - 1$	$p - r + i - 2$	$p - r + i - 2$	$i - 1$	$\gamma + p - n$	$l - i - 1 - n$	$l - 1 + p - n$		
			3.4				$M - p - r + i - 1$				$M - n$
			4.4				$-$				$-$
			5.4				$-$				$M + r - 1$

Fig. 6.9 Energy responses of interframe differences: **a** reset pixel numbered (-19), **b** reset pixel numbered 0



From Eqs. 6.20–6.21 and Figs. 6.9–6.10, it follows that a component $\Delta\gamma_s$ contributes to the FR-IFD at whatever p value, moreover, this contribution substantially depends both on s number and on $\Delta\gamma_s$ component amount, as well as on p size of changing.

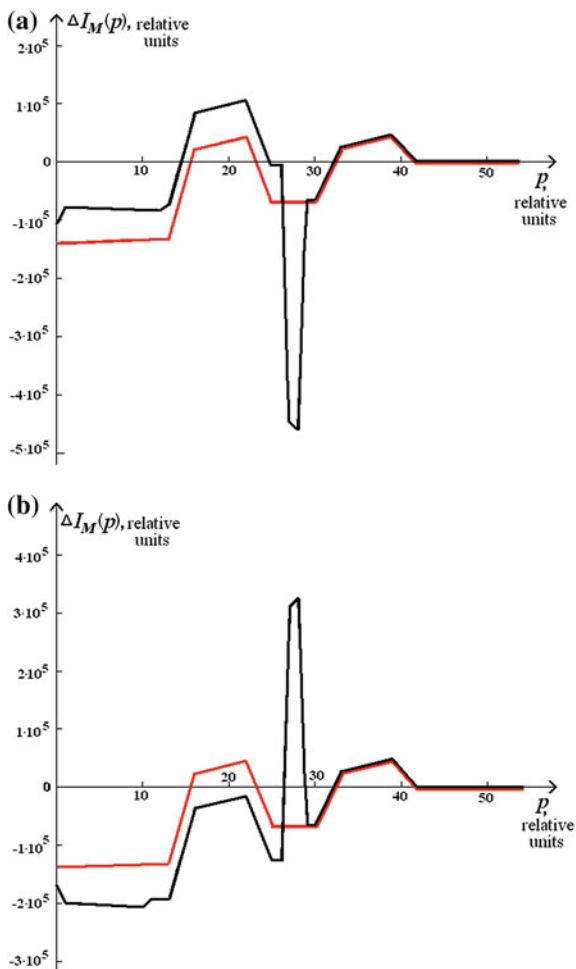
At nullification of s -pixel Eqs. 6.22–6.23 follow from Eqs. 6.20–6.21.

$$\Delta S_{M1}(p) = \Delta S_M(p) - \gamma'_s \cdot (\gamma''_{s-p} + \gamma''_{s+p}) \quad (6.22)$$

$$\Delta I_{M1}(p) = \Delta I_M(p) - 2s\gamma'_s \cdot (\gamma''_{s-p} + \gamma''_{s+p}) + p\gamma''_s \cdot (\gamma''_{s-p} - \gamma''_{s+p}) \quad (6.23)$$

A nullification of pixels at image edges (as well as addition of new zero-order pixels) allows to pick out several moving objects and to determine their motion parameters [34].

Fig. 6.10 Phase-energetic responses of interframe differences: **a** reset pixel numbered (-1), **b** reset pixel numbered 1



During processing of images distorted by additive noise (that is typical, for example, for uttermost low illuminance conditions in real scenes), the FR-IFD estimations begin to fail. This appears a necessity to decrease a resolving power at image edges in order to reduce a noise impact. To achieve this, the edge pixels need to be zeroed in increments at regular intervals. The best results are obtained, if the same line of the first frame is zeroed by the length of k_1 and $k_1 + k_2$ pixels, and frequency responses and their difference are found. Then a corresponding second frame line should undergo the same operations, and after that the difference of differences (or the second difference) should be found [34]. At k_1 and k_2 defined values, the so-called extreme double differences are formed allowing a separation of moving objects from a noise.

The moving objects appearing within eyeshot are most dangerous or interesting for living beings. However, an eye retina has decreased resolving power just at the periphery. This allows to presume that the biological systems apply the algorithms for movement detection similar to the above-mentioned.

6.7 Image Energy at Discrete Spatial Filter Output

The tasks of visual objects sharpening into cluttered background (image enhancement) and a noise reduction (contrast decay) are usually solved by the signals filtering. The projects of visual objects may be interpreted as the small-size patterns of fixed form with contours that can also be considered as the small-size elements, and other structures occupying a lesser part of an image. All they are characterized by a low energy respect to other image components with larger size. Therefore, it is desirable to increase a relative energy of small-size objects' projects by a filtering.

As a rule, the large-size structures, namely spots of brightness, atmospheric formations, shadowing, etc. cause a noise that impair the images' analysis process. The noise-causing structures may be the regular texture structures like sea disturbance, urban development, forest areas, etc., which can be described in terms of narrow-band spatial process. Still in any case, it is desirable to reduce the respect energy of noise image signals.

Thus, a relative change of image signals energy can serve as an index of filtering efficiency for image signals. This index should connect the characteristics of input and output signals with filter parameters.

A signal spectrum by the Discrete Filter (DF) output will have a view of Eq. 6.24, where $\dot{S}_{\text{input}}(\varphi_x, \varphi_y)$ and $\dot{S}_{\text{output}}(\varphi_x, \varphi_y)$ are an input and output image spectra, respectively; $\dot{H}(\varphi_x, \varphi_y)$ is the DF gain; $\varphi_x, \varphi_y \in [-\pi, \pi]$ are the normed spatial frequencies.

$$\dot{S}_{\text{output}}(\varphi_x, \varphi_y) = \dot{H}(\varphi_x, \varphi_y) \cdot \dot{S}_{\text{input}}(\varphi_x, \varphi_y) \quad (6.24)$$

Let us perform the complex conjugation operation from Eq. 6.24 as Eq. 6.25.

$$\dot{S}_{\text{output}}^*(\varphi_x, \varphi_y) = \dot{H}^*(\varphi_x, \varphi_y) \cdot \dot{S}_{\text{input}}^*(\varphi_x, \varphi_y) \quad (6.25)$$

After multiplication the corresponding parts of Eqs. 6.24–6.25, Eq. 6.26 is obtained, where $\tilde{S}_{\text{input}}(\varphi_x, \varphi_y)$ и $\tilde{S}_{\text{output}}(\varphi_x, \varphi_y)$ are energy spectra of input and output image signals, respectively.

$$\tilde{S}_{\text{output}}(\varphi_x, \varphi_y) = |\dot{H}(\varphi_x, \varphi_y)|^2 \cdot \tilde{S}_{\text{input}}(\varphi_x, \varphi_y) \quad (6.26)$$

The energy of processed image signal will be obtained, if $\tilde{S}_{\text{output}}(\varphi_x, \varphi_y)$ energy spectrum is integrated by all arguments from Eq. 6.27.

$$E = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |\dot{H}(\varphi_x, \varphi_y)|^2 \cdot \tilde{S}_{\text{input}}(\varphi_x, \varphi_y) d\varphi_x d\varphi_y \quad (6.27)$$

For the DF gain squared absolute value, Eq. 6.28 can be obtained [34], where $\alpha_{*,*}$ are the elements of impulse response of the DF with aperture $(a + b + 1) \times (c + d + 1)$, a, b, c, d are nonnegative integers (b and d can be equal to zero).

$$\begin{aligned} |\dot{H}(\varphi_x, \varphi_y)|^2 = & 2 \sum_{l=-a-i}^b \sum_{m=-c-j}^d \sum_{i=-(a+b)}^0 \sum_{\substack{j=-(c+d) \\ i+j \neq 0}}^0 \alpha_{l+i, m+j} \cdot \alpha_{l, m} \cdot \cos(i\varphi_x + j\varphi_y) \\ & + 2 \sum_{l=-a-i}^b \sum_{m=-c}^{d-j} \sum_{i=-(a+b)}^{-1} \sum_{j=1}^{-i} \alpha_{l+i, m+j} \cdot \alpha_{l, m} \cdot \cos(i\varphi_x + j\varphi_y) \\ & + 2 \sum_{l=-a}^{b+i} \sum_{m=-c+j}^d \sum_{i=-(a+b)}^{-1} \sum_{j=-i+1}^{c+d} \alpha_{l-i, m-j} \cdot \alpha_{l, m} \cdot \cos(i\varphi_x + j\varphi_y) + \sum_{l=-a}^b \sum_{m=-c}^d \alpha_{l, m}^2 \end{aligned} \quad (6.28)$$

From Eq. 6.28, it follows that the DF squared amplitude frequency spatial characteristics form the weighted cosine curves of all possible arguments. Therefore, Eq. 6.27 should be transcribed as Eq. 6.29.

$$E = \sum_{l, m, i, j} \alpha_{l+i, m+j} \alpha_{l, m} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \cos(i\varphi_x + j\varphi_y) \cdot \tilde{S}_{\text{input}}(\varphi_x, \varphi_y) d\varphi_x d\varphi_y \quad (6.29)$$

A dual integral in Eq. 6.29 can be considered as an expanding (i, j) -coefficient of $\tilde{S}_{\text{input}}(\varphi_x, \varphi_y)$ function in a bivariate Fourier cosine transformation

$$a_{i, j} = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \cos(i\varphi_x + j\varphi_y) \cdot \tilde{S}_{\text{input}}(\varphi_x, \varphi_y) d\varphi_x d\varphi_y.$$

Taking into account Eq. 6.28, Eq. 6.30 is obtained.

$$\begin{aligned}
E_{\text{output}} &= 2 \sum_{l=-a-i}^b \sum_{m=-c-j}^d \sum_{i=-(a+b)}^0 \sum_{\substack{j=-(c+d) \\ i+j \neq 0}}^0 \alpha_{l+i,m+j} \cdot \alpha_{l,m} \cdot a_{i,j} \\
&+ 2 \sum_{l=-a-i}^b \sum_{m=-c}^{d-j} \sum_{i=-(a+b)}^{-1} \sum_{j=1}^{-i} \alpha_{l+i,m+j} \alpha_{l,m} a_{i,j} \\
&+ 2 \sum_{l=-a}^{b+i} \sum_{m=-c+j}^d \sum_{i=-(a+b)}^{-1} \sum_{j=-i+1}^{c+d} \alpha_{l-i,m-j} \cdot \alpha_{l,m} \cdot a_{i,j} + \sum_{l=-a}^b \sum_{m=-c}^d \alpha_{l,m}^2 E_{\text{input}}
\end{aligned} \tag{6.30}$$

Thus, the energy of output signal is determined by a bivariate Fourier cosine transformation of energy spectrum cosines for input signal and $\alpha_{i,j}$ coefficients.

Let us norm Eq. 6.30 dividing by the input signal energy:

$$E_{\text{input}} = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \tilde{S}_{\text{input}}(\varphi_x, \varphi_y) d\varphi_x d\varphi_y$$

and obtain Eq. 6.31, where $s_{i,j} = \frac{a_{i,j}}{E_{\text{input}}}$, obviously that $s_{0,0} = 1$.

$$\begin{aligned}
e = \frac{E_{\text{output}}}{E_{\text{input}}} &= 2 \sum_{l=-a-i}^b \sum_{m=-c-j}^d \sum_{i=-(a+b)}^0 \sum_{\substack{j=-(c+d) \\ i+j \neq 0}}^0 \alpha_{l+i,m+j} \cdot \alpha_{l,m} \cdot s_{i,j} \\
&+ 2 \sum_{l=-a-i}^b \sum_{m=-c}^{d-j} \sum_{i=-(a+b)}^{-1} \sum_{j=1}^{-i} \alpha_{l+i,m+j} \cdot \alpha_{l,m} \cdot s_{i,j} \\
&+ 2 \sum_{l=-a}^{b+i} \sum_{m=-c+j}^d \sum_{i=-(a+b)}^{-1} \sum_{j=-i+1}^{c+d} \alpha_{l-i,m-j} \cdot \alpha_{l,m} \cdot s_{i,j} + \sum_{l=-a}^b \sum_{m=-c}^d \alpha_{l,m}^2
\end{aligned} \tag{6.31}$$

It can be shown [34], that Eq. 6.32 is executed, i.e. the expansion coefficients of energy spectrum of video signal into the double Fourier series relative to the cosines coincide with values of autocorrelation function $K(k, l)$ normed by energy video signal of input image.

$$s_{k,l} = \frac{1}{E_{\text{input}}} \cdot K(k, l) \tag{6.32}$$

In other words, the expansion coefficients coincide with values of input image autocorrelation coefficient. Equation 6.31 can be considered as a functional defined by a set of impulse responses of the DF.

6.8 Input Signal Matrix

In general, an energy spectrum has central inversion, that is $s_{ij} = s_{-i,-j}$. Let us apply only s_{ij} coefficients, for which i is a non-positive. If $i = 0$, then j is a non-positive. Therefore, the rule can be obtained (Eq. 6.33).

$$s_{i,j} = \begin{cases} s_{ij} & \text{if } i < 0 \\ s_{0,-|j|} & \text{if } i = 0 \\ s_{-i,-j} & \text{if } i > 0 \end{cases} \quad (6.33)$$

For all $s_{i,j}$ coefficients should make $(a + b + 1)(c + d + 1) + (a + b)(c + d)$.

A system for finding critical points of e can be found from Eq. 6.31 as Eq. 6.34, where $e'_{i,j}$ is a derivative of e according to $\alpha_{i,j}$; $i \in [-a; b]$, $j \in [-c; d]$.

$$e'_{i,j} = 2 \sum_{l=0}^{a+b} \sum_{m=0}^{c+d} s_{i+a-l,j+c-m} \cdot \alpha_{-a+l,-c+m} = 0 \quad (6.34)$$

It is a system of $(a + b + 1)(c + d + 1)$ linear homogeneous equations. For its non-trivial solution, it is necessary and enough to have a system singular matrix composed of $s_{i,j}$ coefficients.

Let us call this matrix an input signal matrix and indicate it by a **S** symbol. This matrix is a square symmetric matrix having $[(a + b + 1)(c + d + 1)]$ order. Figure 6.11 shows a system **S**-matrix for $a = b = c = d = 1$ [34]. In addition, the indices of all $s_{i,j}$ elements are changed in accordance with rule (Eq. 6.33).

The **S**-matrix has a block structure, i.e. it consists of square matrices of a $(c + d + 1)$ size. In Fig. 6.11, these blocks are marked out by the dot lines. In Fig. 6.12, an input signal matrix for $a = 2$ and $b = 1$ is shown as an example, where A, B, C, D are also square matrices of $(c + d + 1)$ order, “ T ” means conjugation.

Figure 6.13 shows **S**-matrix for the DF (3×4) , in which $a = b = d = 1$ and $c = 2$. Blocks of $(c + d + 1) = 4$ order are marked out by a dot line.

The order of $\alpha_{i,j}$ and $e'_{i,j}$ (Figs. 6.11 and 6.13) is suitable, when a filter aperture is known. But if a filter aperture shall have to be installed, then another sequence order beginning from $\alpha_{0,0}$ and $e'_{0,0}$ is desirable. In this case, the increase of filter aperture changes for the **S**-matrix occur only in one direction.

One of possible variants is suggested in Fig. 6.14, and the **S**-matrix fragment is shown in Fig. 6.15. During the increase of filter aperture, new columns will be added from the right and new lines from below.

$$\begin{matrix}
 & \alpha_{1,1} & \alpha_{1,0} & \alpha_{1,-1} & & \alpha_{0,1} & \alpha_{0,0} & \alpha_{0,-1} & & \alpha_{-1,1} & \alpha_{-1,0} & \alpha_{-1,-1} \\
 \begin{matrix} e'_{1,1} \\ e'_{1,0} \\ e'_{1,-1} \\ \dots \\ e'_{0,1} \\ e'_{0,0} \\ e'_{0,-1} \\ \dots \\ e'_{-1,1} \\ e'_{-1,0} \\ e'_{-1,-1} \end{matrix} & \left(\begin{array}{cccccccccccc}
 1 & s_{0,-1} & s_{0,-2} & \vdots & s_{-1,0} & s_{-1,-1} & s_{-1,-2} & \vdots & s_{-2,0} & s_{-2,-1} & s_{-2,-2} \\
 s_{0,-1} & 1 & s_{0,-1} & \vdots & s_{-1,1} & s_{-1,0} & s_{-1,-1} & \vdots & s_{-2,1} & s_{-2,0} & s_{-2,-1} \\
 s_{0,-2} & s_{0,-1} & 1 & \vdots & s_{-1,2} & s_{-1,1} & s_{-1,0} & \vdots & s_{-2,2} & s_{-2,1} & s_{-2,0} \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 s_{-1,0} & s_{-1,1} & s_{-1,2} & \vdots & 1 & s_{0,-1} & s_{0,-2} & \vdots & s_{-1,0} & s_{-1,-1} & s_{-1,-2} \\
 s_{-1,-1} & s_{-1,0} & s_{-1,1} & \vdots & s_{0,-1} & 1 & s_{0,-1} & \vdots & s_{-1,1} & s_{-1,0} & s_{-1,-1} \\
 s_{-1,-2} & s_{-1,-1} & s_{-1,0} & \vdots & s_{0,-2} & s_{0,-1} & 1 & \vdots & s_{-1,2} & s_{-1,1} & s_{-1,0} \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 s_{-2,0} & s_{-2,1} & s_{-2,2} & \vdots & s_{-1,0} & s_{-1,1} & s_{-1,2} & \vdots & 1 & s_{0,-1} & s_{0,-2} \\
 s_{-2,-1} & s_{-2,0} & s_{-2,1} & \vdots & s_{-1,-1} & s_{-1,0} & s_{-1,1} & \vdots & s_{0,-1} & 1 & s_{0,-1} \\
 s_{-2,-2} & s_{-2,-1} & s_{-2,0} & \vdots & s_{-1,-2} & s_{-1,-1} & s_{-1,0} & \vdots & s_{0,-2} & s_{0,-1} & 1
 \end{array} \right)
 \end{matrix}$$

Fig. 6.11 S-matrix with a filter aperture (3 × 3)

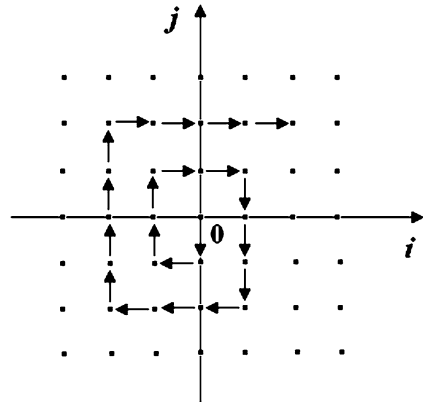
$$\begin{pmatrix}
 A & B & C & D \\
 B^T & A & B & C \\
 C^T & B^T & A & B \\
 D^T & C^T & B^T & A
 \end{pmatrix}$$

Fig. 6.12 Block structure of the S-matrix of (c + d + 1) order

$$\begin{matrix}
 & \alpha_{1,1} & \alpha_{1,0} & \alpha_{1,-1} & \alpha_{1,-2} & & \alpha_{0,1} & \alpha_{0,0} & \alpha_{0,-1} & \alpha_{0,-2} & & \alpha_{-1,1} & \alpha_{-1,0} & \alpha_{-1,-1} & \alpha_{-1,-2} \\
 \begin{matrix} e'_{1,1} \\ e'_{1,0} \\ e'_{1,-1} \\ e'_{1,-2} \\ \dots \\ e'_{0,1} \\ e'_{0,0} \\ e'_{0,-1} \\ e'_{0,-2} \\ \dots \\ e'_{-1,1} \\ e'_{-1,0} \\ e'_{-1,-1} \\ e'_{-1,-2} \end{matrix} & \left(\begin{array}{cccccccccccccccc}
 1 & s_{0,-1} & s_{0,-2} & s_{0,-3} & \vdots & s_{-1,0} & s_{-1,-1} & s_{-1,-2} & s_{-1,-3} & \vdots & s_{-2,0} & s_{-2,-1} & s_{-2,-2} & s_{-2,-3} \\
 s_{0,-1} & 1 & s_{0,-1} & s_{0,-2} & \vdots & s_{-1,1} & s_{-1,0} & s_{-1,-1} & s_{-1,-2} & \vdots & s_{-2,1} & s_{-2,0} & s_{-2,-1} & s_{-2,-2} \\
 s_{0,-2} & s_{0,-1} & 1 & s_{0,-1} & \vdots & s_{-1,2} & s_{-1,1} & s_{-1,0} & s_{-1,-1} & \vdots & s_{-2,2} & s_{-2,1} & s_{-2,0} & s_{-2,-1} \\
 s_{0,-3} & s_{0,-2} & s_{0,-1} & 1 & \vdots & s_{-1,3} & s_{-1,2} & s_{-1,1} & s_{-1,0} & \vdots & s_{-2,3} & s_{-2,2} & s_{-2,1} & s_{-2,0} \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 s_{-1,0} & s_{-1,1} & s_{-1,2} & s_{-1,3} & \vdots & 1 & s_{0,-1} & s_{0,-2} & s_{0,-3} & \vdots & s_{-1,0} & s_{-1,-1} & s_{-1,-2} & s_{-1,-3} \\
 s_{-1,-1} & s_{-1,0} & s_{-1,1} & s_{-1,2} & \vdots & s_{0,-1} & 1 & s_{0,-1} & s_{0,-2} & \vdots & s_{-1,1} & s_{-1,0} & s_{-1,-1} & s_{-1,-2} \\
 s_{-1,-2} & s_{-1,-1} & s_{-1,0} & s_{-1,1} & \vdots & s_{0,-2} & s_{0,-1} & 1 & s_{0,-1} & \vdots & s_{-1,2} & s_{-1,1} & s_{-1,0} & s_{-1,-1} \\
 s_{-1,-3} & s_{-1,-2} & s_{-1,-1} & s_{-1,0} & \vdots & s_{0,-3} & s_{0,-2} & s_{0,-1} & 1 & \vdots & s_{-1,3} & s_{-1,2} & s_{-1,1} & s_{-1,0} \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 s_{-2,0} & s_{-2,1} & s_{-2,2} & s_{-2,3} & \vdots & s_{-1,0} & s_{-1,1} & s_{-1,2} & s_{-1,3} & \vdots & 1 & s_{0,-1} & s_{0,-2} & s_{0,-3} \\
 s_{-2,-1} & s_{-2,0} & s_{-2,1} & s_{-2,2} & \vdots & s_{-1,-1} & s_{-1,0} & s_{-1,1} & s_{-1,2} & \vdots & s_{0,-1} & 1 & s_{0,-1} & s_{0,-2} \\
 s_{-2,-2} & s_{-2,-1} & s_{-2,0} & s_{-2,1} & \vdots & s_{-1,-2} & s_{-1,-1} & s_{-1,0} & s_{-1,1} & \vdots & s_{0,-2} & s_{0,-1} & 1 & s_{0,-1} \\
 s_{-2,-3} & s_{-2,-2} & s_{-2,-1} & s_{-2,0} & \vdots & s_{-1,-3} & s_{-1,-2} & s_{-1,-1} & s_{-1,0} & \vdots & s_{0,-3} & s_{0,-2} & s_{0,-1} & 1
 \end{array} \right)
 \end{matrix}$$

Fig. 6.13 S-matrix with a filter aperture (4 × 4)

Fig. 6.14 One possible sequence for elements of the IR



Such **S**-matrix construction results in other symmetry types' origination. For example, in Fig. 6.15a the sub-determinants of the even order periodically repeating along matrix main diagonal are marked out, and Fig. 6.15b shows the sub-determinants of the odd order. The repeating of sub-determinants order is uninterruptedly increased, and every previous sub-determinant is "inserted" in the following one.

The general recommendations can hardly be given by extremum of the e finding. This is related to that a quadric form matrix (for second-order differential) coincides with an input signal matrix, i.e. with the matrix for first differential; and all differentials higher than those of second-order are identically zero. Therefore, in every particular case the additional research is required.

6.9 Average Squared Error Minimization

During a preliminary processing of images, the essential task of certain class objects finding on a random background appears. For this purpose, it is desirable to weaken a video signal of image background respect to a video signal corresponding to an object of interest as greatly as possible.

Let us determine, what elements of the IR in a linear DF must be, minimizing mean-square difference of a filter output video signal and an object video signal. As an $q_{i,j}$ image model, let us set an applicative model determined by Eq. 6.35, where $g_{i,j}$ is a video signal of object; $f_{i,j}$ is a video signal of background.

$$q_{i,j} = r_{i,j}g_{i,j} + (1 - r_{i,j})f_{i,j} \quad r_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \text{object} \\ 0 & \text{if } (i,j) \in \text{background} \end{cases} \quad (6.35)$$

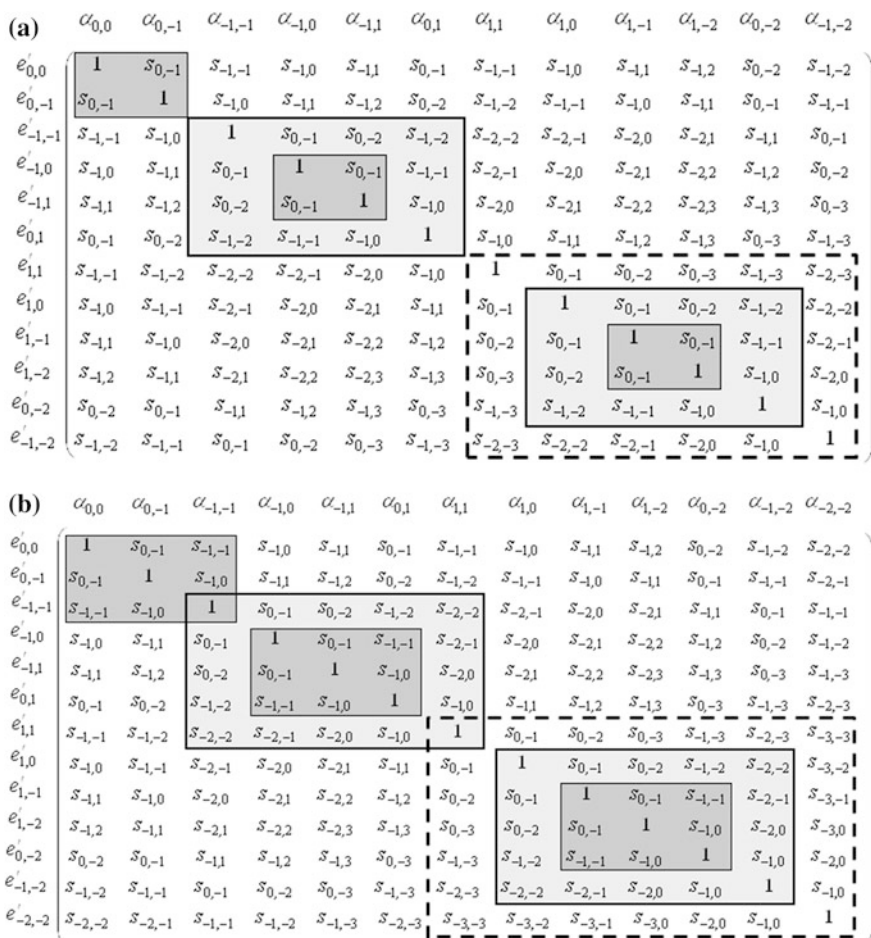


Fig. 6.15 Symmetry of S-matrix: **a** sub-determinants of the even order, **b** sub-determinants of the odd order

At that, a squared error will be as follows as Eq. 6.36.

$$\Delta^2 = \sum_{i=-N}^N \sum_{j=-M}^M \left(r_{i,j} g_{i,j} - \sum_{l=-a}^b \sum_{m=-c}^d q_{i+l,j+m} \alpha_{l,m} \right)^2 \tag{6.36}$$

Let us insert Eq. 6.35 into Eq. 6.36, square it and find a mathematical expectation. Then Eq. 6.37 will be received, where E_g is an object video signal energy, $K_g(l, m)$ are values of object autocorrelation function, $K_{g,f}(l, m)$ are values of

object/background intercorrelation function, $K_{\text{input}}(l-k, m-p)$ are values of autocorrelation function for input video signal.

$$\begin{aligned} \mu(\Delta^2) = & E_g - 2 \sum_{l=-a}^b \sum_{m=-c}^d K_g(l, m) \alpha_{l,m} - 2 \sum_{l=-a}^b \sum_{m=-c}^d K_{g,f}(l, m) \alpha_{l,m} \\ & + \sum_{l=-a}^b \sum_{m=-c}^d \sum_{k=-a}^b \sum_{p=-c}^d K_{\text{input}}(l-k, m-p) \alpha_{l,m} \alpha_{k,p} \end{aligned} \quad (6.37)$$

Values of functions $K_g(l, m)$ и $K_{g,f}(l, m)$ are obtained by corresponding image areas, as it follows from Eq. 6.37.

Let us find the critical points by differentiation of Eq. 6.37 respect to $\alpha_{i,j}$ and then obtain Eq. 6.38.

$$-K_g(i, j) - K_{g,f}(i, j) + \sum_{l=-a}^b \sum_{m=-c}^d K_{\text{input}}(i-l, j-m) \cdot \alpha_{l,m} = 0 \quad (6.38)$$

Let us divide both parts of Eq. 6.38 by E_{input} , then Eq. 6.39 will be received, where R' represents an autocorrelation coefficient, a stroke (') in the right part of Eq. 6.39 means that it has been obtained by a way for normalization of input signal energy and not of corresponding energy.

$$\sum_{l=-a}^b \sum_{m=-c}^d R_{\text{input}}(i-l, j-m) \cdot \alpha_{l,m} = R'_g(i, j) + R'_{g,f}(i, j) \quad (6.39)$$

Let us consider that $R'_g(i, j) + R'_{g,f}(i, j)$ can be exchanged with $R'_{g,f}(i, j)$, if it is to be found throughout the whole image area. Thus, Eq. 6.40 has a view:

$$\sum_{l=-a}^b \sum_{m=-c}^d R_{\text{input}}(i-l, j-m) \alpha_{l,m} = R'_{g,f}(i, j). \quad (6.40)$$

The solution of system (Eq. 6.40) is the elements $\alpha_{l,m}$ of the IR for optimal DF that minimizes an average squared error. In this connection, the Wiener-Hopf filtering is performed.

It follows from Eq. 6.32 that Eq. 6.40 can be transcribed as a system of Eq. 6.41, where $s'_{i,j}$ are values of intercorrelation coefficients of object and input image.

$$\sum_{l=-a}^b \sum_{m=-c}^d s_{i-l, j-m} \alpha_{l,m} = s'_{i,j} \quad (6.41)$$

Compare Eqs. 6.41 and 6.34. The left part of Eq. 6.41 is a system for Eq. 6.34 functional critical points. If the elements of the IR being determined from Eq. 6.41, then the DF is optimal for highlighting of a pre-defined object, and suppression of the rest image video signal.

6.10 Quasi-Optimal Spatial Filtering

An input image determines the finding elements of the IR for optimal filter by means of Eq. 6.41, $s_{i,j}$. A more complicated task is to find values of object and image intercorrelation coefficient $s'_{i,j}$. A joint correlation function with some object called “standard” is usually applied, the standard having the form similar to an object form in a source image. In this connection, a processing result will depend on a “standard” position on an image field.

If the second addend is to be ignored in the right part of Eq. 6.39, then such filtering can be called quasi-optimal. In this case at determining elements of the IR for a quasioptimal DF, the system of Eqs. 6.42 will be obtained, where $s^o_{i,j}$ are values of standard autocorrelation functions normed by input image energy.

$$\sum_{l=-a}^b \sum_{m=-c}^d s_{i-l,j-m} \alpha_{l,m} = s^o_{i,j} \quad (6.42)$$

Let us call such filtering as “Wiener-Hopf-type filtering”.

Let us consider vector determined by Eq. 6.43.

$$\mathbf{H}^T = (\alpha_{-a,-c} \quad \alpha_{-a,-c+1} \quad \dots \quad \alpha_{b,d-1} \quad \alpha_{b,d}) \quad (6.43)$$

Dimensionality of \mathbf{H}^T is determined by number of $\alpha_{i,j}$. A vector $\mathbf{S}_0^T = (s^o_{-a,-c} \quad s^o_{-a,-c+1} \quad \dots \quad s^o_{b,d-1} \quad s^o_{b,d})$ corresponds to the found characteristics of the standard.

An average squared error for quasi-optimal filtering can be obtained by Eq. 6.44 similar to Eq. 6.37, where $E_{\text{std.}}$ is a standard energy, eE_{input} is an energy of processed image, $(\mathbf{H}^T \cdot \mathbf{S}_0)$ is a matrix composition.

$$\mu(\Delta^2) = E_{\text{std.}} + eE_{\text{input}} - 2E_{\text{input}} (\mathbf{H}^T \cdot \mathbf{S}_0) \quad (6.44)$$

In that case, a quasioptimal filtering efficiency can be estimated by Eq. 6.45, where $\xi = \frac{E_{\text{std.}}}{E_{\text{input}}}$.

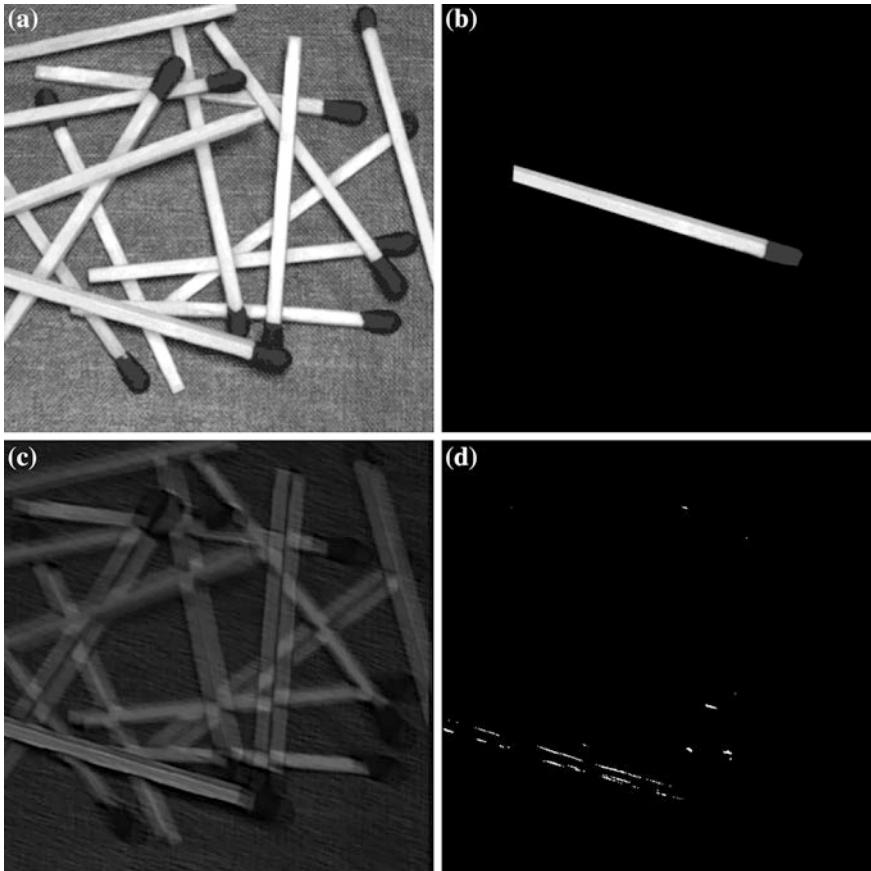


Fig. 6.16 Example of image processing: **a** an original image, **b** an image of the chosen object, **c** the application of a filter aperture (11×11) before threshold processing, **d** the application of a filter aperture (11×11) after threshold processing

$$\eta = \frac{E_{\text{etalon}}}{\mu(\Delta^2)} = \frac{\xi}{\xi + e - 2(\mathbf{H}^T \cdot \mathbf{S}_0)} = \frac{\xi}{\xi - e} \quad (6.45)$$

Let us consider application of algorithm (Eq. 6.42) for solving the task of highlighting a sizable object of certain orientation. Let us highlight the image of a match (Fig. 6.16), its orientation is shown in Fig. 6.16b. In Fig. 6.16c, d, the results of image processing by a filter with (11×11) aperture before and after threshold value applying are shown. Figure 6.16d indicates the possibility of highlighting an object of interest. Therefore, the algorithm (Eq. 6.42) applying conjointly with nonlinear processing allows the pre-defined objects highlighting in the first stage of prefiltering.

The following techniques can be recommended for a quasioptimal DF aperture selection. The system (Eq. 6.42) can be written in the matrix form (Eq. 6.46).

$$\mathbf{S} \cdot \mathbf{H} = \mathbf{S}_0 \quad (6.46)$$

A relative energy of a processed image will be in the form of Eq. 6.47.

$$e = \mathbf{H}^T \cdot \mathbf{S} \cdot \mathbf{H} \quad (6.47)$$

The solution for Eq. 6.46 looks like Eq. 6.48.

$$\mathbf{H} = \mathbf{S}^{-1} \cdot \mathbf{S}_0 \quad (6.48)$$

Let us insert Eq. 6.48 into Eq. 6.47 and after transformation the following correlation is obtained (Eq. 6.49).

$$e = \mathbf{S}_0^T \cdot \mathbf{H} \quad (6.49)$$

Finally from Eqs. 6.48 and 6.49, Eq. 6.50 is obtained.

$$e = \mathbf{S}_0^T \cdot \mathbf{S}^{-1} \cdot \mathbf{S}_0 \quad (6.50)$$

Equation 6.50 allows to define a processing quality at various filter apertures applying for an image. By the e function, the filter aperture ensuring the best possible processing can be selected. Then by means of Eq. 6.48, it can be synthesized.

6.11 Canonical Form of Output Energy Functional

The e function (Eq. 6.31) is a quadratic form and can be transcribed as Eq. 6.47. From the context of e definition as a relation of processed and input images energies, it follows that this quadratic form is positively definite quadratic or positively semidefinite one. If it becomes positively semidefinite in the case, when the \mathbf{S} -matrix order is equal to k , then for matrices of lesser degree this quadratic form appears to be positively definite.

Consider that the \mathbf{S} -matrix of n -order is nondegenerate, i.e. the corresponding quadratic form appears to be positively definite. So according to the Jacobi algorithm, on condition that all principal sub-determinants of the \mathbf{S} -matrix are not equal to zero, Eq. 6.51 can be obtained in this connection with Eq. 6.52, where α_{i+j} are elements of \mathbb{R} for the DF, Δ_i^0 are principal sub-determinants of the \mathbf{S} -matrix, $\Delta_0^0 = 1$, Δ_i^j are subsidiary sub-determinants, for which the last column of the principal sub-determinant is necessary to be substituted by a corresponding part of (i, j) column of the \mathbf{S} -matrix.

$$e = \sum_{i=1}^n \frac{\Delta_{i-1}^0}{\Delta_i^0} A_i^2 \quad (6.51)$$

$$A_i = \frac{1}{\Delta_{i-1}^0} \sum_{j=0}^{n-i} \Delta_i^j \alpha_{i+j} \quad (6.52)$$

According to Sylvester's criterion, all principal subdeterminants of the \mathbf{S} -matrix are in this case positive quantities.

From Eqs. 6.51 and 6.52, Eq. 6.53 follows.

$$e = \sum_{i=1}^{n-1} \frac{\Delta_{i-1}^0}{\Delta_i^0} A_i^2 + \frac{\Delta_n^0}{\Delta_{n-1}^0} \alpha_n \quad (6.53)$$

Equation 6.53 is a quadratic form; its matrix is shown in Eq. 6.54.

$$\begin{pmatrix} \frac{\Delta_1^0}{\Delta_0^0} & 0 & \dots & 0 \\ 0 & \frac{\Delta_2^0}{\Delta_1^0} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\Delta_n^0}{\Delta_{n-1}^0} \end{pmatrix} \quad (6.54)$$

Let us assume that n is a number, at which the \mathbf{S} -matrix is degenerate. In that case $\Delta_n^0 = 0$.

It can be shown that Eq. 6.55 is satisfied in critical points of the e function.

$$A_i = 0 \quad i = \overline{1, n} \quad (6.55)$$

Therefore, a processing efficiency in critical points is equal to zero, i.e. a complete rejection of corresponding input image is performed.

The Eq. 6.55 criterion can be applied for estimation of synthesized filter response to confirm of the required processing quality. In this connection, it is more suitable to apply the converted expressions Eqs. 6.51–6.52 in the form of Eq. 6.56, where \bar{A}_i and A_i are determined from Eq. 6.57.

$$e = \sum_{i=1}^n \frac{\Delta_i^0}{\Delta_{i-1}^0} \bar{A}_i^2 \quad (6.56)$$

$$\bar{A}_i = \frac{\Delta_{i-1}^0}{\Delta_i^0} \quad A_i = \sum_{j=0}^{n-i} \frac{\Delta_i^j}{\Delta_i^0} \alpha_{i+j} \quad (6.57)$$

In this case, also for all critical points $\overline{A}_i = 0$, however in other points \overline{A}_i value is much greater than A_i .

Thus, $\delta(i) = \frac{\Delta_i^0}{\Delta_{i-1}^0}$ relations of principal sub-determinants are located on **S**-matrix diagonal, the **S**-matrix being brought to a diagonal form. The principal sub-determinants define a matrix grade and, consequently, a discrete filter response. For **S**-matrix determination, the matrix entries are normed for signal energy. Therefore, after the subtraction of constant component from an image and the definition of consequent **S**-matrix coefficients the result does not depend from an object contrast [34].

6.12 Conclusion

The proposed way of energy analysis is an alternative approach to the analysis of image and video sequences. The distinctive feature of the considered methods is the accounting of edge effects. A separation of movement and calculation of its characteristics are based on this statement. The analysis of values changing in an image signal at the object edges leads to improvement a performance of motion detection and movement's characteristics. Also criteria for efficiency of image processing are received. The offered approaches can promote the understanding mechanisms of operation in the highly developed vision such as a human vision.

References

1. Behar J, Porat M, Zeevi YY (1992) Image reconstruction from localized phase. *IEEE Trans Signal Process* 40(4):736–743
2. Burt PJ (1984) The pyramid as a structure for efficient for computation. In Rosenfeld A (ed) *Multiresolution image processing and analysis*, Springer Series in Information Sciences 12:6–35
3. Dong DW, Atick JJ (1995) Statistics of natural time-varying images. *Comput Neural Syst* 6:345–358
4. Dunn D, Higgins WE, Wakeley J (1994) Texture segmentation using 2-D Gabor Elementary-Functions. *IEEE Trans Pattern Anal Mach Intell* 16(2):130–149
5. Jähne B (2005) *Digital image processing*, 6th edn. Springer, Berlin
6. Wong A, Clausi DA (2007) ARRSI: automatic registration of remote-sensing images. *IEEE Trans Geosci Remote Sens* 45(5):1483–1492
7. Rozenfeld A (ed) (1984) *Multiresolution image processing and analysis*, of springer series in information sciences, vol 12. Springer, New York
8. Stocker AA, Simoncelli EP (2006) Noise characteristics and prior expectations in human visual speed perception. *Nat Neurosci* 9:578–585
9. Porat M, Zeevi YY (1988) The generalized Gabor scheme of image representation in biological and machine vision. *IEEE Trans Pattern Anal Mach Intell* 10(4):452–468
10. Olshausen BA (2003) Learning sparse, overcomplete representations of time-varying natural images. *IEEE Int Conf Image Proc* 1:41–44

11. Gonzalez RC, Woods RE (2002) Digital image processing, 2nd edn. Prentice Hall, Upper Saddle River
12. Oppenheim AV, Willsky AS, Nawab SH (1996) Signals and systems, 2nd edn. Prentice-Hall, Upper Saddle River
13. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 12(7):629–639
14. Randen T, Husoy JH (1999) Filtering for texture classification: a comparative study. *IEEE Trans Pattern Anal Mach Intell* 21(4):291–310
15. Skarbnik N, Sagiv C, Zeevi YY (2009) Edge Detection and Skeletonization using Quantized phase. In: 17th European signal processing conference (EUSIPCO 2009), pp 1542–1546
16. Ruderman DL (1996) The statistics of natural images. *Comput Neural Syst* 5:517–548
17. Torralba A, Oliva A (2003) Statistics of natural image categories. *Network* 14(3):391–412
18. Weiss Y, Simoncelli EP, Adelson EH (2002) Motion illusions as optimal percepts. *Nat Neurosci* 5:598–604
19. Morrone MC, Burr DC (1988) Feature detection in human vision: a phase-dependent energy model. *Proc R Soc Lond B* 235:221–245
20. Xiao Z, Hou Z (2004) Phase based feature detector consistent with human visual system characteristics. *Pattern Recogn Lett* 25(10):1115–1121
21. Bonnet N, Vautrot P (1997) Image analysis: Is the Fourier transform becoming obsolete. *Microsc Microanal Microstruct* 8(1):59–75
22. Daugman J (2001) Statistical richness of visual phase information: update on recognizing persons by iris patterns. *Int J Comput Vis* 45(1):25–38
23. Gegenfurtner KR, Braun DI, Wichmann FA (2003) The importance of phase information for recognizing natural images. *J Vis* 3(9):519
24. Hansen BC, Hess RF (2007) Structural sparseness and spatial phase alignment in natural scenes. *J Opt Soc Am A* 24(7):1873–1885
25. Morgan MJ, Ross J, Hayes A (1991) The relative importance of local phase and local amplitude in patchwise image reconstruction. *Biol Cybern* 65(2):113–119
26. Kuglin CD, Hines DC (1975) The phase correlation image alignment method. In: *IEEE conference on cybernetics and society*, pp 163–165
27. Michael G, Porat M (2001) Image reconstruction from localized Fourier magnitude. In: *Proceedings of international conference on image processing (Cat. No. 01CH37205)*, pp 213–16
28. Oppenheim AV, Lim JS (1981) The importance of phase in signals. *Proc IEEE* 69(5):529–541
29. Shapiro Y, Porat M (1998) Image representation and reconstruction from spectral amplitude or phase. In: *IEEE international conference on electronics, circuits and systems*, pp 461–464
30. Tanaka H, Yoshida Y, Fukami K, Nakano H (2004) Texture segmentation using amplitude and phase information of Gabor filters. *Electron Commun Jpn Part 3 (Fundam Electron Sci)* 87(4):66–79
31. Kovési P (1999) Image features from phase congruency. *Videre* 1(3):1–14
32. Urieli S, Porat M, Cohen N (1998) Optimal reconstruction of images from localized phase. *IEEE Trans Image Process* 7(6):838–853
33. Li Y, Belkasim S, Chen X, Fu X (2006) Contour-based object segmentation using phase congruency. *IntCongress of Imaging Science ICIS'06* 661–664
34. Bogoslovski AV (2013) Multidimensional signal processing. Radiotec, Moscow (in Russian)
35. Dufaux F, Moscheni F (1995) Motion estimation techniques for digital TV: a review and a new contribution. *Proc IEEE* 83(6):858–876
36. Magarey JFA, Kingsbury NG (1998) Motion estimation using a complex-valued wavelet transform. *IEEE Trans Signal Proc* 46:1069–1084

Chapter 7

Optimal Measurement of Visual Motion Across Spatial and Temporal Scales

Sergei Gepshtein and Ivan Tyukin

Abstract Sensory systems use limited resources to mediate the perception of a great variety of objects and events. Here a normative framework is presented for exploring how the problem of efficient allocation of resources can be solved in visual perception. Starting with a basic property of every measurement, captured by Gabor's uncertainty relation about the location and frequency content of signals, prescriptions are developed for optimal allocation of sensors for reliable perception of visual motion. This study reveals that a large-scale characteristic of human vision (the spatiotemporal contrast sensitivity function) is similar to the optimal prescription, and it suggests that some previously puzzling phenomena of visual sensitivity, adaptation, and perceptual organization have simple principled explanations.

Keywords Resource allocation • Contrast sensitivity • Perceptual organization • Sensory adaptation • Automated sensing

S. Gepshtein (✉)

Systems Neurobiology Laboratories, The Salk Institute for Biological Studies,
10010 North Torrey Pines Road, La Jolla, CA 92037, USA
e-mail: sergei@salk.edu

I. Tyukin

Department of Mathematics, University of Leicester, University Road,
Leicester LE1 7RH, UK
e-mail: i.tyukin@le.ac.uk

I. Tyukin

Department of Automation and Control Processes, Saint-Petersburg State Electrotechnical
University, 5 Professora Popova Str, 197376 Saint-Petersburg, Russia

7.1 Introduction

Biological sensory systems collect information from a vast range of spatial and temporal scales. For example, human vision can discern modulations of luminance that span nearly seven octaves of spatial and temporal frequencies, while many properties of optical stimulation (such as the speed and direction of motion) are analyzed within every step of the scale.

The large amount of information is encoded and transformed for the sake of specific visual tasks using limited resources. In biological systems, the resources are the large but finite number of neural cells. The cells are specialized: each is sensitive to a small subset of optical signals, presenting sensory systems with the problem of allocation of limited resources. This chapter is concerned with how this problem is solved by biological vision. How are the specialized cells distributed across the great number of potential optical signals in the environments that are diverse and variable?

The extensive history of vision science suggests that any attempt of vision theory should begin with an analysis of the tasks performed by visual systems. Following Aristotle, one may begin with the definition of vision as “knowing what is where by looking” [1]. The following argument concerns the basic visual tasks captured by this definition. The “what” and “where” of visual perception are associated with two characteristics of optical signals: their frequency content and locations, in space and time. The last statement implicates at least five dimensions of optical signals (which will become clear in a moment).

The basic visual tasks are bound by first principles of measurement. To see that, consider a measurement device (a “sensor” or “cell”) that integrates its inputs over some spatiotemporal interval. An individual device of an arbitrary size will be more suited for measuring the location or the frequency content of the signal, reflected in the uncertainties of measurement. The uncertainties associated with the location and the frequency content are related by a simple law formalized by Gabor [2], who showed that the two uncertainties trade off across scales. As the scale changes, one uncertainty rises and the other falls.

Assuming that the visual systems in question are interested in both the locations and frequency content of optical signals (“stimuli”), the tradeoff of uncertainties will attain a desired (“optimal”) balance of uncertainties at some intermediate scale. The notion of the optimal tradeoff of uncertainty has received considerable attention in studies of biological vision. This is because the “receptive fields” of single neural cells early in the visual pathways appear to approximate one or another form of the optimal tradeoff [3–10].

Here the tradeoff of uncertainties is formulated in a manner that is helpful for investigating its consequences outside of the optimum: across many scales, and for cell populations rather than for single cells. Then the question is posed of how the scales of multiple sensory cells should be selected for simultaneously minimizing the uncertainty of measurement for all the cells, on several stimulus dimensions.

This article concentrates on how visual motion can be estimated at the lowest overall uncertainty of measurement across the entire range of useful sensor sizes (in artificial systems) or the entire range of receptive fields (in biological systems). In other words, the following is an attempt to develop an economic normative theory of motion-sensitive systems. Norms are derived for efficient design of such systems, and then the norms are compared with facts of biological vision.

This approach from first principles of measurement and parsimony helps to understand the forces that shape the characteristics of biological vision, but which had appeared intractable or controversial using previous methods. These characteristics include the spatiotemporal contrast sensitivity function, adaptive transformations of this function caused by stimulus change, and also some characteristics of the higher-level perceptual processes, such as perceptual organization.

The chapter has the following structure. The uncertainty relation in one dimension is presented in Sect. 7.2, generalized to two dimensions (of space and time) in Sect. 7.3. The optimal conditions for motion measurement in view of the uncertainty are explored in Sect. 7.4. Efficient allocation of sensors according to the optimal conditions is described in Sect. 7.5. Conclusions are situated in Sect. 7.6.

7.2 Gabor's Uncertainty Relation in One Dimension

The outcomes of measuring the location and the frequency content of any signal by a single sensory device are not independent of one another. The measurement of location assigns the signal to interval Δx on some dimension of interest x . The smaller the interval the lower the uncertainty about signal location. The uncertainty is often described in terms of the precision of measurements, quantified by the dispersion of the measurement interval or, even simpler, by the size of the interval, Δx . The smaller the interval, the lower the uncertainty about location, and the higher the precision of measurement.

The measurement of frequency content evaluates how the signal varies over x , i.e., the measurement is best described on the dimension of frequency of signal variation, f_x . The measurement of frequency content is equivalent to localizing the signal on f_x : assigning the signal to some interval Δf_x . Again, the smaller the interval, the lower the uncertainty of measurement and the higher the precision.¹

The product of uncertainties about the location and frequency content of the signal is bounded “from below” [2, 11–13]. The product cannot be smaller than some positive constant C_x :

$$U_x U_f \geq C_x, \quad (7.1)$$

¹ For brevity, here “frequency content” will sometimes be shortened to “content.”

where U_x and U_f are the uncertainties about the location and frequency content of the signal, respectively, measured on the intervals Δx and Δf_x .

Equation 7.1 means that any measurement has a limit at $U_x U_f = C_x$. At the limit, decreasing one uncertainty is accompanied by increasing the other. For simplicity, let us quantify the measurement uncertainty by the size of the measurement interval. Gabor's uncertainty relation may therefore be written as

$$\Delta x \Delta f_x \geq C_x, \quad (7.2)$$

and its limiting condition as

$$\Delta x \Delta f_x = C_x. \quad (7.3)$$

Let us consider the consequences of the uncertainty relation for sensory measurement, first for single sensors (Sect. 7.2.1) and then for sensor populations (Sect. 7.2.2), which afford several benefits (Sect. 7.2.3).

7.2.1 Single Sensors

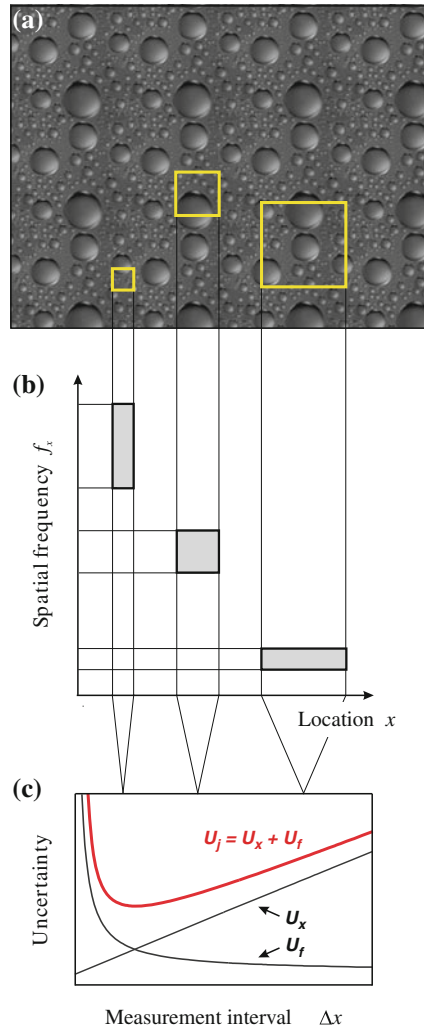
First, consider how the uncertainty relation constrains the measurements by a single measuring device: a "sensor." Figure 7.1 illustrates three spatial sensors of different sizes. In Fig. 7.1a, the measurement intervals of the sensors are defined on two spatial dimensions. For simplicity, let us consider just one spatial dimension, x , so the interval of measurement ("sensor size") is Δx .

The limiting effect of the uncertainty relation for such sensors has a convenient graphic representation called "information diagram" (Fig. 7.1b). Let the two multiplicative terms of Eq. 7.3 be represented by the two sides of a rectangle in coordinate plane (x, f_x) . Then C_x is the rectangle area. Such rectangles are called "information cells" or "logons." Three logons, of different shapes but of the same area C_x , are shown in Fig. 7.1b, representing the three sensors:

- The logon of the smallest sensor (smallest Δx , left) is thin and tall, indicating that the sensor has a high precision on x and a low precision on f_x .
- The logon of the largest sensor (right) is thick and short, indicating a low precision on x and a high precision on f_x .
- The above sensors are specialized for measuring either the location or frequency content of signals. The medium-size sensor (middle) offers a compromise: its uncertainties are not as low as the lowest uncertainties (but not as high as the highest uncertainties) of the specialized sensors. In this respect, the medium-size sensor trades one kind of uncertainty for another.

The medium-size sensors are most useful for jointly measuring the locations and frequency content of signals.

Fig. 7.1 Components of measurement uncertainty. **a** The image is sampled by three sensors of different sizes. **b** The three sensors are associated with Gabor's logons: three *rectangles* that have the same areas but different shapes, according to the limiting condition of the uncertainty relation in Eq. 7.3. **c** Functions U_x and U_f represent the uncertainties about the location and content of the measured signal (the *horizontal* and *vertical* extents of the logons in **b**, respectively), and function U_j represents the joint uncertainty about signal location and content



The ranking of sensors can be formalized using an additive model of uncertainty (Fig. 7.1c). The motivation for such an additive model is presented in Appendix 1. Let us assume that visual systems have no access to complete prior information about the statistics of measured signals (such as the joint probability density functions for the spatial and temporal locations of stimuli and their frequency content). Instead, the systems can reliably estimate only the means and variances of the measured quantities.

Accordingly, the overall uncertainty in Fig. 7.1c has the following components. The increasing function represents the uncertainty about signal location: $U_x = \Delta x$. The decreasing function represents the uncertainty about signal content:

$U_f = \Delta f_x = C_x/\Delta x$ (from Eq. 7.3). The joint uncertainty of measuring signal location and content is represented by the non-monotonic function U_j :

$$U_j = \lambda_x U_x + \lambda_f U_f = \lambda_x \Delta x + \lambda_f \frac{1}{\Delta x}, \quad (7.4)$$

where λ_x and λ_f are positive coefficients reflecting how important the components of uncertainty are relative to one another.

The additive model of Eq. 7.4 implies a worst-case estimate of the overall uncertainty (explained just below in terms of the minimax principle). The additive components are weighted, while the weights are playing several roles. They bring the components of uncertainty to the same units, allowing for different magnitude of C_x , and representing the fact that the relative importance of the components depends on the task at hand.²

The joint uncertainty function (U_j in Fig. 7.1c) has its minimum at an intermediate value of Δx . This is a point of equilibrium of uncertainties, in that a sensor of this size implements a perfect balance of uncertainties about the location and frequency content of the signal [14]. If measurements are made in the interest of high precision, and if the location and the frequency content of the signal are equally important, then a sensor of this size is the best choice for jointly measuring the location and the frequency content of the signal.

The Minimax Principle

What is the best way to allocated resources in order to reduce the chance of gross errors of measurement. One approach to solving this problem is using the minimax strategy devised in game theory for modeling choice behavior [15, 16]. Generally, the minimax strategy is used for estimating the maximal expected loss for every choice and then pursuing the choices for which the expected maximal loss is minimal. In the present case, the choice is between the sensors that deliver information with variable uncertainty.

In the following, the minimax strategy is implemented by assuming the maximal (worst-case) uncertainty of measurement across the sensors spanning the entire range of the useful spatial and temporal scales. This strategy is used in two ways. First, the consequences of Gabor's uncertainty relation are investigated under the assumption that the uncertainty of measurement is as high as possible (i.e., using the limiting case of uncertainty relation; Eq. 7.3). Second, the outcomes of measurement on different sensors are anticipated by adding their component uncertainties, i.e., using the joint uncertainty function of Eq. 7.4. (Advantages of the additive model are explained in Appendix 1.) Suppose that sensor preferences are ranked according to the expected maximal uncertainty: the lower the uncertainty, the higher the preference.

² Different criteria of measurement and sensor shapes correspond to different magnitudes of C_x .

7.2.2 Sensor Populations

Real sensory systems have at their disposal large but limited numbers of sensors. Since every sensor is useful for measuring only some aspects of the stimulus, sensory systems must solve an economic problem: they must distribute their sensors in the interest of perception of many different stimuli. Let us consider this problem using some simple arrangements of sensors.

First, consider a population of identical sensors in which the measurement intervals do not overlap. Figure 7.2a contains three examples of such sensors, using

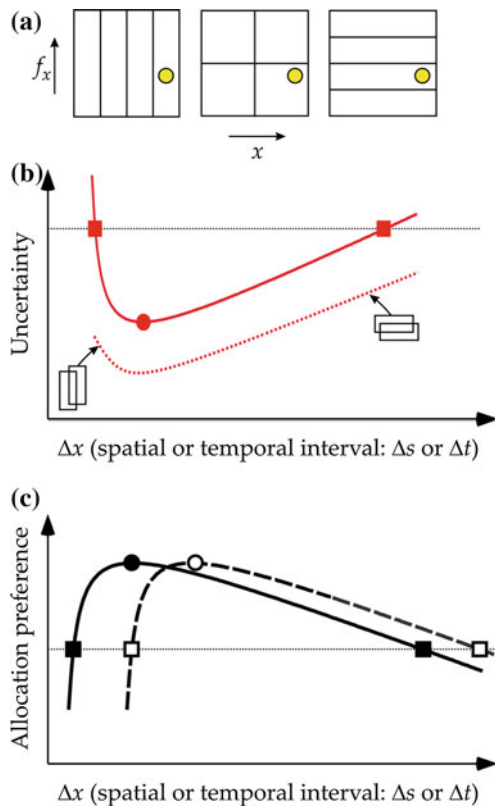


Fig. 7.2 Allocation of multiple sensors. **a** Information diagrams for a population of four sensors, using sensors of the same size within each population, and of different sizes across the populations. **b** Uncertainty functions. The *red curve* is the joint uncertainty function introduced in Fig. 7.1, with the markers indicating special conditions of measurement: the lowest joint uncertainty (the *circle*) and the equivalent joint uncertainty (the *squares*), anticipating the optimal sets and the equivalence classes of measurement in the higher-dimensional systems illustrated in Figs. 7.3 and 7.4. **c** Preference functions. The *solid curve* is a function of allocation preference (here reciprocal to the uncertainty function in **b**: an optimal distribution of sensors, expected to shift (*dashed curve*) in response to change in stimulus usefulness

the information diagram introduced in Fig. 7.1. Each of the three diagrams in Fig. 7.2a portrays four sensors, identical to one another except they are tuned to different intervals on x (which can be space or time). Each panel also contains a representation of a narrow-band signal: the yellow circle, the same across the three panels of Fig. 7.2a. The different arrangements of sensors imply different resolutions of the system for measuring the location and frequency content of the stimulus.

- The population of small sensors (small Δx on the left of Fig. 7.2a) is most suitable for measuring signal location: the test signal is assigned to the rightmost quarter on the range of interest in x . In contrast, measurement of frequency content is poor: signals presented anywhere within the vertical extent of the sensor (i.e., within the large interval on f_x) will all lead to the same response. This system has a good location resolution and poor frequency resolution.
- The population of large sensors (large Δx on the right of Fig. 7.2a) is most suitable for measuring frequency content. The test signal is assigned to a small interval on f_x . Here, measurement of location is poor. This system has a good frequency resolution and poor location resolution.
- The population of medium-size sensors can obtain useful information about both locations and frequency content of signals. It has a better frequency resolution than the population of small sensors, and a better location resolution than the population of large sensors.

Consequences of the different sensor sizes are summarized by the joint uncertainty function in Fig. 7.2b. (For non-overlapping sensors, the function has the same shape as in Fig. 7.1c). The figure makes it clear that the sensors or sensor populations with very different properties can be equivalent in terms of their joint uncertainty. For example, the two filled squares in Fig. 7.2b mark the uncertainties of two different sensor populations: one contains only small sensors and the other contains only large sensors.

The populations of sensors, in which the measurement intervals overlap, are more versatile than the populations of non-overlapping sensors. For example, the sensors with large overlapping intervals can be used to emulate measurements by the sensors with smaller intervals (Appendix 2), reducing the uncertainty of stimulus localization. Similarly, groups of overlapping sensors with small measurement intervals can emulate the measurements by sensors with larger intervals, reducing the uncertainty of identification. Overall, a population of the overlapping sensors can afford lower uncertainties across the entire range of measurement intervals, represented in Fig. 7.2b by the dotted curve: a lower-envelope uncertainty function. Still, the new uncertainty function has the same shape as the previous function (represented by the solid line) because of the limited total number of the sensors.

7.2.3 Cooperative Measurement

To illustrate the benefits of measurement using multiple sensors, suppose that the stimulation was uniform and one could vary the number of sensors in the population at will, starting with a system that has only a few sensors, toward a system that has an unlimited number of sensors.

- A system equipped with very limited resources, and seeking to measure both the location and the frequency content of signals, will have to be unmitigatedly frugal. It will use only the sensors of medium size, because only such sensors offer useful (if limited) information about both properties of signals.
- A system enjoying unlimited resources will be able to afford many specialized sensors or groups of such sensors (represented by the different information diagrams in Fig. 7.2a).
- A moderately wealthy system: a realistic middle ground between the extremes outlined above will be able to escape the straits of Gabor's uncertainty relation using different specialized sensors and thus measuring the location and content of signals with high precision.

As one considers systems with different numbers of sensors, from small to large, one expects to find an increasing ability of the system to afford the large and small measurement intervals. As the number of sensors increases, their allocation will expand in two directions, up and down on the dimension of sensor scale: from using only the medium-size sensors in the poor system, to using also the small and large sensors in the wealthier systems. This allocation policy is illustrated in Fig. 7.2c. The preference function in Fig. 7.2c indicates that, as the more useful sensors are expected to grow in number, the distribution of sensors will form a smooth function across the scales. As mentioned, the sensitivity of the system is expected to follow a function monotonically related to the preference function.

Increasing the number of sensors selective to the same stimulus condition is expected to improve sensory performance, manifested in lower sensory thresholds. One reason for such improvement in biological sensory systems is the fact that integrating information across multiple sensors will help to reduce the detrimental effect of the noisy fluctuations of neural activity, in particular when the noises are uncorrelated.

The preference function in Fig. 7.2c is exceedingly simple: it merely mirrors the joint uncertainty function of Fig. 7.2b. This example helps to illustrate some special conditions of the uncertainty of measurement and to anticipate their consequences for sensory performance. First, the minimum of uncertainty corresponds to the maximum of allocation preference, where the highest sensitivity is expected. Second, equal uncertainties correspond to equal allocation preferences, where equal sensitivities are expected. Allocation policies are considered again in Sects. 7.4 and 7.5, where the relationship is studied between a normative prescription for resource allocation and a characteristic of performance in biological vision.

7.3 Gabor’s Uncertainty in Space-Time

Let us consider how Gabor’s uncertainty generalizes to two dimensions. A two-dimensional spatiotemporal uncertainty function is introduced in Sect. 7.3.1, and the equivalence classes of uncertainty are derived in Sect. 7.3.2. The interaction between the spatial and temporal dimensions of uncertainty is discussed in Sect. 7.3.3.

7.3.1 Uncertainty in Two Dimensions

Consider a more complex case, where signals vary on two dimensions: space and time. Here, the measurement uncertainty has four components, illustrated in Fig. 7.3a. The bottom of Fig. 7.3a is a graph of the spatial and temporal sensor sizes $(T, S) = (\Delta t, \Delta s)$. Every point in this graph corresponds to a “condition of measurement” associated with the four properties of sensors.³ By Gabor’s uncertainty relation, spatial and temporal intervals $(\Delta t, \Delta s)$ are associated with, respectively, the spatial and temporal frequency intervals $(\Delta f_t, \Delta f_s)$.

The four-fold dependency is explained on the side panels of the figure using Gabor’s logons, each associated with a sensor labeled by a numbered disc. For example, in sensor 7 the spatial and temporal intervals are small, indicating a good precision of spatial and temporal localization (i.e., concerning “where” and “when” the stimuli occurs). But the spatial and temporal frequency intervals are large, indicating a low precision in measuring spatial and temporal frequency content (a low capacity to serve the “what” task of stimulus identification). This pattern is reversed in sensor 3, where the precision of localization is low but the precision of identification is high.

As in the previous example (Fig. 7.1b, c), here the one-dimensional uncertainties are summarized using joint uncertainty functions: the red curves on the side panels of Fig. 7.3b. Each function has the form of Eq. 7.4, applied separately to spatial:

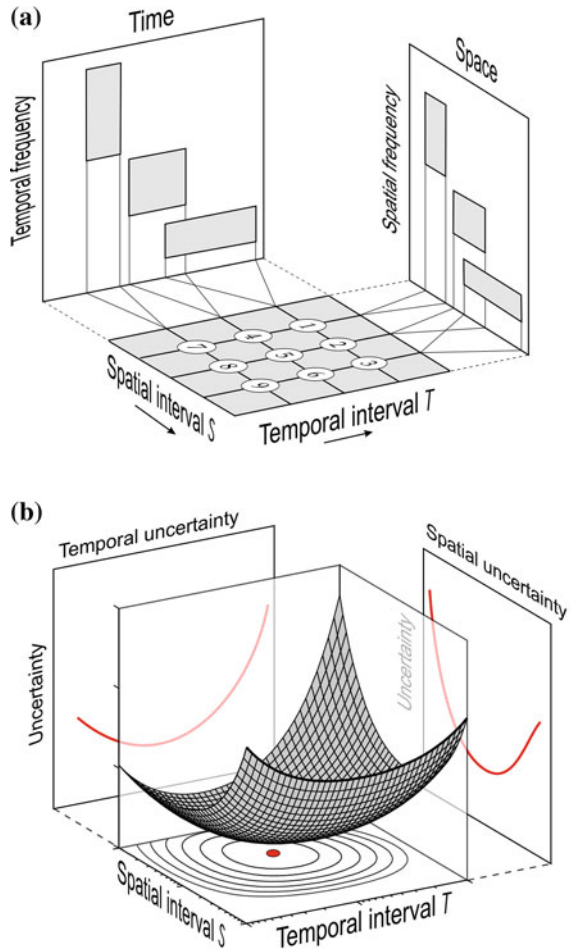
$$U_S = \lambda_1 S + \lambda_2 / S$$

and temporal:

$$U_T = \lambda_3 T + \lambda_4 / T$$

³ Here the sensors are characterized by intervals following the standard notion that biological motion sensors are maximally activated when the stimulus travels some distance Δs over some temporal interval Δt [17].

Fig. 7.3 Components of measurement uncertainty in space-time. **a** Spatial and temporal information diagrams of spatiotemporal measurements. The numbered *discs* each represents a sensor of particular spatial and temporal extent, $S = \Delta s$ and $T = \Delta t$. The *rectangles* on side panels are the spatial and temporal logons associated with the sensors. **b** The surface represents the joint uncertainty about signal location and frequency content of signals across sensors of different spatial and temporal size. The contours in the *bottom* plane (S, T) are sets of equivalent uncertainty (reproduced for further consideration in Fig. 7.4). *Panel A* is adopted from [18] and *panel B* from [19]



dimensions, where $S = \Delta s$ and $T = \Delta t$. Next, spatial and temporal uncertainties are combined for every spatiotemporal condition:

$$U_{ST} = U_T + U_S$$

to obtain a bivariate spatiotemporal uncertainty function:

$$U_{ST} = \lambda_1 S + \frac{\lambda_2}{S} + \lambda_3 T + \frac{\lambda_4}{T} \tag{7.5}$$

represented in Fig. 7.3b by a surface.

The spatiotemporal uncertainty function in Fig. 7.3b has a unique minimum, of which the projection on graph (T, S) is marked by the red dot: the point of perfect balance of the four components of measurement uncertainty. Among the conditions

of imperfect balance of uncertainties, consider the conditions of an equally imperfect balance. These are the equivalence classes of measurement uncertainty, represented by the level curves of the surface. The concentric contours on the bottom of Fig. 7.3b are the projections of some of the level curves.

7.3.2 Equivalence Classes of Uncertainty

Contours of equal measurement uncertainty are reproduced in Fig. 7.4 from the bottom of Fig. 7.3b. The pairs of connected circles indicate that the slopes of equivalence contours vary across the conditions of measurement. This fact has several interesting implications for the perception of visual motion.

First, if the equivalent conditions of motion perception were consistent with the equivalent conditions of uncertainty, then some lawful changes in the perception of motion would be expected for stimuli that activate sensors in different parts of the sensor space. This prediction was confirmed in studies of apparent motion, which is the experience of motion from discontinuous displays, where the sequential views of the moving objects (the “corresponding image parts”) are separated by spatial (σ) and temporal (τ) distances. Perceptual strength of apparent motion in such displays was conserved: sometimes by changing σ and τ in the same direction (both increasing or both decreasing), which is the regime of space-time coupling [22], and sometimes by trading off one distance for another: the regime of space-time tradeoff [23]. Gepshtein and Kubovy [20] found that the two regimes of apparent motion were special cases of a lawful pattern: one regime yielded to another as a function of speed, consistent with the predictions illustrated in Fig. 7.4.

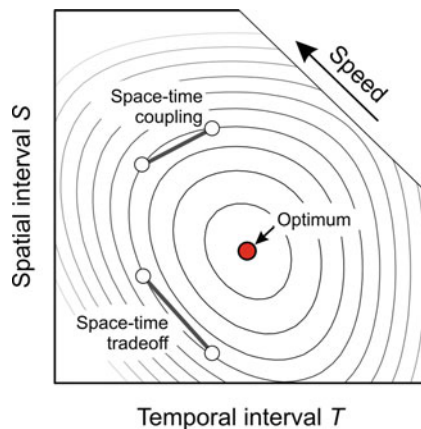


Fig. 7.4 Equivalence classes of uncertainty. The contours represent equal measurement uncertainty (reproduced from the *bottom panel* of Fig. 7.3b and the *red circle* represents the minimum of uncertainty. The pairs of connected *circles* labeled “space-time coupling” and “space-time tradeoff” indicate why some studies of apparent motion discovered different regimes of motion perception in different stimuli [20, 21]

Second, the regime of space-time coupling undermines one of the cornerstones of the literature on visual perceptual organization: the proximity principle of perceptual grouping [24, 25]. The principle is an experimental observations from the early days of the Gestalt movement, capturing the common observation that the strength of grouping between image parts depends on their distance: the shorter the distance the stronger the grouping. In space-time, the principle would hold, if the strength of grouping had not changed, when increasing one distance (σ or τ) was accompanied by decreasing the other distance (τ or σ): the regime of tradeoff [26]. The fact that the strength of grouping is maintained by increasing both σ and τ , or by decreasing both σ and τ , is inconsistent with the proximity principle [21].

7.3.3 Spatiotemporal Interaction: Speed

Now let us consider the interaction of the spatial and temporal dimensions of measurement. A key aspect of this interaction is the speed of stimulus variation: the rate of temporal change of stimulus intensity across spatial location. The dimension of speed has been playing a central role in the theoretical and empirical studies of visual perception [17, 27, 28]. Not only is the perception of speed crucial for the survival of mobile animals, but it also constitutes a rich source of auxiliary information for parsing the optical stimulation [29, 30].

What is more, speed appears to play the role of a control parameter in the organization of visual sensitivity. The shape of a large-scale characteristic of visual sensitivity (measured using continuous stimuli) is invariant with respect to speed [31, 32]. And a characteristic of the strength of perceived motion in discontinuous stimuli (giving rise to “apparent motion”) collapse onto a single function, when plotted against speed [20].

From the present normative perspective, the considerations of speed measurement (combined with the foregoing considerations of measuring the location and frequency content) of visual stimuli have two pervasive consequences, which are reviewed in some detail next. First, in a system optimized for the measurement of speed, the expected distribution of the quality of measurement has an invariant shape, distinct from the shape of such a distribution conceived before one has taken into account the measurement of speed (Fig. 7.4). Second, the dynamics of visual measurement, and not only its static organization, will depend on the manner of interaction of the spatial and temporal aspects of measurement.

In Figs. 7.3 and 7.4, a distribution of the expected uncertainty of measurement was derived from a local constraint on measurement. The local constraint was defined separately for the spatial and temporal intervals of the sensor. The considerations of speed measurement add another constraint, which has to do with the relationship between the spatial and temporal intervals.

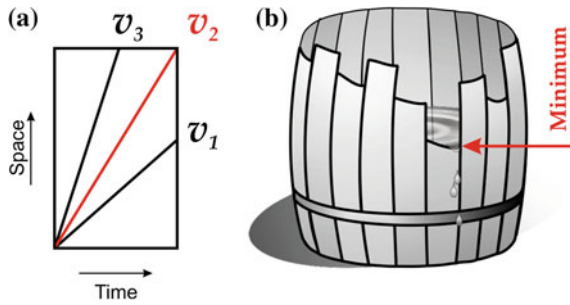


Fig. 7.5 Economic measurement of speed. **a** The *rectangle* represents a sensor defined by spatial and temporal intervals (S and T). From considerations of parsimony, the sensor is more suitable for measurement of speed $v_2 = S/T$ than v_1 or v_3 since no part of S or T is wasted in measurement of v_2 . **b** Liebig's barrel. The *shortest stave* determines barrel's capacity. Parts of *longer staves* are wasted since they do not affect the capacity

The ability to measure speed by a sensor defined by spatial and temporal intervals depends on the extent of these intervals. As it is shown in Fig. 7.5a, different ratios of the spatial extent to the temporal extent make the sensor differently suitable for measuring different magnitudes of speed.

This argument is one consequence of the Law of The Minimum [33], illustrated in Fig. 7.5b using Liebig's barrel. A broken barrel with the staves of different lengths can hold as much content as the shortest stave allows. Using the staves of different lengths is wasteful because a barrel with all staves as short as the shortest stave would do just as well. In other words, the barrel's capacity is limited by the shortest stave.

Similarly, a sensor's capacity for measuring the speed is limited by the extent of its spatial and temporal intervals. The capacity is not used fully, if the spatial and temporal projections of vector v are larger or smaller than the spatial and temporal extents allow (v_1 and v_3 in Fig. 7.5b). Just as the extra length of the long staves is wasted in the Liebig's barrel, the spatial extent of the sensor is wasted in measurement of v_1 and the temporal extent is wasted in measurement of v_3 . Let us therefore start with the assumption that the sensor defined by the intervals S and T is best suited for measuring speed $v = S/T$.

7.4 Optimal Conditions for Motion Measurement

Several special conditions of motion measurements — the minima of uncertainty and the shape of the optimal set of measurement — are introduced, respectively, in Sects. 7.4.1 and 7.4.2.

7.4.1 Minima of Uncertainty

The optimal conditions of measurement are expected, where the measurement uncertainty is the lowest. Using a shorthand notation for the spatial and temporal partial derivatives of U_{ST} in Eq. 7.4, $\partial U_S = \partial U_{ST}/\partial S$ and $\partial U_T = \partial U_{ST}/\partial T$, the minimum of measurement uncertainty is the solution of

$$\partial U_T dT + \partial U_S dS = 0. \quad (7.6)$$

The optimal condition for the entire space of sensors, disregarding individual speeds, is marked as the red point in Fig. 7.4. To find the minima for specific speeds v_i , let us rewrite Eq. 7.6 such that speed appears in the equation as an explicit term. By dividing each side of Eq. 7.6 by dT , and using the fact that $v_i = dS/dT$, Eq. 7.6 becomes

$$\partial U_S v_i + \partial U_T = 0. \quad (7.7)$$

The solution of Eq. 7.7 is a set of optimal conditions of measurement across speeds. To illustrate the solution graphically, consider the vector form of Eq. 7.7, i.e., the scalar product

$$\left\langle \mathbf{g}_{(T,S)}, \mathbf{v}_{(T,S)} \right\rangle = 0, \quad (7.8)$$

where the first term is the gradient of measurement uncertainty function

$$\mathbf{g}_{(T,S)} = (\partial u_T, \partial u_S) \quad (7.9)$$

and the second term is the speed

$$\mathbf{v}_{(T,S)} = (T, vT) \quad (7.10)$$

for sensors with parameters (T, S) . For now, assume that the speed, to which a sensor is tuned is the ratio of spatial to temporal intervals ($v = S/T$) that define the logon of the sensor. (Normative considerations of speed tuning are reviewed in Sect. 7.3.3.)

The two terms of Eq. 7.8 are shown in Fig. 7.6: separately in panels A-B and together in panel C. The blue curve in panel C represents the set of conditions where vectors \mathbf{v} and \mathbf{g} are orthogonal to one another, satisfying Eq. 7.8. This curve is the “optimal set” for measuring speed while minimizing the uncertainty about signal location and content.

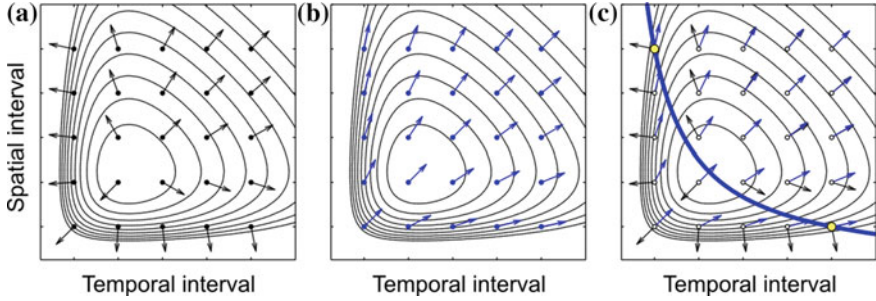


Fig. 7.6 Graphical solution of Eq. 7.8 without integration of speed. **a** Local gradients of measurement uncertainty g . **b** Speeds v , to which the different sensors are tuned. **c** Optimal conditions (blue curve) arise, where g and v are orthogonal to one another (Eq. 7.8). The yellow circles are two examples of locations, where the requirement of orthogonality is satisfied. (Arrow lengths are normalized to avoid clutter.)

7.4.2 The Shape of Optimal Set

The solution of Eq. 7.8 was derived for speed defined at every point in the space of intervals (T,S): the blue arrows in Fig. 7.6b. This picture is an abstraction that disregards the fact that measurements are performed while the sensors integrate stimulation over sensor extent. The solution of Eq. 7.8 that takes this fact into account is described in Fig. 7.7. The integration reduces differences between the directions of adjacent speed vectors (panel B), and so the condition of orthogonality of g and v is satisfied at locations other than in Fig. 7.6.

The red curve Fig. 7.7c is the “integral” optimal set for measuring speed. This figure presents an extreme case, where speeds are integrated across the entire range of stimulation, as if every sensor had access to the expected speed of stimulation across the entire range of stimulus speed:

$$v_e = \int_0^\infty p(v)v dv, \tag{7.11}$$

where $p(v)$ is the distribution of speed in the stimulation. At this extreme, every v is co-directional with the expected speed.

In comparison to the local optimal set (the blue curve in Fig. 7.7c), many points of the integral optimal set (the red curve) are shifted away from the origin of the parameter space. The shift is small in the area of expected speed v_e (the black line in Fig. 7.8), yet the shift increases away from the expected speed, such that the integral optimal set has the shape of a hyperbola.

The position of the optimal set in the parameter space depends on the prevailing speed of stimulation [19], as Fig. 7.8 illustrates. This dependence is expected to be more pronounced in cases where the integration by receptive fields is large.

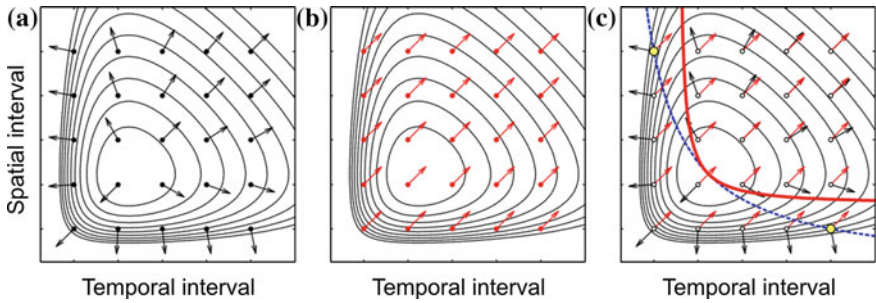


Fig. 7.7 Graphical solution of Eq. 7.8 with integration of speed. **a** Local gradient of measurement uncertainty \mathbf{g} as in Fig. 7.6a. **b** Speeds \mathbf{v} integrated across multiple speeds. **c** Now the optimal conditions (*red curve*) arise at locations different from those in Fig. 7.6 (the *blue curve* is a copy from Fig. 7.6c)

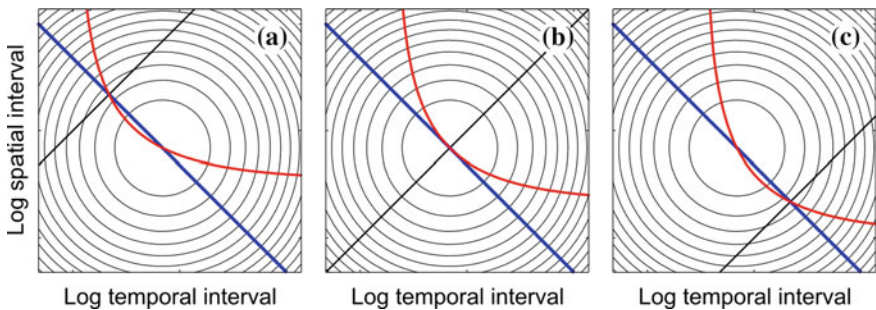


Fig. 7.8 Effect of expected stimulus speed. The *red* and *blue* curves are the optimal sets derived in Figs. 7.6 and 7.7, now shown in logarithmic coordinates to emphasize that the “integral” optimal set (*red*) has the invariant shape of a *rectangular hyperbola*, whereas the “local” optimal set (*blue*) does not. From **a** to **c**, the expected stimulus speed (Eq. 7.11) decreases, represented by the *black lines*. The position of the integral optimal set changes accordingly

To summarize, the above argument has been concerned with how speed integration affects the optimal conditions for speed measurement. At one extreme, with no integration, the set of optimal conditions could have any shape. At the other extreme, with the scope of integration maximally large, the optimal set is a hyperbola. In between, the larger the scope of integration, the more the optimal set resembles a hyperbola. The position of this hyperbola in the parameter space depends on the prevalent speed of stimulation.

This argument has two significant implications. First, the distribution of resources in the visual system is predicted to have an invariant shape, which is consistent with results of measurements in biological vision (Fig. 7.9) using a variety of psychophysical tasks and stimuli [27, 34–36]. Second, it implies that changes in statistics of stimulation will have a predictable effect on allocation of resources, helping the systems adapt to the variable stimulation, a theme developed in the next section.

7.5 Sensor Allocation

In this section, let us consider adaptive allocation of sensors: its norms (Sect. 7.5.1) and mechanisms (Sect. 7.5.2).

7.5.1 Adaptive Allocation

Allocation of sensors is likely to depend on several factors that determine sensor usefulness, such as sensory tasks and properties of stimulation. For example, when the organism needs to identify rather than localize the stimulus, large sensors are more useful than small ones. Allocation of sensors by their usefulness is therefore expected to shift, for example as shown in Fig. 7.2c.

Such shifts of allocation are expected also because the environment is highly variable. To insure that sensors are not allocated to stimuli that are absent or useless, biological systems must monitor their environment and the needs of measurement. As the environment or needs change, the same stimuli become more or less useful. The system must be able to reallocate its resources: change properties of sensors such as to better measure useful stimuli.

Because of the large but limited pool of sensors at their disposal, real sensory systems occupy a middle ground between the extremes of sensor “wealth.” Such systems can afford some specialization but they cannot be wasteful. They are therefore subject to Gabor’s uncertainty relation, but they can alleviate consequences of the uncertainty relation, selectively and to some extent, by allocating sensors to some important classes of stimuli. Allocation preferences of such systems is expected to look like that in Fig. 7.2c, yet generalized to multiple stimulus dimensions.

To summarize, the above analysis suggests that sensory systems are shaped by constraints of measurement and the economic constraint of limited resources. This is because the sensors of different sizes are ordered according to their usefulness in terms of Gabor’s uncertainty relation. These considerations are simple preceding in the one-dimensional analysis undertaken so far. In the more complex case considered in the next section, this approach leads to nontrivial conclusions. In particular, this approach helps to explain several puzzling phenomena in perception of motion and in motion adaptation.

The preceding analysis has led to a prescription for how receptive fields of different spatial and temporal extents ought to be distributed across the full range of visual stimuli. By this prescription, changes in usefulness of stimuli are expected to cause changes in receptive field allocation. Now consider some specific predictions of how the reallocation of resources is expected to bring about systematic changes in spatiotemporal visual sensitivity. Because the overall amount of resources in the system is limited, an improvement of visual performance (such as a higher sensitivity) at some conditions will be accompanied by a deterioration of performance

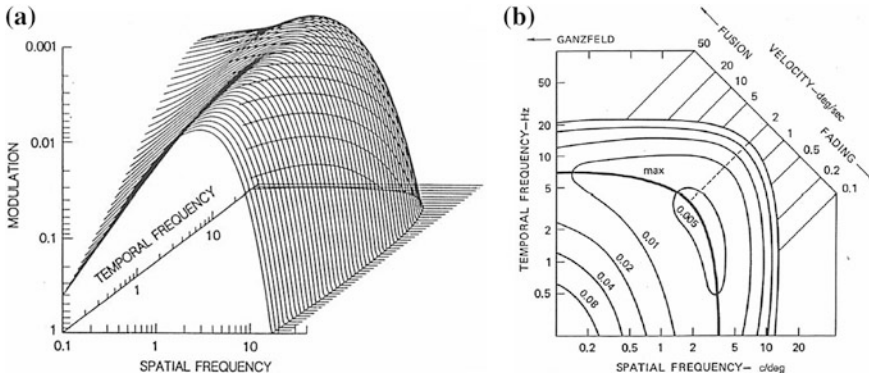


Fig. 7.9 Human spatiotemporal contrast sensitivity function, shown as a surface in **a** and a contour plot in **b**. Conditions of maximal sensitivity across speeds form the *thick curve* labeled “max.” The maximal sensitivity set has the shape predicted by the normative theory: the *red curve* in Fig. 7.7. The mapping from measurement intervals to stimulus frequencies is explained in [19, 27]. Both panels are adopted from [31]

(a lower sensitivity) at other conditions, leading to counterintuitive patterns of sensitivity change.

Let us assume should be allocated to the equally useful stimuli, when certain speeds become more prevalent or more important for perception than other speeds period. Then, the visual system is expected to allocate more resources to the more important speeds. For example, Fig. 7.10a, b contains maps of spatiotemporal sensitivity computed for two environments, with high and low prevailing speeds. Figure 7.10c is a summary of differences between the sensitivity maps. The predicted changes form well-defined foci of increased performance and large areas of decreased performance. Gepshtein et al. [37] used intensive psychophysical methods [38] to measure the entire spatiotemporal contrast sensitivity function in different statistical “contexts” of stimulation. They found that sensitivity changes were consistent with the predictions illustrated in Fig. 7.10.

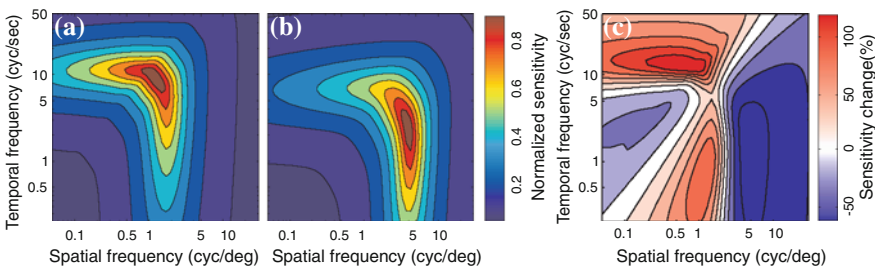


Fig. 7.10 Predictions for adaptive reallocation of sensors. **a, b** Sensitivity maps predicted for two stimulus contexts: dominated by high speed in **a** and low speed in **b**. The *color* stands for normalized sensitivity. **c** Sensitivity changes computed as $100 \times a/b$, where *a* and *b* are map entries in **a** and **b**, respectively. Here, the *color* stands for sensitivity change: *gain* in *red* and *loss* in *blue*

These results suggest a simple resolution to some long-standing puzzles in the literature on motion adaptation. In the early theories, adaptation was viewed as a manifestation of neural fatigue. Later theories were more pragmatic, assuming that sensory adaptation is the organism's attempt to adjust to the changing environment [39–42]. But evidence supporting this view has been scarce and inconsistent. For example, some studies showed that perceptual performance improved at the adapting conditions, but other studies reported the opposite [43, 44]. Even more surprising were systematic changes of performance for stimuli very different from the adapting ones [44]. According to the present analysis, such local gains and losses of sensitivity are expected in a visual system that seeks to allocate its limited resources in face of uncertain and variable stimulation (Fig. 7.10). Indeed, the pattern of gains and losses of sensitivity manifests an optimal adaptive behavior.

This example illustrates that in a sensory system with scarce resources, optimization of performance will lead to a reduction of sensitivity to some stimuli. This phenomenon is not unique to sensory adaptation [45]. For example, demanding tasks may cause impairment of visual performance for some stimuli, as a consequence of the task-driven reallocation of visual resources [46, 47].

7.5.2 Mechanism of Adaptive Allocation

From the above it follows that the shape of the spatiotemporal sensitivity function, and also transformations of this function, can be understood by studying the uncertainties implicit to visual measurement. This idea received further support from simulations of a visual system equipped with thousands of independent (uncoupled) sensors, each having a spatiotemporal receptive field [48, 49].

In these studies, spatiotemporal signals were sampled from known statistical distributions. Receptive fields parameters were first distributed at random. They were then updated according to a generic rule of synaptic plasticity [50–53]. The changes of receptive fields amounted to small random steps in the parameter space, modeled as stochastic fluctuations of the spatial and temporal extents of the receptive fields. Step length was proportional to the (local) uncertainty of measurement by individual receptive fields. The steps were small, where the uncertainty was low, and the receptive fields changed little. Where the uncertainty was high, the steps were larger, so the receptive fields tended to escape the high-uncertainty regions. This stochastic behavior led to a “drift” of receptive fields in the direction of low uncertainty of measurement [49], predicted by standard stochastic methods [54], as if the system sought stimuli that could be measured reliably (*cf.* [55]).

Remarkably, the independent stochastic changes of receptive fields (their uncoupled “stochastic tuning”) steered the system toward the distribution of receptive field parameters predicted by the normative theory described in Sect. 7.4, and leading to the distribution of sensitivity observed in human vision (Fig. 7.9). When the distribution of stimuli changed, mimicking a change of sensory environment, the system was able to spontaneously discover an arrangement of sensors

optimal for the new environment, in agreement with the predictions illustrated in Fig. 7.10 [56]. This is an example of how efficient allocation of resources can emerge in sensory systems by way of self-organization, enabling a highly adaptive sensory behavior in face of the variable (and sometimes unpredictable) environment.

7.6 Conclusions

A study of allocation of limited resources for motion sensing across multiple spatial and temporal scales revealed that the optimal allocation entails a shape of the distribution of sensitivity similar to that found in human visual perception. The similarity suggested that several previously puzzling phenomena of visual sensitivity, adaptation, and perceptual organization have simple principled explanations. Experimental studies of human vision have confirmed the predictions for sensory adaptation. Since the optimal allocation is readily implemented in self-organizing neural networks by means of unsupervised learning and stochastic optimization, the present approach offers a framework for neuromorphic design of multiscale sensory systems capable of automated efficient tuning to the varying optical environment.

Acknowledgments This work was supported by the European Regional Development Fund, National Institutes of Health Grant EY018613, and Office of Naval Research Multidisciplinary University Initiative Grant N00014-10-1-0072.

Appendices

Appendix 1. Additivity of Uncertainty

For the sake of simplicity, the following derivations concern the stimuli that can be modeled by integrable functions $I : \mathbb{R} \rightarrow \mathbb{R}$ of one variable x . Generalizations to functions of more than one variable are straightforward. Consider two quantities:

- Stimulus location on x , where x can be space or time, and the “location” indicates respectively “where” or “when” the stimulus has occurred.
- Stimulus content on f_x , where f_x can be spatial or temporal frequency of stimulus modulation.

Suppose a sensory system is equipped with many measuring devices (“sensors”), each used to estimate both stimulus location and frequency content from “image” (or “input”) $I(x)$. Assume that the outcome of measurement is a random variable with probability density function $p(x, f)$. Let

$$\begin{aligned} p_x(x) &= \int p(x,f) df, \\ p_f(f) &= \int p(x,f) dx \end{aligned} \quad (7.12)$$

be the (marginal) means of $p(x,f)$ on dimensions x and f_x (abbreviated as f).

It is sometimes assumed that sensory systems “know” $p(x,f)$, which is not true in general. Generally, one can only know (or guess) some properties of $p(x,f)$, such as its mean and variance. Reducing the chance of gross error due to the incomplete information about $p(x,f)$ is accomplished by a conservative strategy: finding the minima on the function of maximal uncertainty, i.e., using a minimax approach [15, 16].

The minimax approach is implemented in two steps. The first step is to find such $p_x(x)$ and $p_f(f)$, for which measurement uncertainty is maximal. (The uncertainty is characterized conservatively, in terms of variance alone [2]). The second step is to find the condition(s), at which the function of maximal uncertainty has the smallest value: the minimax point(s).

Maximal uncertainty is evaluated using the well-established definition of entropy [58] (cf. [59, 60]):

$$H(X, F) = - \int p(x,f) \log p(x,f) dx df.$$

According to the independence bound on entropy (Theorem 2.6.6 in [61]),

$$H(X, F) \leq H(X) + H(F) = H^*(X, F), \quad (7.13)$$

where

$$\begin{aligned} H(X) &= - \int p_x(x) \log p_x(x) dx, \\ H(F) &= - \int p_f(f) \log p_f(f) df. \end{aligned}$$

Therefore, the uncertainty of measurement cannot exceed

$$\begin{aligned} H^*(X, F) &= - \int p_x(x) \log p_x(x) dx \\ &\quad - \int p_f(f) \log p_f(f) df. \end{aligned} \quad (7.14)$$

Eq. 7.14 is the “envelope” of maximal measurement uncertainty: a “worst-case” estimate.

By the Boltzmann theorem on maximum-entropy probability distributions [61], the maximal entropy of probability densities with fixed means and variances is attained, when the functions are Gaussian. Then, the maximal entropy is a sum of their variances [61] and

$$\begin{aligned} p_x(x) &= \frac{1}{\sigma_x \sqrt{2\pi}} e^{-x^2/2\sigma_x^2}, \\ p_f(f) &= \frac{1}{\sigma_f \sqrt{2\pi}} e^{-f^2/2\sigma_f^2}, \end{aligned}$$

where σ_x and σ_f are the standard deviations. Then maximal entropy is

$$H = \sigma_x^2 + \sigma_f^2. \quad (7.15)$$

That is, when $p(x, f)$ is unknown, and all one knows about marginal distributions $p_x(x)$ and $p_f(f)$ is their means and variances, the maximal uncertainty of measurement is the sum of variances of the estimates of x and f . The following *minimax* step is to find the conditions of measurement, at which the sum of variances is the smallest.

Appendix 2. Improving Resolution by Multiple Sampling

How does an increased allocation of resources to a specific condition of measurement affect the (spatial or temporal) resolution at that condition? Consider set Ψ of sampling functions

$$\psi(s\sigma + \delta), \quad \sigma \in \mathbb{R}, \quad \sigma > 0, \quad \delta \in \mathbb{R},$$

where σ is a scaling parameter and δ is a translation parameter. For a broad class of functions $\psi(\cdot)$, any element of Ψ can be obtained by addition of weighted and shifted $\psi(s)$. The following argument proves that *any* function from a sufficiently broad class that includes $\psi(s\sigma + \delta)$ can be represented by a weighted sum of translated replicas of $\psi(s)$.

Let $\psi^*(s)$ be a continuous function that can be expressed as a sum of a converging series of harmonic functions:

$$\psi^*(s) = \sum_i a_i \cos(\omega_i s) + b_i \sin(\omega_i s).$$

For example, Gaussian sampling functions of arbitrary widths can be expressed as a sum of $\cos(\cdot)$ and $\sin(\cdot)$. Let us show that, if $|\psi(s)|$ is Riemann-integrable, i.e., if

$$-\infty < \int_{-\infty}^{\infty} \psi(s) |ds| < \infty$$

and its Fourier transform $\widehat{\psi}$ does not vanish for all $\omega \in \mathbb{R}$: $\widehat{\psi}(\omega) \neq 0$ (i.e., its spectrum has no “holes”), then the following expansion of ψ^* is possible:

$$\psi^*(s) = \sum_i c_i \psi(s + d_i) + \varepsilon(s), \quad (7.16)$$

where $\varepsilon(s)$ is a residual that can be arbitrarily small. This goal is attained by proving identities

$$\begin{aligned}\cos(\omega_0 s) &= \sum_i c_{i,1} \psi(s + d_{i,1}) + \varepsilon_1(s), \\ \sin(\omega_0 s) &= \sum_i c_{i,2} \psi(s + d_{i,2}) + \varepsilon_2(s),\end{aligned}\tag{7.17}$$

where $c_{i,1}$, $c_{i,2}$ and $d_{i,1}$, $d_{i,2}$ are real numbers, while $\varepsilon_1(s)$ and $\varepsilon_2(s)$ are arbitrarily small residuals.

First, write the Fourier transform of $\psi(s)$ as

$$\widehat{\psi}(\omega) = \int_{-\infty}^{\infty} \psi(s) e^{-i\omega s} ds$$

and multiply both sides of the above expression by $e^{i\omega_0 v}$:

$$e^{i\omega_0 v} \widehat{\psi}(\omega) = e^{i\omega_0 v} \int_{-\infty}^{\infty} \psi(s) e^{-i\omega s} ds = \int_{-\infty}^{\infty} \psi(s) e^{-i(\omega s - \omega_0 v)} ds.\tag{7.18}$$

Change the integration variable:

$$x = \omega s - \omega_0 v \Rightarrow dx = \omega ds, \quad s = \frac{x + \omega_0 v}{\omega},$$

such that Eq. 7.18 transforms into

$$e^{i\omega_0 v} \widehat{\psi}(\omega) = \frac{1}{\omega} \int_{-\infty}^{\infty} \psi\left(\frac{x + \omega_0 v}{\omega}\right) e^{-ix} dx.$$

Notice that $\widehat{\psi}(\omega) = a(\omega) + ib(\omega)$. Hence

$$e^{i\omega_0 v} \widehat{\psi}(\omega) = e^{i\omega_0 v} (a(\omega) + ib(\omega)) = (\cos(\omega_0 v) + i \sin(\omega_0 v))(a(\omega) + ib(\omega))$$

and

$$e^{i\omega_0 v} \widehat{\psi}(\omega) = (\cos(\omega_0 v)a(\omega) - \sin(\omega_0 v)b(\omega)) + i(\cos(\omega_0 v)b(\omega) + \sin(\omega_0 v)a(\omega)).$$

Since $\widehat{\psi}(\omega) \neq 0$ is assumed for all $\omega \in \mathbb{R}$, then $a(\omega) + ib(\omega) \neq 0$. In other words, either $a(\omega) \neq 0$ or $b(\omega) \neq 0$ should hold. For example, suppose that $a(\omega) \neq 0$. Then

$$\operatorname{Re}\left(e^{i\omega_0 v} \widehat{\psi}(\omega)\right) + \frac{b(\omega)}{a(\omega)} \operatorname{Im}\left(e^{i\omega_0 v} \widehat{\psi}(\omega)\right) = \cos(\omega_0 v) \left(\frac{a^2(\omega) + b^2(\omega)}{a(\omega)}\right).$$

Therefore,

$$\begin{aligned} \cos(\omega_0 v) &= \left(\frac{a(\omega)}{a^2(\omega) + b^2(\omega)}\right) \operatorname{Re}\left(\frac{1}{\omega} \int_{-\infty}^{\infty} \psi\left(\frac{x + \omega_0 v}{\omega}\right) e^{-ix} dx\right) \\ &\quad + \left(\frac{b(\omega)}{a^2(\omega) + b^2(\omega)}\right) \operatorname{Im}\left(\frac{1}{\omega} \int_{-\infty}^{\infty} \psi\left(\frac{x + \omega_0 v}{\omega}\right) e^{-ix} dx\right). \end{aligned} \quad (7.19)$$

Because function $\psi(s)$ is Riemann-integrable, the integrals in Eq. 7.19 can be approximated as

$$\operatorname{Re}\left(\frac{1}{\omega} \int_{-\infty}^{\infty} \psi\left(\frac{x + \omega_0 v}{\omega}\right) e^{-ix} dx\right) = \frac{\Delta}{\omega} \sum_{k=1}^N \psi\left(\frac{x_k + \omega_0 v}{\omega}\right) \cos(x_k) + \frac{\bar{\varepsilon}_1(v, N)}{\omega}, \quad (7.20)$$

$$\operatorname{Im}\left(\frac{1}{\omega} \int_{-\infty}^{\infty} \psi\left(\frac{x + \omega_0 v}{\omega}\right) e^{-ix} dx\right) = \frac{\Delta}{\omega} \sum_{p=1}^N \psi\left(\frac{x_p + \omega_0 v}{\omega}\right) \sin(x_p) + \frac{\bar{\varepsilon}_2(v, N)}{\omega}, \quad (7.21)$$

where x_k and x_p are some elements of \mathbb{R} .

From Eqs. 7.19–7.21 it follows that

$$\cos(\omega_0 v) = \sum_{j=1}^{2N} c_{j,1} \psi\left(\frac{\omega_0 v}{\omega} + d_{j,1}\right) + \varepsilon_1(v, N).$$

Given that $\widehat{\psi}(\omega) \neq 0$ for all ω and letting $\omega = \omega_0$, it follows that

$$\cos(\omega_0 v) = \sum_{j=1}^{2N} c_{j,1} \psi(v + d_{j,1}) + \varepsilon_1(v, N), \quad (7.22)$$

where

$$\frac{\bar{\varepsilon}_1(v, N)}{\omega_0} \frac{a(\omega_0)}{a^2(\omega_0) + b^2(\omega_0)} + \frac{\bar{\varepsilon}_2(v, N)}{\omega_0} \frac{b(\omega_0)}{a^2(\omega_0) + b^2(\omega_0)} \quad (7.23)$$

An analogue of Eq. 7.22 for $\sin(\omega_0 v)$ follows from $\sin(\omega_0 v) = \cos(\omega_0 v + \pi/2)$. This completes the proof of Eq. 7.17 and hence of Eq. 7.16.

References

1. Marr D (1982) Vision: a computational investigation into the human representation and processing of visual information. W. H. Freeman, San Francisco
2. Gabor D (1946) Theory of communication. *Inst Electr Eng*, Part III 93:429–457
3. Marcelja S (1980) Mathematical description of the response by simple cortical cells. *J Opt Soc Am* 70:1297–1300
4. MacKay DM (1981) Strife over visual cortical function. *Nature* 289:117–118
5. Daugman JG (1985) Uncertainty relation for the resolution in space spatial frequency, and orientation optimized by two-dimensional visual cortex filters. *J Opt Soc Am A* 2(7):1160–1169
6. Glezer VD, Gauzel'man VE, Iakovlev VV (1986) Principle of uncertainty in vision. *Neirofiziologija [Neurophysiology]* 18(3):307–312
7. Field DJ (1987) Relations between the statistics of natural images and the response properties of cortical cells. *J Opt Soc Am A* 4:2379–2394
8. Jones A, Palmer L (1987) An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *J Neurophysiol* 58:1233–1258
9. Simoncelli EP, Olshausen B (2001) Natural image statistics and neural representation. *Annu Rev Neurosci* 24:1193–1216
10. Saremi S, Sejnowski TJ, Sharpee TO (2013) Double-Gabor filters are independent components of small translation-invariant image patches. *Neural Comput* 25(4):922–939
11. Gabor D (1952) Lectures on communication theory. Technical report 238, MIT Research Laboratory of Electronics, Cambridge, MA, USA
12. Resnikoff HL (1989) The illusion of reality. Springer, New York
13. MacLennan B (1994) Gabor representations of spatiotemporal visual images. Technical report. University of Tennessee, Knoxville, TN, USA
14. Gepshtein S, Tyukin I (2006) Why do moving things look as they do? vision. *J Vis Soc Jpn*, Supp. 18:64
15. von Neumann J (1928) Zur Theorie der Gesellschaftsspiele. [On the theory of games of strategy]. *Mathematische Annalen* 100 (1928):295–320, English translation in [57]
16. Luce RD, Raiffa H (1957) Games and decisions. John Wiley, New York
17. Watson AB, Ahumada AJ (1985) Model of human visual-motion sensing. *J Opt Soc Am A* 2(2):322–341
18. Gepshtein S (2010) Two psychologies of perception and the prospect of their synthesis. *Philos Psychol* 23:217–281
19. Gepshtein S, Tyukin I, Kubovy M (2007) The economics of motion perception and invariants of visual sensitivity. *J Vis* 7, 8(8):1–18
20. Gepshtein S, Kubovy M (2007) The lawful perception of apparent motion. *J Vis* 7, 8(9):1–15
21. Gepshtein S, Tyukin I, Kubovy M (2011) A failure of the proximity principle in the perception of motion. *Humana Mentis* 17:21–34
22. Korte A (1915) Kinematoskopische Untersuchungen. *Zeitschrift für Psychologie* 72:194–296
23. Burt P, Sperling G (1981) Time, distance, and feature tradeoffs in visual apparent motion. *Psychol Rev* 88:171–195

24. Wertheimer M (1923) Untersuchungen zur Lehre von der Gestalt, II. *Psychologische Forschung* 4:301–350
25. Kubovy M, Holcombe AO, Wagemans J (1998) On the lawfulness of grouping by proximity. *Cogn Psychol* 35:71–98
26. Koffka K (1935/1963) *Principles of Gestalt psychology*. A Harbinger Book, Harcourt, Brace and World, Inc., New York
27. Nakayama K (1985) Biological image motion processing: a review. *Vis Res* 25(5):625–660
28. Weiss Y, Simoncelli EP, Adelson EH (2002) Motion illusions as optimal percepts. *Nat Neurosci* 5(6):598–604
29. Longuet-Higgins HC, Prazdny, K (1981) The interpretation of a moving retinal image. *Proc Roy Soc London. Ser B, Biol Sci* 208(1173):385–397
30. Landy MS, Maloney LT, Johnsten E, Young M (1995) Measurement and modeling of depth cue combinations: in defense of weak fusion. *Vis Res* 35:389–412
31. Kelly DH (1979) Motion and vision II. Stabilized spatio-temporal threshold surface. *J Opt Soc Am* 69(10):1340–1349
32. Kelly DH (1994) Eye movements and contrast sensitivity. In: Kelly DH (ed) *Visual science and engineering (models and applications)*. Marcel Dekker Inc, New York, pp 93–114
33. Gorban A, Pokidysheva L, Smirnova E, Tyukina T (2011) Law of the minimum paradoxes. *Bull Math Biol* 73(9):2013–2044
34. van Doorn AJ, Koenderink JJ (1982) Temporal properties of the visual detectability of moving spatial white noise. *Exp Brain Res* 45:179–188
35. van Doorn AJ, Koenderink JJ (1982) Spatial properties of the visual detectability of moving spatial white noise. *Exp Brain Res* 45:189–195
36. Laddis P, Lesmes LA, Gepshtein S, Albright TD (2011) Efficient measurement of spatiotemporal contrast sensitivity in human and monkey. In: 41st annual meeting of the society for neuroscience, vol 577.20
37. Gepshtein S, Lesmes LA, Albright TD (2013) Sensory adaptation as optimal resource allocation. *Proc Natl Acad Sci, USA* 110(11):4368–4373
38. Lesmes LA, Gepshtein S, Lu Z-L, Albright TD (2009) Rapid estimation of the spatiotemporal contrast sensitivity surface. *J Vis* 9(8):696
39. Sakitt B, Barlow HB (1982) A model for the economical encoding of the visual image in cerebral cortex. *Biol Cybern* 43:97–108
40. Laughlin SB (1989) The role of sensory adaptation in the retina. *J Exp Biol* 146(1):39–62
41. Wainwright MJ (1999) Visual adaptation as optimal information transmission. *Vis Res* 39:3960–3974
42. Laughlin SB, Sejnowski TJ (2003) Communication in neuronal networks. *Science* 301(5641):1870–1874
43. Clifford CWG, Wenderoth P (1999) Adaptation to temporal modulation can enhance differential speed sensitivity. *Vis Res* 39:4324–4332
44. Kregelberg B, Van Wezel RJ, Albright TD (2006) Adaptation in macaque MT reduces perceived speed and improves speed discrimination. *J Neurophysiol* 95:255–270
45. Gepshtein S (2009) Closing the gap between ideal and real behavior: scientific versus engineering approaches to normativity. *Philos Psychol* 22:61–75
46. Yeshurun Y, Carrasco M (1998) Attention improves or impairs visual performance by enhancing spatial resolution. *Nature* 396:72–75
47. Yeshurun Y, Carrasco M (2000) The locus of attentional effects in texture segmentation. *Nat Neurosci* 3(6):622–627
48. Jurica P, Gepshtein S, Tyukin I, Prokhorov D, van Leeuwen C (2007) Unsupervised adaptive optimization of motion-sensitive systems guided by measurement uncertainty. In: *International conference on intelligent sensors, sensor networks and information, ISSNIP 2007*, vol 3, Melbourne, Qld:179–184
49. Jurica P, Gepshtein S, Tyukin I, van Leeuwen C (2013) Sensory optimization by stochastic tuning. *Psychol Rev* 120(4):798–816
50. Hebb DO (1949) *The organization of behavior*. John Wiley, New York

51. Bienenstock EL, Cooper LN, Munr PW (1982) Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J Neurosci* 2: 32–48
52. Paulsen O, Sejnowski TJ (2000) Natural patterns of activity and long-term synaptic plasticity. *Curr Opin Neurobiol* 10(2):172–180
53. Bi G, Poo M (2001) Synaptic modification by correlated activity: Hebb's postulate revisited. *Annu Rev Neurosci* 24:139–166
54. Gardiner CW (1996) *Handbook of stochastic methods: for physics, chemistry and the natural sciences*. Springer, New York
55. Vergassola M, Villermaux E, Shraiman BI (2007) 'Infotaxis' as a strategy for searching without gradients. *Nature* 445:406–409
56. Gepshtein S, Jurica P, Tyukin I, van Leeuwen C, Albright TD (2010) Optimal sensory adaptation without prior representation of the environment. In: 40th Annual Meeting of the Society for Neuroscience, vol 731.7
57. von Neumann J (1963) *Theory of games, astrophysics, hydrodynamics and meteorology*. In: Taub AH (ed) *Collected works*, vol VI. Pergamon Press, New York
58. Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech J* 27:379–423, 623–656
59. Jaynes ET (1957) Information theory and statistical mechanics. *Phys Rev* 106:620–630
60. Gorban A (2013) Maxallent: maximizers of all entropies and uncertainty of uncertainty. *Comput Math Appl* 65(10):1438–1456
61. Cover TM, Thomas JA (2006) *Elements of information theory*. John Wiley, New York

Chapter 8

Scene Analysis Using Morphological Mathematics and Fuzzy Logic

Victoria Lynn Fox, Mariofanna Milanova and Salim Al-Ali

Abstract Owing to compound textural features, intensity inhomogeneity, image layers, and variations of statistics inherent, the segmenting of complicated images into areas of similarity for scene analysis is a challenging task. In this work, a morphological active contour is developed to increase efficiency of current active contour schemes and a fuzzy clustering energy is incorporated into the active contour algorithm to increase accuracy and flexibility. Finally, to aid in the segmentation of figures for scene analysis, a visual attention is incorporated into the fuzzy clustering. The savings in computational efficiency garnered from using a morphological curve evolution rather than a partial differential equation and corresponding Euler-Lagrange equations combined with the expert knowledge garnered from a visual attention fuzzy logic scheme translates into a highly accurate and efficient segmentation method for scene analysis.

Keywords Segmentation · Morphological active contour · Fuzzy energy · Natural images · Visual attention

V.L. Fox (✉)

Department of Applied Science, University of Arkansas at Little Rock,
2801 S. University Avenue Little Rock, Little Rock, AR 72204, USA
e-mail: vlfox@ualr.edu

M. Milanova · S. Al-Ali

Department of Computer Science, University of Arkansas at Little Rock,
2801 S. University Avenue Little Rock, Little Rock, AR 72204, USA
e-mail: mgmilanova@ualr.edu

S. Al-Ali

e-mail: sgsaeed@ualr.edu

8.1 Introduction

While the statistics of large data sets of images follow certain regularities, statistics of singular images are found to be capable of a large variance in statistical analysis [1]. Therefore, the applications, which work with individual images, must seek to exploit the variability of the image while acknowledging the established statistical properties of natural images as a group. As an example, consider the spatial structure of a natural image, which is often irregular with contours produced by different boundaries, markings, and shadows. These boundaries can sometimes be determined by exploiting luminance and contrast within the given natural scene. In a survey of natural image statistics, Geisler [2] notes, in general, natural images, i.e. images with significant local covariance, tend to have large variations in local luminance and contrast with a low correlation in their average joint distribution. However, global statistics have shown that strong features tend to cluster in natural images and, thus, ignoring global information in light of only local information often results in poor segmentation results, when considering natural images [3]. Therefore, the segmentation protocols seeking to exploit luminance and contrast in a natural image must consider both local and global statistics, which leads to an increase in computational cost in segmentation algorithms. It bears mentioning that the statistics involving luminance and contrast are of only one group of statistical models for natural images. To effectively segment natural images, one must also consider other statistical information, such as textural information given by entropy and homogeneity, depth given by the scene scale of the image or levels of color saturation in a multispectral image. With each additional feature space, the complexity of a given algorithm grows. Therefore, it is very important for the mechanism of an image segmentation method to be as efficient as possible with low computational cost while minimizing error.

While the last two decades have seen a large variety of image segmentation methods with many able to produce reasonable segmentations on images with moderate complexity, see [1, 2, 4] and their references as examples of effective segmentation techniques, computational efficiency is still a concern, when segmenting a complex image. Many state-of-the-art methods can become impractically time-consuming or are limited in the types or size of images that can be processed. Often, feature vectors are sacrificed in order to increase computational speed resulting in a lower level of accuracy in general for the algorithm [3]. The proposed method presented in this work is a computational low-cost segmentation method that effectively segments a variety of complex images. This method makes use of mathematical morphology, fuzzy logic clustering, and visual attention, all of which are incorporated into a hybrid, level set active contour method.

The chapter is organized as the follows. Background material is presented in Sect. 8.2. The proposed method is developed in Sect. 8.3. Conclusion is situated in Sect. 8.4.

8.2 Background Material

Let us consider briefly the background materials including segmentation methods (Sect. 8.2.1), the basics of morphological mathematics (Sect. 8.2.2), application of fuzzy logic in imaging (Sect. 8.2.3), and visual attention issues (Sect. 8.2.4).

8.2.1 Segmentation Methods

The objective of segmentation is to partition an image into regions. With the assumption that every section in an image is sufficiently homogenous, edge-based segmentation determines the transition between two sections on the basis of discontinuities alone. When this assumption is not valid, a region-based segmentation usually provides a more realistic segmentation product.

In order to detect meaningful discontinuities between sections, most edge-detection techniques employ the use of first- and second-derivatives. The first-order derivative of choice is the gradient vector of an image $I(x, y)$ given as Eq. 8.1, which is obtained by the partial derivatives at every pixel location.

$$\nabla I = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial I / \partial x \\ \partial I / \partial y \end{bmatrix} : \Omega \rightarrow \Re^2 \quad (8.1)$$

To determine the presence of edges, the magnitude of the gradient vector is computed with Eq. 8.2.

$$|\nabla I| = \sqrt{g_x^2 + g_y^2} = \sqrt{(\partial I / \partial x)^2 + (\partial I / \partial y)^2} : \Omega \rightarrow \Re \quad (8.2)$$

Since it is zero in areas of constant intensity and its values are related to the degree of intensity change in areas of variable intensity. The Laplacian of an image function $I(x, y)$ is the sum of the second-order derivatives, defined by Eq. 8.3.

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} : \Omega \rightarrow \Re \quad (8.3)$$

While the Laplacian is seldom used by itself for edge detection due to its sensitivity to noise, it is powerful, when used in combination with other edge-detection techniques. The edge detection by gradient operations generally performs well only in images with defined intensity transitions and relatively low noise. However, computationally, the gradient operator methods have a relatively lower cost than other segmentation methods because the computation can be performed with a local filtering operation [5].

While the edge-based segmentation focuses on discontinuities in intensity levels, the region-based techniques find the regions directly. The basic formulation of region-based segmentation with R representing a region in the image can be given with a series of conditions. The first condition states that every pixel must be assigned a region while the second condition requires that points in a region be connected (e.g., 4- or 8-connected) and the third condition indicates the regions must be disjoint. The fourth condition states that pixels in a segmented region must share some predefined common features while the fifth condition indicates that adjacent regions are different in the sense of the predefined feature spaces [6].

The region growing is a technique that merges regions of interest into a larger region of interest. The pixel aggregation is an example of a region growing technique. In pixel aggregation, an initial set of seed points grows regions from the seeds by joining neighboring pixels, if they satisfy given criteria. In its most basic form, segmentation starts with two initial seeds, and then the region grows, if neighboring pixels satisfy the following criteria: $|I(x, y) - I(seed)| < \tau$, in which τ is some predefined threshold. The selection of initial seeds is often based on the nature of applications or images. If a priori information is not obtainable, then the clustering techniques must be used to determine the pixels that can be used as seeds. Despite the simple nature of the algorithm, there are several problematic areas in the implementation of region growing: descriptors of region properties alone can yield misleading results, if connectivity is not taken into consideration.

The statistical estimation is another common approach in a region-based segmentation [7]. When considering the statistical segmentation of images, authors generally suppose the existence of two random fields: the field of “classes” and the field of “measurements.” With this method, two sections are considered to be homogenous and accordingly merged, if they have common parameter values within a given threshold. In application, the parameters of a section cannot be directly observed; rather they can only be inferred from the observed data. This inference, if often made using Bayes’s rule and the conditional probability density function $p(\mathbf{I}(x, y) | \theta_m)$, which presets the conditional probability statistic derived from the data $(\mathbf{I}(x, y))$, will be observed, given that section m has the parameter values of θ_m . In typical statistical region merging algorithms [8], stochastic estimates in the parameter space are obtained for different sections, and merging decisions are based on the similarity of these parameters.

Unfortunately, there is a limitation of most estimation-based segmentation methods in that they do not explicitly represent the uncertainty in the estimated parameter values and, therefore, are prone to error, when the parameter estimates are poor. To counteract this limitation, a Bayesian probability of homogeneity uses all of the information contained in the statistical image model rather than just estimating parameter values. The probability of homogeneity is based on the ability to formulate a prior probability density function on the parameter space, and measures the uniformity by taking the expectation of the data likelihood over a posterior parameter space.

In general, the region-based methods yield more reasonable segmentations than edge-based algorithms, when an image has relatively large noise and/or requires the use of local properties combined with global properties. However, the complexity and computational cost of region-based methods can be large, particularly when considering methods based upon partial differential equations, e.g. active contour methods. Active contours are the energy-based segmentation methods that seek to guide partitioning of an image via the minimization of a cost functional.

The first efforts in formulating the boundary detection problem as an energy minimization problem resulted in the energy-minimizing splines guided by external constraint forces that pull the splines towards objects of interest [9]. These splines were dubbed snakes by their creators, Kass, Witkin, and Terzopoulos, because the contours appeared to slither across an image as it moved toward local minima. In the classical formulation, the boundary detection consisted of matching a deformable model to an image by means of energy minimization. Representing the position of a snake parametrically by $C(s) = (x(s), y(s)): 0 \leq s \leq L: \mathfrak{R} \rightarrow \Omega$, where L denotes the length of the contour C , s is the arc length, and Ω represents the entire domain of an image $I(x, y)$, the energy functional can be written as Eq. 8.4, where E_{int} and E_{ext} represent the internal energy and external energy functions.

$$E(C) = E_{int} + E_{ext} \quad (8.4)$$

The internal energy function determines the smooth shape (regularity) of the contour. A common choice for the internal energy is a functional given by Eq. 8.5, where $C(s) \approx C(n) = \{(x(n), y(n)): 0 \leq n \leq N, s = 0 + n\Delta s\}$ and $L = N\Delta s$.

$$E_{int} = \int_0^L \alpha |C'(s)|^2 + \beta |C''(s)|^2 ds \approx \sum_0^N \alpha |C'(n)|^2 + \beta |C''(n)|^2 \Delta s \quad (8.5)$$

Here α controls the tension of the contour and β controls the rigidity of the contour. The external energy term determines the criteria of contour evolution depending on the image $I(x, y)$ and can be defined as Eq. 8.6, where $E_{img}(x, y)$ denotes a scalar function defined on the image plane, so the local minimum of E_{img} attracts the snakes to edges.

$$E_{ext} = \int_0^L E_{img}(C(s)) ds \approx \sum_{n=0}^N E_{img}(C(n)) \Delta s \quad (8.6)$$

The edge attraction function, E_{img} , was originally presented as a combination of three separate functionals: $E_{img} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term}$. Since the presentation of the original algorithm, there have been many other functionals suggested for E_{img} and one common example is a function of image gradient, given

by Eq. 8.7, which incorporates a Gaussian smoothing filter, G_σ , with standard deviation σ and a suitably chosen constant λ .

$$E_{img}(x, y) = \frac{1}{\lambda |\nabla G_\sigma * I(x, y)|} : \Omega \rightarrow \Re \quad (8.7)$$

In order to solve the geometric contour problem, one must find the contour C that minimizes the total energy term E within the given set of weights, α , β , and λ .

In order to achieve an accurate location of edges, the classical snake algorithm must be initialized sufficiently near the edge or object of interest. Estimating a correct position of an initial contour without prior knowledge is a challenging problem. Also, classical geometric contours are only able to separate a region into two sub-regions and cannot, subsequently split into multiple boundaries or merge from multiple initial contours. In order to correct this deficiency, Hamilton-Jacobi formulations [5] were applied to active contours and resulted in the creation of geodesic active contours.

The level set function $\phi(x, y)$ was proposed by Osher and Sethian [5] as a formulation to implement active contours. Osher and Sethian represented a contour implicitly via a two-dimensional Lipschitz-continuous function $\phi(x, y) : \Omega \rightarrow \Re$ defined on the image plane. On a particular level, usually the zero level, the level set function is defined as a contour, such as Eq. 8.8, where Ω denotes the entire image plane.

$$C = \{(x, y) : \phi(x, y) = 0\}, \forall (x, y) \in \Omega \quad (8.8)$$

As the level set function increases from the initial stage, the corresponding set of contours, C , moves toward the exterior.

By using the zero level, the contour can be defined as the border between a positive area and negative area. Thus, the contour can be identified by checking the sign of $\phi(x, y)$. Using the zero level, the level set is usually represented by Eq. 8.9.

$$\phi(x, y) = \begin{cases} < 0(x, y) \text{ inside } C \\ = 0(x, y) \text{ on } C \\ > 0(x, y) \text{ outside } C \end{cases} \quad (8.9)$$

The initial level set function $\phi_0(x, y) : \Omega \rightarrow \Re$ is usually given as a signed distance from the initial contour such as in Eq. 8.10 in a way that $\pm D(a, b)$ denotes a signed distance between a and b and $N_{x,y}(C_0)$ denotes the nearest neighbor pixel on the initial contours $C = C(t = 0)$ from (x, y) .

$$\phi_0(x, y) = \{\phi(x, y) : t = 0\} = \pm D((x, y), N_{x,y}(C_0)), \forall (x, y) \in \Omega \quad (8.10)$$

The deformation of the contour is generally represented as a Partial Differential Equation (PDE). The initial proposal for a formulation of contour evolution using the magnitude of the gradient, given by Osher and Sethian [5], states by Eq. 8.11,

where v denotes a constant speed term to push or pull the contour and $\kappa(\cdot) : \Omega \rightarrow \Re$ denotes the mean curvature of the level set function $\phi(x, y)$ given by Eq. 8.12.

$$\frac{\partial \phi(x, y)}{\partial t} = |\nabla \phi(x, y)| (v + \epsilon \kappa(\phi(x, y))) \quad (8.11)$$

$$\kappa(\phi(x, y)) = \operatorname{div} \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}} \quad (8.12)$$

The curvature term is used to control the regularity of the contour as the internal energy term does in the classic snake model while ϵ controls the balance between the smoothness and the robustness of the evolution.

Chan and Vese [10] proposed a new form of contour evolution that is very popular in current research methods, the active contour without the edges method. The length of the contour $|\mathcal{C}|$ can be approximated by a function of $\phi(x, y)$ such as in Eq. 8.13, where $H_\epsilon(\cdot)$ denotes the regularized form of the unit step function, $H(\cdot) : \Omega \rightarrow \Re$ given by Eq. 8.14 and $\delta_\epsilon(\cdot)$ denotes the derivative of $H_\epsilon(\cdot)$.

$$|\mathcal{C}| \approx L_\epsilon(\phi(x, y)) = \int |\nabla H_\epsilon(\phi(x, y))| dx dy = \int \delta_\epsilon(\phi(x, y)) |\nabla \phi(x, y)| dx dy \quad (8.13)$$

$$H(x, y) = \begin{cases} 1 & \text{if } \phi(x, y) \geq 0 \\ 0 & \text{if } \phi(x, y) < 0 \end{cases} \quad \forall (x, y) \in \Omega \quad (8.14)$$

Since $H_\epsilon(\cdot)$ produces either a 0 or 1 depending on the sign of the input, $\delta_\epsilon(\cdot)$ produces nonzero results only on the contour of Eq. 8.13, where $\phi(x, y) = 0$. The associated Euler-Lagrange equation [11] obtained by minimizing $L_\epsilon(\cdot)$ with respect to ϕ and parameterizing the descent directions by an artificial time t is given by Eq. 8.15.

$$\frac{\partial \phi(x, y)}{\partial t} = \delta_\epsilon(\phi(x, y)) \kappa(\phi(x, y)) \quad (8.15)$$

The contour evolution motivated by this equation can be interpreted as the motion by mean curvature minimizing the length of the contour. Therefore, Eq. 8.12 is considered as the motion motivation by partial differential equation, while Eq. 8.15 is considered as the motion motivated by energy minimization.

A convenient characteristic of level-set contours is that the contour can split or merge as the topology of the level set function changes. As a result, level set methods can detect more than one boundary simultaneously and several initial contours can be placed. The computational cost of level set methods, however, is high because computation should be done on the same dimension as the image plane. Yet, due to the convenience and flexibility of level set methods, they are a practical method to use in the research of segmenting multi-region images.

8.2.2 Morphological Mathematics

First introduced by Matheron [12] and Serra [13], the mathematical morphology views an image as a set of geometric structures then transforms it with the use of a smaller geometrically defined set, commonly referred to as a structuring element. The structuring element is translated over the image set and with the use of basic set operations (i.e. union and intersection), the fundamental operations of dilation and erosion are obtained.

For a binary image, entries consisting of 0 represent background information and entries consisting of 1 represent foreground entries. In a binary erosion, $A \ominus B$, in which A is eroded by the structuring element B , consists of all points, for which the translation of B over A fits inside of A . In other words, it is the set operation $A \ominus B = \{x | B_x \subseteq A\}$. A binary dilation is the dual operation to erosion and is defined by the set complementation of erosion. The dilation of a set A by structuring element B is given by $A \oplus B = [A^c \ominus (-B)]^c$. To dilate A by B , B is rotated around the origin to create $(-B)$, the complement of A is eroded by $(-B)$, and then the complement of the erosion is taken. To illustrate with a sample binary image matrix, let the binary image be represented by A and the structuring element is represented by B as given by the matrix in Eq. 8.16. The erosion of A by B is given by the matrix in Eq. 8.17 while the dilation of A by B is given by the matrix in Eq. 8.18.

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad (8.16)$$

$$A \ominus B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (8.17)$$

$$A \oplus B = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (8.18)$$

From erosion and dilation, the morphological operations of opening and closing can be defined. A morphological opening on a binary image is defined as $A \odot B = (A \ominus B) \oplus B$, where the image A is eroded by B , and then the result of the erosion is dilated by B . Using the sets defined in Eq. 8.16, $A \odot B$ has the results represented in Fig. 8.1.

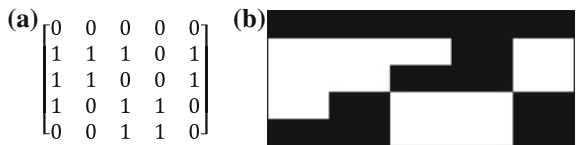


Fig. 8.1 Example of morphological opening on a binary set. **a** A matrix form of $A \ominus B$. **b** An image representation of $A \ominus B$

The morphological closing on a binary image is defined as $A \odot B = (A \oplus B) \ominus B$. Using the sets defined in Eq. 8.16, $A \odot B$ results in the matrix and image representation in Fig. 8.2.

For a binary structuring element B , the locations, where B is equal to zero, are referred to as neutral elements since they do not affect the image during the morphological operation. For gray-scale morphology, the morphological operations transform the gray-scale image into a binary data set with an extra dimension representing the gray-level. Since the gray-scale level of an image is bound to a finite domain of $[0, m]$, the neutral elements of a gray-scale structuring element are the elements with values of m [14]. This is due to the nature of the definition of gray-scale morphology operators, in which they are used as the invariants to the maximum and minimum operators.

In continuous gray-scale morphology, images are viewed as functions mapping a grid to $\mathbb{R} \cup (-\infty, \infty)$. Since gray-scale images are restricted to integer values between some range of values (i.e. 0–255 for 8 bit), it is necessary to restrict the discrete gray-scale morphology mapping to the integer range imbedded in the image format. With these preliminaries stated, the erosion and dilation of an gray-scale image, $A(x)$, by a structuring element $B(x)$, can be defined as $(A \ominus B)(x) =$

$$\inf_{y \in B} [A(y) - B(y - x)] \text{ and } (A \oplus B)(x) = \sup_{y \in B} [A(y) - B(y - x)], \text{ respectively.}$$

8.2.3 Fuzzy Logic

Fuzzy logic is composed of multi-logic systems that have been developed in opposition to the classical logic, which uses an “on/off” switch in its assessment of membership. In fuzzy logic sets, the membership is determined by values assigned

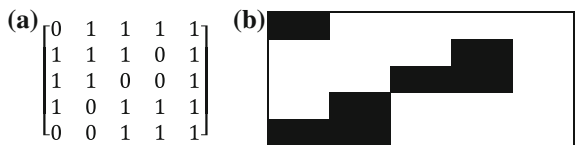


Fig. 8.2 Example of morphological closing on a binary set. **a** A matrix form of $A \odot B$. **b** An image representation of $A \odot B$

to linguistic expressions and human decisions. The use of fuzzy logic in imaging is primarily in the practice of image clarity and identification of objects [15]. It is a modification of this second usage, i.e. identification of objects, where segmentation has begun to use fuzzy logic.

A linguistic variable is a term used in our natural language to describe some concept that usually has vague or ill-defined values. For example, if researcher tried to describe the frequency heights of an image in the frequency domain, the linguistic variable would be “height” and the typical values would be “low”, “medium”, and “high” and would define our clustering. The fuzzy expert system process is composed of four steps:

1. Fuzzification—convert the data to fuzzy sets via membership functions.
2. Inference—perform all fuzzy logical operations and apply an implication method.
3. Composition—apply an aggregation method for fuzzy sets acquired in the inference step.
4. Defuzzification—convert the final fuzzy conclusion back to raw data to obtain final weights.

It is important to note that fuzzy logic is not logic that is fuzzy; rather it is the logic of fuzziness. While the linguistic variable may be filled with ambiguity, the output of the defuzzification is a value that will guide the image segmentation.

There are several fuzzy membership functions to help cluster the values of the linguistic variable. In choosing the linguistic variables and terms for the fuzzy logic model, it is important to be directed by the following guidelines [16]:

1. The features should carry enough information about the image and should not require any domain-specific knowledge for their extraction.
2. They should be easy to compute in order for the approach to be feasible for a large image collection.
3. They should relate well with the human perceptual characteristics since users will finally determine the suitability of the retrieved images.

8.2.4 Visual Attention

In 1964, Neisser [17] presented a popular model, in which human vision consists of pre-attentive and attentive stages. The pre-attentive stage focuses on local spatial discontinuity, while in the attentive stage, relationships between these discontinuities are created and clustering takes place. In the pre-attentive stage, the principles of proximity, simplest form, and continuity factor into the decision, of where the spatial discontinuities take place. The attention stage additionally takes note of similarity features (e.g. color, luminosity, texture) and shape to aid in the process.

In computational vision, the numerous approaches to the attention model of scene analysis can be categorized as two methodologies: bottom-up and top-down.

In bottom-up attention, dissimilarities attract the attention of the vision model. In the top-down methodology, the viewer searches for a specific feature and the expected objects receive the attention [18]. Through both of these methodologies, a saliency map can be created to aid in the segmentation of the visual scene. In general, bottom-up attention models are based upon feature detection (e.g. orientation, shape, texture) that is easily estimated by a computer while top-down models are more subjective and increase computational complexity in that they depend on contextual clues, objectives, and expert knowledge of the viewer. In an effort to reduce computational cost, the method presented here will make use of the bottom-up saliency model known as Context-Aware Saliency [19, 20].

Context-Aware Saliency Detection (CASD) makes use of the principles of Gestalt vision psychology in its formulation. Particularly, the model makes use of following information:

1. Local, pre-attentive features such as color, texture, and contrast.
2. Global attentiveness, which identify features that deviate from the norm.
3. Perceptual organization rules such as a visual scene containing at least one center of gravity.
4. High-level factors such as distance priors or shape priors.

Using the local-global feature fusion and perceptual organization rules, the CASD detects the salient objects along with the regions of the image around the salient object in order to lend context to the salient region. The local, pre-attentive features give distinctive areas a high saliency and homogenous regions a low saliency score. Frequently occurring features are classified as part of the ground and rare features are classified as part of the figure per global attentiveness. Perceptual organization groups salient pixels that are in close proximity to each other and discounts salient pixels that are not connected. Finally, a center prior contributes to the determination, of which of the salient pixels have the highest fixation levels. Thus, the CASD makes use of the similarity and proximity principles of Gestalt psychology with exceptionally low computational cost. In particular, the method evaluates pixels in patches in order to evaluate the context of each pixel. Considering a single patch p_i of scale s at each pixel, a single pixel i is salient, when the patch containing the pixel is unique with respect to all other patches in the image. This capability of incorporating context patches with the salient object gives the CASD model the flexibility needed for scene analysis.

8.3 Proposed Method

The proposed method is based on morphological operations (Sect. 8.3.1), hybrid morphological contour (Sect. 8.3.2), and representation of fuzzy morphological contour with visual attention (Sect. 8.3.3).

8.3.1 Morphological Mean Curvature

In the image processing application of level set active contours, the curve, $\mathcal{C}: \mathbb{R}^+ \times [0, 1] \rightarrow \mathbb{R}^2$, is represented implicitly as a level set of an embedding function. If we set $u: \mathbb{R}^+ \times \mathbb{R}^2 \rightarrow \mathbb{R}$ as an implicit representation of our contour, it will become $\mathcal{C}(t) = \{(x, y) | u(t, (x, y)) = 0\}$. During the contour evolution, infinitesimal change of the contour is controlled by differential operators. In other words, a differential operator \mathcal{D} guides the contour evolution with the partial differential equation $\mathcal{C}(t) = \mathcal{D}(\mathcal{C})$. In partial differential equation formulations of active contours, the smoothing force regularly takes the form of mean curvature motion and acts as a regularization term.

The underlying principal of mean curvature motion is the evolution of a simple closed curve, whose points move in the direction of the normal with specified velocity. Rewriting $\mathcal{D}(\mathcal{C}) = \mathcal{F} \cdot \mathcal{N}$, where \mathcal{N} is the normal to the contour and \mathcal{F} is a scalar field, one can determine the velocity of evolution at each point on the contour. In level set implementations, the evolution of $u(x, y)$ is $\frac{\partial u}{\partial t} = |\nabla u| \cdot \mathcal{F}$ and will equal Eq. 8.19, when \mathcal{F} is the divergence of the normalized gradient (i.e. Euclidean curvature of \mathcal{C}) and gives the curvature of the implicit curve at each point.

$$\frac{\partial u}{\partial t} = |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \quad (8.19)$$

The parameter $u(x, y)$ must be discretized in image processing in order to be applied to the grid of image information, which is usually expressed as pixels in two dimensional applications. The discretization of the differential operator is not always a trivial task and results in one of the losses of efficiency in many image processing contour applications. As a result, the search for a low-cost estimator of mean curvature motion is an area of active research.

One of the more significant contributions to the topic is provided by [21], in which it is proven the two-dimensional mean curvature term can be replaced by the mean of two morphological operators for a single iteration of the method. To morphologically approximate mean curvature, we let \mathcal{B} represent line segments of set length then define the morphological continuous line operators as mentioned in Eqs. 8.20–8.21.

$$(\mathcal{A}_h u)(x) = \sup_{B \in \mathcal{B}} \inf_{y \in x + hB} u(y) \quad (8.20)$$

$$(\mathcal{J}_h u)(x) = \inf_{B \in \mathcal{B}} \sup_{y \in x + hB} u(y) \quad (8.21)$$

Using these operators, let us then define the mean operator as Eq. 8.22, in which the scheme in [21] relates the mean operator with the mean curvature motion by Eq. 8.23.

$$(\mathcal{F}_h u)(\mathbf{x}) = \frac{(\mathcal{A}_h u)(\mathbf{x}) + (\mathcal{J}_h u)(\mathbf{x})}{2} \quad (8.22)$$

$$(\mathcal{F}_h u)(x) = u(x) + \frac{1}{4} h^2 |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) (x) + O(h^3) \quad (8.23)$$

Using a small h and subtracting $u(x)$ from each side of (Eq. 8.23), results in the infinitesimal generator of the \mathcal{F}_h operator will be the following:

$$\lim_{h \rightarrow 0^+} h^{-1} \left[(\mathcal{F}_{\sqrt{4h}} u)(\mathbf{x}) - u(\mathbf{x}) \right] = |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) (\mathbf{x}). \quad (8.24)$$

From Eq. 8.24 one can solve the mean curvature motion by means of the \mathcal{F}_h operator. However, since the \mathcal{F}_h operator generates new level set values after a single iteration, it ceases to be morphological. In [22] and [23], Alarez et al. modify the Catta, Dibos, and Koepfler scheme with the use of operator composition, which states that given any two operators \mathcal{P}_h^1 and \mathcal{P}_h^2 , we have, for a small h , Eq. 8.25.

$$\mathcal{P}_{h/2}^2 \circ \mathcal{P}_h^1 u \approx \frac{\mathcal{P}_h^2 u + \mathcal{P}_h^1 u}{2} \quad (8.25)$$

From this, Alarez et al. show that the non-morphological operator $\mathcal{F}_{\sqrt{4h}}$ can be approximated by the morphological operator represented in Eq. 8.26 with a base of \mathcal{B}^2 and is equivalent to Eq. 8.11.

$$\mathcal{A}_{\sqrt{h}} \circ \mathcal{J}_{\sqrt{h}} \approx \frac{\mathcal{A}_{\sqrt{h}} u + \mathcal{J}_{\sqrt{h}} u}{2} \quad (8.26)$$

8.3.2 A Hybrid Morphological Contour

Some authors lay out the format for a hybrid morphological contour [24, 25]. This section is a brief review of the algorithm for the contour extraction. While the combination of an edge based and region based active contour, commonly referred to as a hybrid contour, results in increased computational complexity in order to mitigate the shortcomings of either method alone, the hybrid morphological contour has a low complexity and circumventing a method's shortcoming does not significantly add to the computational cost.

In the hybrid method, the coupling of the strong edge term and region statistics creates a symbiotic relationship. When the edge term is low, the curve is attracted toward the region of interest. However, when the curve is far away from an edge, the region statistics take control of the curve evolution and the contour resists becoming a stationary model. Using the Active Contour without Edges presented as a basis for the active contour (Eq. 8.27) with the region statistics are used by the third and fourth terms.

$$\begin{aligned}
F(c_1, c_2, \mathcal{C}) = & \mu(\text{Length of } \mathcal{C}) + p(\text{Area in } \mathcal{C}) \\
& + \lambda_1 \int_{in\mathcal{C}} |u - c_1|^2 d\mathbf{x} + \lambda_2 \int_{out\mathcal{C}} |u - c_2|^2 d\mathbf{x}
\end{aligned} \tag{8.27}$$

Parameters λ_1 and λ_2 weight the importance of the regions inside and outside the curve, respectively, while c_1 and c_2 are the average intensity levels inside and outside the contour. In the hybrid morphological active contour, the third and fourth terms of Eq. 8.27 are incorporated directly into the algorithm.

The first term of the Active Contour without Edges is replaced with the morphological mean curvature evolution described in Eq. 8.25 while the second term becomes the edge-based portion of the hybrid method. In edge based methods, the contour flow is often represented with the formulation given in Eq. 8.28, where $g(I)|\nabla u|v$ is the balloon force, $\nabla g(I)\nabla u$ is the edge attraction force, and $g(I)|\nabla u|div\left(\frac{\nabla u}{|\nabla u|}\right)$ is mean curvature motion.

$$\frac{\partial u}{\partial t} = g(I)|\nabla u|v + \nabla g(I)\nabla u + g(I)|\nabla u|div\left(\frac{\nabla u}{|\nabla u|}\right) \tag{8.28}$$

The parameter $g(I)$ represents an edge image attractor usually obtained from an edge detector, u denotes the contour, and v is an inflation (or deflation) constant. Focusing on the balloon force, $g(I)$ could be obtained from any edge detector appropriate for the image. Traditionally, one would use an edge detector, which is low in the edges of the image such as Eq. 8.29.

$$g(I) = \frac{1}{\sqrt{1 + \alpha|\nabla G_\sigma * I|}} \tag{8.29}$$

In the hybrid morphological active contour method, morphological operations of dilation and erosion are used to approximate the balloon force. The dilation of a function is defined as $(\mathcal{D}_h u)(\mathbf{x}) = \sup_{y \in hB} u(\mathbf{x} - \mathbf{y})$ while erosion is defined by $(\mathcal{E}_h u)(\mathbf{x}) = \inf_{y \in hB} u(\mathbf{x} - \mathbf{y})$. The radius of the operator is denoted by h and B is a disk structuring element of radius one. The function $u_d: \mathbb{R}^+ \times \mathbb{R}^2 \rightarrow \mathbb{R}$, where $u_d(t, \mathbf{x}) = \mathcal{D}_t u_o(\mathbf{x})$ is the solution to Eq. 8.30 for the initial condition $u_d(0, \mathbf{x}) = u_o(\mathbf{x})$ [26].

$$\frac{\partial u_d}{\partial t} = |\nabla u_d| \tag{8.30}$$

As a result, \mathcal{D}_h is the infinitesimal generator of Eq. 8.9. Using a comparable rational, we have the function $u_d: \mathbb{R}^+ \times \mathbb{R}^2 \rightarrow \mathbb{R}$, where $u_e(t, \mathbf{x}) = \mathcal{E}_t u_o(\mathbf{x})$ is the solution to Eq. 8.31.

$$\frac{\partial u_e}{\partial t} = -|\nabla u_e| \quad (8.31)$$

Using the morphological operators \mathcal{D}_h and \mathcal{E}_h , one can now solve level set evolution PDEs. In the balloon force term, $g(I)$ manages the balloon force in individual sections of the curve. The smaller $g(I)$ becomes, the closer the curve is to the edge. With the use of a threshold, factor $g(I)$ can be discretized into the morphological formulation. The product $|\nabla u|v$ leads to the PDES in Eqs. 8.30 and 8.31. If v is positive, the PDE becomes the dilation PDE. Likewise, if v is negative, then the erosion PDE is used.

In the Active Contour without Edges, the internal and external forces are combined through addition of the terms. Our hybrid morphological active contour combines them by iteratively interchanging their discretized formulations. In every iteration, first the balloon force with the edge attraction energy will be applied, then the region force is applied, and at last the mean curvature motion over the embedded level set function u is computed. Given the contour evolution at iteration, $u^n : \mathbb{R}^2 \rightarrow \{0, 1\}$, u^{n+1} is defined using the steps of the algorithm mentioned below (Eq. 8.32).

Algorithm (1)

Step 1

$$u_{balloon}^{n+1}(\mathbf{x}_i) = \begin{cases} (\mathcal{D}_d u)(\mathbf{x}_i) & \text{if } g(I)(\mathbf{x}_i) > 0 \text{ and } v > 0 \\ (\mathcal{E}_d u)(\mathbf{x}_i) & \text{if } g(I)(\mathbf{x}_i) > 0 \text{ and } v < 0 \\ u_{balloon}^n & \text{otherwise.} \end{cases}$$

Step 2

$$u_{region}^{n+1} = \begin{cases} 1 & \text{if } |\nabla u_{balloon}^{n+1}| \left[(\lambda_1(I - c_1)^2 - \lambda_2(I - c_2)^2) \right](\mathbf{x}_i) < 0 \\ 0 & \text{if } |\nabla u_{balloon}^{n+1}| \left[(\lambda_1(I - c_1)^2 - \lambda_2(I - c_2)^2) \right](\mathbf{x}_i) > 0 \\ u_{balloon}^{n+1}(\mathbf{x}_i) & \text{otherwise} \end{cases} \quad (8.32)$$

$$c_1 = \frac{\int_{\Omega} I * H(u) dx}{\int_{\text{inside } c} dx} \quad \text{and} \quad c_2 = \frac{\int_{\Omega} I * (1 - H(u)) dx}{\int_{\text{inside } c} dx}$$

Step 3

$$u^{n+1} = \begin{cases} (\mathcal{A}_d \circ \mathcal{J}_d u_{region}^{n+1})(\mathbf{x}_i) & \text{if } g(I)(\mathbf{x}) > 0 \\ u_{region}^{n+1}(\mathbf{x}_i) & \text{otherwise} \end{cases}$$

The experimental results presented in [24] and [25] clearly show the efficiency and robustness of the hybrid morphological active contour.

8.3.3 A Fuzzy Morphological Contour with Visual Attention

A fuzzy morphological approach for contour with visual attention supposes the analysis of fuzzy energy, building of visual attention model, and creation of combined algorithm.

In [27], Krindis and Chatzis introduced a new type of energy to drive active contours during the segmentation process. This energy, referred to as fuzzy energy, was derived from using a fuzzy logic clustering method and then employs the membership values and weights into the active contour formulation. Specifically, the following functionals are incorporated into a regional active contour model provided by Eq. 8.33, where Ω is the image domain and C is an evolving curve such that $C \subset \Omega$.

$$F_1(C) + F_2(C) = \int_{\Omega} [u(x, y)]^m |I(x, y) - c_1|^2 dx dy + \int_{\Omega} [1 - u(x, y)]^m |I(x, y) - c_2|^2 dx dy \quad (8.33)$$

An image $I(x, y)$ is clustered into two regions by a fuzzy clustering algorithm, where $u(x, y)$ represents the membership values of a pixel for each region and m is a weighting exponent on each fuzzy membership. The model is formulated in a pseudo-level set due to the fact that membership values of $u(x, y) = [0, 1]$. The pseudo-level set is a set of Lipschitz similar function $u: I \rightarrow \mathbb{R}$ presented in Eq. 8.34 and maintains the ideology of using membership values to define $u(x, y)$.

$$\begin{cases} C = \{(x, y) \in I: u(x, y) = 0.5\} \\ C_{inside} = \{(x, y) \in I: u(x, y) > 0.5\} \\ C_{outside} = \{(x, y) \in I: u(x, y) < 0.5\} \end{cases} \quad (8.34)$$

Unfortunately, while the Fuzzy Energy based Active Contour proves to be computationally efficient, it suffers in its inability to robustly segment textural or multispectral images. However, the flexibility provided by fuzzy energy can be translated into the hybrid morphological active contour by changing the region-based step of the algorithm and morphing the level set into the pseudo-level set of Eq 8.34. The region step will take the form of Eq. 8.35, where c_1 represents the average inside the contour and c_2 is the average outside the contour.

$$u_{region}^{n+1} = \frac{1}{1 + \left(\frac{\lambda_1(I-c_1)^2(x)}{\lambda_2(I-c_2)^2(x)} \right)^{\frac{1}{m-1}}} \quad (8.35)$$

$$c_1 = \frac{\int_{\Omega} I * (u)^m dx}{\int_{\Omega} (u)^m dx} \quad c_2 = \frac{\int_{\Omega} I * (1-u)^m dx}{\int_{\Omega} (1-u)^m dx}$$



Fig. 8.3 Example of boundary extraction. **a** An original image courtesy of Caltech [28]. **b** A saliency map generated by CASD. **c** Edges from saliency map

The parameter m represents the fuzzy weighting exponent defined in the fuzzy rules for the clustering.

The visual attention model is incorporated into the hybrid morphological active contour in two ways. First, the saliency of an image is calculated and then transformed into an edge image to help define the boundary of the salient object (see Fig. 8.3) giving the algorithm the image created by $g(I)$ from Eq. 8.28. Second, the visual attention result is compared to the fuzzy clustering results, and the fuzzy clustering result, which most closely matches the saliency image, is used as the basis for the fuzzy energy (see Fig. 8.4).

The algorithm incorporating fuzzy energy and visual attention into the hybrid morphological active contour is given below, where the membership values from the fuzzy clustering most similar to the saliency image give u_0 .

Algorithm (2)

$$pixel\ is \begin{cases} \text{on } C & \text{if } I : u_0(x, y) = 0.5 \\ \text{inside } C & \text{if } I : u_0(x, y) > 0.5 \\ \text{outside } C & \text{if } I : u_0(x, y) < 0.5 \end{cases}$$

The Algorithm 2 includes the following steps.

Step 1

$$u_{balloon}^{n+1}(x) = \begin{cases} (D_d u)(x) & \text{if } g(I)(x) > 0 \text{ and } v > 0 \\ (\mathcal{E}_d u)(x) & \text{if } g(I)(x) > 0 \text{ and } v < 0 \\ u_{balloon}^n & \text{otherwise.} \end{cases}$$



Fig. 8.4 Examples of fuzzy clustering result with three classes. **a** Class 1 membership map. **b** Class 2 membership map. **c** Class 3 membership map

The parameter $g(I)$ is an edge attractor image calculated using Eq. 8.28 on the saliency image.

Step 2

$$u_{region}^{n+1} = \frac{1}{1 + \left(\frac{\lambda_1(I-c_1)^2(\mathbf{x})}{\lambda_2(I-c_2)^2(\mathbf{x})} \right)^{\frac{1}{m-1}}}$$



Fig. 8.5 Segmentation results of fuzzy morphological active contour with visual attention. **a** An original image. **b** Segmentation result. **c** An original image. **d** Segmentation result

where $c_1 = \frac{\int_{\Omega} I*(u_{balloon})^m dx}{\int_{\Omega} (u_{balloon})^m dx}$, $c_2 = \frac{\int_{\Omega} I*(1-u_{balloon})^m dx}{\int_{\Omega} (1-u_{balloon})^m dx}$, and m is the fuzzy energy membership weighting exponent.

Step 3

$$u^{n+1} = \begin{cases} (\mathcal{A}_d \circ \mathcal{J}_d u_{region}^{n+1})(x_i) & \text{if } g(I)(x) > 0 \\ u_{region}^{n+1}(x_i) & \text{otherwise} \end{cases}$$

The images in the experiment are taken from the computational vision dataset at Caltech [28], GrabCut [29], and the Berkley Segmentation Dataset and Benchmark [30]. The images are chosen at random and are grayscale or RGB. The images have illumination artifacts, shadows, texture, the multiple objects to segment, intensity inhomogeneity, and noise, i.e. typical image artifacts that make segmentation of visual scenes a nontrivial task. Figure 8.5 demonstrates a random sampling of the results.

The comparison of the fuzzy c-means clustering to the saliency image is conducted automatically and uses similarity cues to make the selection, of which result to use. The membership function values of each pixel in the chosen clustering result are then used to create the level set function. This results in the use of fuzzy energy in the active contour and prompts a change in the calculations of the level set during the region competition portion of the method. As demonstrated with the sample image results, the algorithm successfully segments salient figures in interior and exterior environments. It also successfully segmented foreground figures in areas of high texture. These results lend themselves to the effective segmentation of static images for scene analysis.

8.4 Conclusion

The images segmented in this work represent just the tip of the possible image types this method has the potential to segment. It would be trivial to extend the work to a true multispectral algorithm as well as incorporating more texture cues into the fuzzy membership rules. The extension of the algorithm to video sequences would also be a simple matter of changing the image parameters to video and incorporating a comparison module for each frame of videos. The use of a saliency model to compare clustering results leading into the membership values for the contour evolution helps ensure the method will segment a salient object as defined by human vision research. As a final point, the savings in computational efficiency garnered from using a morphological curve evolution rather than a partial differential equation and corresponding Euler-Lagrange equations translates into a highly accurate and efficient segmentation method.

References

1. Wei W, Xin Y (2007) A modified multiphase level set evolution scheme for ariel image segmentation. *Int J Pattern Recogn Artif Intel* 21(7):1195–1212
2. Ilea DE, Whelan PF (2011) Image segmentation based on the integration of colour-texture descriptors—a review. *Pattern Recogn* 44(10):2479–2501
3. Mittal A, Sofat S, Hancock E (2012) Detection of edges in color images: a review and evaluative comparison of state-of-the-art techniques. In: Kamel M, Karray F, Hagrais H (eds) *Autonomous and intelligent systems*, LNCS. Springer, Berlin
4. Chan T, Vese L, Sandberg Y (2000) Active contours without edges for vector-valued images. *J Vis Commun Image Represent* 11(2):130–141
5. LaValle M, Hutchinson S (1995) A Bayesian segmentation methodology for parametric image models. *IEEE Trans Pattern Anal Mach Intel* 17(2):211–217
6. Gonzalez R, Woods R, Eddins S (2009) *Digital image processing using matlab*, 2nd edn. Gatesmark Publishing, Knoxville
7. Pieczynski W (1992) Statistical image segmentation. *Mach Graph Vis* 1(1–2):261–268
8. Silverman J, Cooper D (1988) Bayesian clustering for unsupervised estimation of surface and texture models. *IEEE Trans Patt Anal Mach Intel* 10(4):482–496
9. Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. *Int J Comput Vis* 1(4):321–331
10. Chan T, Vese L (2001) Active contours with edges. *IEEE Trans Image Process* 10(2):266–277
11. Sethian J (1999) *Level set methods and fast marching methods*, 2nd edn. Cambridge monographs on applied and computational mathematics. Cambridge University Press, Cambridge
12. Matheron G (1975) *Random sets and integral geometry*. Wiley, New York
13. Serra J (1982) *Image analysis and mathematical morphology*, vol 1. Academic Press, London
14. Gurillo P, Frago N, Fuentes R (2003) Fuzzy morphological operators. *Image Process Mathware Soft Comput* 10(3):8–100
15. Mario I, Chacon M (2006) Fuzzy logic for image processing: definition and applications of a fuzzy image processing scheme. In: Bai Y, Zhuang H, Wang D (eds) *Advanced fuzzy logic technology in industrial applications*
16. Mondal K, Dutta P, Bhattacharyya S (2012) Fuzzy logic based gray image extraction and segmentation. *Int J Sci EngResearch* 3(4):1–14
17. Neisser U (1964) Visual search. *Sci Am* 210(6):94–102
18. Zhang Q, Gu G, Xiao H (2009) Image segmentation based on visual attention mechanism. *J Multimedia* 4(6):363–370
19. Goferman S, Zelnik-Manor L, Tal A (2010) Context-aware saliency detection. *IEEE Int Conf Comput Vis Patt Recogn (CVPR'2010)*:2376–2383
20. Goferman S, Zelnik-Manor L, Tal A (2012) Context-aware saliency detection. *IEEE Trans Patt Anal Mach Intel* 34(10):1915–1926
21. Catte F, Dibos F, Koepfler G (1995) A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets. *SIAM J Numer Anal* 32(6):1895–1909
22. Alvarez L, Baumela L, Henriquez P, Marquez-Neila P (2010) Morphological snakes. *IEEE Int Conf Comput Vis Pat Recogn (CVPR'2010)*:2197–2202
23. Marquez-Neila P, Baumela L, Alvarez L (2014) A morphological approach to curvature-based evolution of curves and surfaces. *IEEE Trans Patt Anal Mach Intel* 36(1):2–17
24. Fox V, Milanova M, Al-Ali S (2013) A morphological multiphase active contour for vascular segmentation. *Int J Bioinf Biosci* 3(3):1–12
25. Fox V, Milanova M, Al-Ali S (2013) A hybrid morphological active contour for natural images. *Int J Comput Sci Eng Appl* 3(4):1–13
26. Kimmel R (2004) *Numerical geometry of images: theory, algorithms, and applications*. Springer Science + Business Media, New York

27. Krinidis S, Chatzis V (2009) Fuzzy energy-based active contours. *IEEE Trans Image Process* 18(2):2747–2755
28. Fei-Fei L, Fergus R, Perona P (2007) Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. *J Comput Vis Image Underst* 106(1):59–70
29. Rother C, Kolmogorov V, Blake A (2004) GrabCut: interactive foreground extraction using iterated graph cuts. In: *SIGGRAPH'04*, vol 23(3), pp 309–314
30. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *8th International Conference on Computer Vision (ICCV'2001)*, vol 2, pp 416–423

Chapter 9

Digital Video Stabilization in Static and Dynamic Scenes

Margarita N. Favorskaya, Lakhmi C. Jain
and Vladimir Buryachenko

Abstract The digital video stabilization is oriented on the removal of unintentional motions from video sequences caused by camera vibrations under external conditions, motion of robots stabilized platforms in a rugged landscape, a sea, oceans, or jitters during a non-professional hand-held shooting. The approaches for digital video stabilization in static and dynamic scenes are similar. However, objectively the analysis of dynamic scenes is needed in advanced intelligent methods. Several sequential stages include the choice of the key frames, the local and global motion estimations, the jitters compensation algorithm, the inpainting of frames boundaries, and the blurred frames restoration, for which the novel methods and algorithms were developed. The proposed application of fuzzy logic operators improves the separation results between the unwanted motion and the real motion of rigid objects. The corrective algorithm compensates the unwanted motion in frames; thereby the scene is aligned. The quality of stabilization in test video sequences was estimated by Peak Signal to Noise Ratio (PSNR) and Interframe Transformation Fidelity (ITF) metrics. During experiments, the PSNR and ITF estimations were received for six video sequences received from the static camera and eight video sequences received from the moving camera. The ITF estimations increase up on 3–4 dB or 15–20 % relative to the original video sequences.

Keywords Video stabilization · Video sequence · Motion estimation · Robust detectors · Fuzzy logic · Motion inpainting · Frame deblurring

M.N. Favorskaya (✉) · V. Buryachenko
Institute of Informatics and Telecommunications, Siberian State Aerospace University,
31 Krasnoyarsky Rabochy, Krasnoyarsk 660014, Russian Federation
e-mail: favorskaya@sibsau.ru

V. Buryachenko
e-mail: buryachenko@sibsau.ru

L.C. Jain
Faculty of Education, Science, Technology and Mathematics, University of Canberra,
Canberra, ACT 2601, Australia
e-mail: lakhmi.jain@unisa.edu.au

9.1 Introduction

The unintentional video camera motions are usual artifacts in non-professional hand-held shooting, surveillance tasks, or shooting by cameras, which are maintained on the mobile moving platforms in outdoor environment. As a result, the processing of non-stabilized original video sequence by the well-known conventional filters will not provide good segmentation, recognition, and surveillance of moving objects in static and dynamic scenes [1]. Also such technique is wide applied in video encoding tasks [2]. In this chapter, the novel methods of Digital Video Stabilization (DVS) for video surveillance task are developed for the static and dynamic scenes. The DVS algorithms as pseudo real-time and unreal-time applications are represented, depending from criteria of accuracy/computer speed. The novelty consists in the application of Takagi-Sugeno-Kang (TSK) model for improvement the motion vectors clustering, the decision procedure of key frames choice, and the reconstruction procedure of frame boundaries in static scenes and texture tiles from neighbor frames in dynamic scenes. Some often cases of objects surveillance are studied.

All variety of methods for videos stabilization techniques can be classified as mechanical, optical, electronic, and digital approaches. Historically, the mechanical stabilization based on a feedback from vibration sensors (gyros, accelerometers, etc.) were the first applied in video cameras [3]. Various control techniques are used for the stabilized platforms. Conventional design methods, modern synthesis tools such as linear quadratic regulator or linear quadratic Gaussian with loop transfer recovery, and fuzzy control systems can be used for these purposes [4]. The scope of such devices is wide but sometimes it is required the additional video stabilization, when a magnitude of vibrations has large values.

The optical image stabilization manipulates the images before their getting to the Charge-Coupled Device (CCD). The optical devices use prisms or lens of a moving assembly for tuning of light length way through camera lens systems. Vibrations occur the shifting of the lens group on a plane perpendicular to the optical axis in both horizontal and vertical directions. Two vibration-detecting sensors are used to detect the angle and speed of movement [3]. Usually for these purposes, the additional knowledge about physical motion of camera is required. Also the optical stabilization is not suitable for small sizes mobile cameras. Thus, the DVS became the most appropriate decision in modern compact video devices [5].

The electronic stabilization systems detect the camera jitters through their sensors, when the light hits the CCD. This responds by a slightly moving, and the image remains in the same position on the CCD. Such effect decreases a video quality because the pseudo-stabilized CCD area becomes smaller. Therefore, a digital zooming or oversized CCD is required. Electronic stabilization has the advantage against the optical stabilization by reducing of lens complexity and price.

The DVS approach is achieved by the synthesis of new imagery based on removal of unintentional motions between key frames and the reconstruction of frame boundaries after frame stabilization. The complexity of this task connects with a

separation the motion of objects from the unwanted camera jitters. In the case of static scenes, the proposed stabilization method does not consider specialties of camera motion trajectory. Such interpretation permits to edit static scenes faster and exactly and allows the pseudo real-time applications. However, such approach failures under the fast or changeable motions, which are often occurred in the dynamic scenes. In this case, it is needed to choice the criterion – accuracy or computer speed. According to the chosen criterion, some novel procedures are developed. In the pseudo real-time applications, the last 25–30 frames are analyzed. In the unreal-time applications, a whole video sequence is acquired, and then new frames are generated to compensate the warping between two successive original frames [6]. The most of stabilization techniques have a deal with the rigid objects. The non-rigid objects possess such specialties, which do not permit to generalize their processing.

The DVS algorithms ought to be robust to cluttered scene background, moving objects, and lighting change. One of such fast algorithms uses the feature-histogram building [7]. Various 2D and 3D stabilization algorithms are presented in [8–10]. A mosaic-based registration technique was described by Hansen in the pioneer research [11]. This system was based on a multi-resolution iterative procedure that estimated the affine motion parameters between levels of image Laplacian pyramid. The optical flow of local patches was computed by using a cross-correlation scheme.

The chapter is organized as follows. In Sect. 9.2, the problem statement of DVS for static and dynamic scenes is discussed. The description of the existing approaches for DVS is provided by Sect. 9.3. The main novel methods and algorithms suitable for static and dynamic scenes stabilization are detailed in Sects. 9.4 and 9.5, respectively. Section 9.6 presents a discussion of experimental results, involving experiments with stationary and moving video cameras. Conclusion and future development remarks are given in Sect. 9.7.

9.2 Problem Statement

The processing of video sequence occurs in the spatio-temporal domain. The DVS task is not the exclusion. Let an original video sequence $VS_{or}(FR)|_z$, where FR is a frame, z is a common number of frames, be a non-stabilized video sequence. Its transformation to the stabilized video sequence $VS_{st}(FR)|_z$ includes the sequential sub-transformations. For static scenes, they are represented by Eq. 9.1, where the operator O_{sf} selects a current set of frames $FS_{FR_t \dots FR_{t+n}}$ for current processing, $t \in \{0, 1, 2, \dots, z - 1\}$ is a number of frame, n is a number of selected frames, the operator O_{me} estimates an unwanted motion in a scene, the operator O_{mc} compensates an unwanted motion, the operator O_{mi} scales an area of stabilized frames.

$$VS_{or}(FR)|_z \xrightarrow{O_{sf}} FS_{FR_t \dots FR_{t+n}} \xrightarrow{O_{me}} FS_{FR_t \dots FR_{t+n}} \xrightarrow{O_{mc}} FS_{FR_t \dots FR_{t+n}} \xrightarrow{O_{mi}} VS_{st}(FR)|_z \quad (9.1)$$

As a result, the stabilized video sequence $VS_{st}(FR)|_z$ with the same number of frames will be created. Such fractional processing of frames is actual procedure for pseudo real-time applications. For non-real time applications in static scene, a finite set of frames $CS_{FR_0 \dots FR_{z-1}}$ may be used.

For dynamic scenes, Eq. 9.1 is extended by the additional operator O_{ss} , which divides an original video sequence in the relatively static scenes with non-essential changing for current video processing. Equation 9.2 provides the transformations for dynamic scenes, where SS_i is a set of scenes, $i \in \{0, 1, \dots, m-1\}$ is a number of scene, m is a total amount of scenes.

$$\begin{aligned} VS_{or}(FR)|_z &\xrightarrow{O_{ss}} SS_{FR_t \dots FR_{t+n}} \xrightarrow{O_{sf}} \\ FS_{FR_t \dots FR_{t+n}} &\xrightarrow{O_{me}} FS_{FR_t \dots FR_{t+n}} \xrightarrow{O_{mc}} FS_{FR_t \dots FR_{t+n}} \xrightarrow{O_{mi}} VS_{st}(FR)|_z \end{aligned} \quad (9.2)$$

Sometimes the blurred frames or frames with strong jutties appear in a video sequence. Their interpolation is executed by one of known methods, if the first and the last frames of the current scene have the appropriate quality for the DVS; otherwise the frames with high jutties are replaced by the interpolated frames. In this case Eq. 9.2 is replaced by Eq. 9.3, where O_{fi} is the operator of frames interpolation. Each of Eqs. 9.1–9.3 corresponds to the special task of computer vision.

$$\begin{aligned} VS_{or}(FR)|_z &\xrightarrow{O_{ss}} SS_{FR_t \dots FR_{t+n}} \xrightarrow{O_{sf}} \\ FS_{FR_t \dots FR_{t+n}} &\xrightarrow{O_{me}} FS_{FR_t \dots FR_{t+n}} \xrightarrow{O_{fi}} VS_{st}(FR)|_z \end{aligned} \quad (9.3)$$

9.3 Related Work

The DVS methods smooth and compensate the undesired motion of video camera and then restore frames by algorithms of digital video processing. Usually the DVS task is divided in three sub-tasks: a motion estimation, a motion compensation, and a motion inpainting. The motion estimation is a crucial aspect of video stabilization. A great variability of motion estimation methods, which were actively developed during last years, can be classified in two main categories [12]: the comparative and the gradient methods shown in Table 9.1. The dynamic textures are the special class of objects, usually background, which are characterized by an alternate motion (a motion of growth or water under a wind). In the DVS task, it is not required to estimate such motion, only to detect and ignore. The majority objects of interest are the rigid objects, and the main motion estimation methods are developed for their segmentation [13]. The non-rigid objects have a gaseous or liquid structure and do not require the exact estimations of their motion in a scene for the DVS [14]. Let us

Table 9.1 Classification of motion estimation methods

Groups of objects	Comparative methods	Gradient methods
Dynamic textures		Method of spatio-temporal fractal analysis Analysis based on autoregression functions
Rigid objects	Background subtraction* Block-matching algorithm* Density motion functions* Motion patterns	Edge points tracking Feature points tracking Optical flow Kurtosis estimations
Non-rigid objects	Background subtraction* Block-matching algorithm*	Feature points tracking Optical flow

High speed and less accurate methods are labeled by symbol ‘*’

notice that all motion estimation methods are calculated in the spatio-temporal domain of the original video sequences.

The efficient algorithm of statistical DVS was presented by Shakoor and Moattari [15]. The algorithm reduces the computational cost of Block-Matching Algorithm (BMA) by using a mean and a variance of pixels in each analyzed block with predetermined sizes (usually 16×16 pixels). According to such approach, the best block should not have a uniform area and should belong to a background without moving objects.

The basic BMA is suitable for motion analysis of rigid objects under the assumption that shift, rotation, and scale changing between frames are non-significant and can be neglected. First, a previous frame is divided on non-crossed blocks with similar sizes, which are defined by an intensity function $I_t(x, y)$, where (x, y) are coordinates, t is a discrete time instant. Second, for each block in the small neighborhood $-S_x < d_x < +S_x$ and $-S_y < d_y < +S_y$, the most similar block in a current frame $I_{t+1}(x + d_x, y + d_y)$ is searched, which is also divided on the non-crossed blocks with the same sizes as a previous frame. The similarity is determined by a minimization of the error functional $e_{(\cdot)}$ according to the applied metric. Usually three metrics are used such as Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD), and Mean of Squared Differences (MSD) (Eq. 9.4), where n is a number of analyzed surrounding blocks.

$$\begin{aligned}
 e_{SAD}(d_x, d_y) &= \sum_{x=1}^N \sum_{y=1}^N |I_{t+1}(x, y) - I_t(x + d_x, y + d_y)| \\
 e_{SSD}(d_x, d_y) &= \sum_{x=1}^N \sum_{y=1}^N (I_{t+1}(x, y) - I_t(x + d_x, y + d_y))^2 \\
 e_{MSD}(d_x, d_y) &= \frac{1}{n \times n} \sum_{x=1}^N \sum_{y=1}^N (I_{t+1}(x, y) - I_t(x + d_x, y + d_y))^2
 \end{aligned} \tag{9.4}$$

To reduce the unnecessary computation, the partial distortion elimination was introduced in SAD-metric (pSAD) by Shakoor and Moattari [15]. Therefore, $e_{SAD}(d_x, d_y)$ is transformed in Eq. 9.5, where $k = 1, 2, \dots, N$, k is a k th partial SAD.

$$e_{pSAD}(d_x, d_y) = \sum_{x=1}^k \sum_{y=1}^N |I_{t+1}(x, y) - I_t(x + d_x, y + d_y)| \quad (9.5)$$

If an intermediate sum in k rows is larger than the minimum value of matching distortion, then the following computation is unnecessary. When the best block from four current estimated blocks is selected, a Local Motion Vector (LMV) is calculated only for the best block by use a full search strategy. This algorithm gives the main attention to search a stabilized macro-block in a frame, for which the LMVs and a Global Motion Vector (GMV) are calculated.

Methods from Table 9.1 include the steps for the LMVs and the GMVs definition. The GMVs are the basic for scene correction. For the GMVs estimations, a Speeded-Up Robust Features (SURF) tracking and a discrete Kalman filter were proposed in the researches [16, 17]. The SURF algorithm is used to obtain the stable feature points in the neighbor frames for global motion estimation by six parameters in 2D affine camera model [18]. The matching of SURF descriptors was done by the nearest neighbor distance ratio method. The Kalman filter estimated the process state at some time and then obtained the feedback measurements. The Kalman filter smoothed the estimated accumulated affine transformations by removing the high frequency components. During the motion compensation stage, the difference between the smoothed parameters and the estimated ones permitted to reconstruct the stabilized video sequence.

To account the temporal lighting variations, the generalized optical flow constraint under the non-uniform lighting change was used in the research [19]. Instead of using the traditional optical flow, the generalized optical flow constraint was applies in a local window of frame. The authors represented their performance of optical flow by Eq. 9.6, where $I_{t-1}(x, y)$ and $I_t(x, y)$ are intensity functions in a previous frame ($t - 1$) and a current frame t , respectively, u and v are velocity vector components along axes OX and OY, respectively, w is a constant for compensating the non-uniform lighting change.

$$\frac{\partial I_{t-1}(x, y)}{\partial x} u + \frac{\partial I_{t-1}(x, y)}{\partial y} v + I_{t-1}(x, y) \cdot w + I_{t-1}(x, y) - I_t(x, y) = 0 \quad (9.6)$$

Three parameters (u, v, w) are estimated by the iterative linear least squares method in a local window. The main idea of such iterative process is to move the block with the newly estimated motion vector and to compute the updated flow constraints in a recursive manner. Let us notice that the Lucas and Kanade optical flow computation method cannot provide the reliable motion estimation in the close homogeneous regions. Chang et al. [19] suggested the procedure for homogeneous regions detection by Eq. 9.7, where T is a user-defined threshold.

$$\sum_{x,y \in W_{ij}} \left(\left| \frac{\partial I_{t-1}(x,y)}{\partial x} \right| + \left| \frac{\partial I_{t-1}(x,y)}{\partial y} \right| \right) < T \quad (9.7)$$

Such homogeneous regions are skipped. Thereby, the approach can be called the sparse optical flow vectors estimation. The camera motion parameters are smoothed temporally to reduce the motion fluctuations by using a regularization method, which is considered the cost function for the penalty of data deviations and the cost function corresponding to the temporal motion smoothness constraint.

One of interesting approaches for fast and accurate global motion estimation is based on the Levenburg–Marquardt Algorithm (LMA) and several sub-sampling patterns with their combinations [20]. The LMA is a method based on the gradient descent for iteratively estimation of parameters of perspective model. The LMA is a highly expensive computationally method but it was substantially accelerated by using the sub-set selection methods based on the pixel-sub-sampling patterns. Partly the pixel sub-sampling patterns are similar to the BMA applied to the gradient frames.

One of the most popular approaches for motion compensation is based on the assumption that the GMVs have a high-frequency component, and the application of the low-pass filtering will free the original video sequence from the unwanted motion. In this case, the application of a first-order infinite impulse response of a low-pass filter integrates a differential motion in a scene and smoothes the global movement trajectory. Also a smoothing algorithm based on the smoothing absolute frame positions provides a successful stabilization performance [21, 22]. To other decisions, it may be concerned the application of discrete Fourier transform, Kalman filter, fuzzy systems, and fuzzy Kalman systems [23, 24]. Some adaptive filters with a smoothing factor and the adaptive procedures were proposed to remove the camera jitters [25, 26].

In the research of Puglisi and Battiato [27], the fast and the accurate block-based local motion estimator based only a translational motion together with a robust alignment algorithm using a voting are proposed. The collected information from the different spatial locations in a frame is applied to compute the GMVs through a voting strategy. The GMVs are related to a similarity motion model: two translations, one zoom factor, and one rotation. An integral projection-based error function is used in a search strategy. Instead of usual intensity function, the authors applied the gradient of integral projections that provides a high accuracy.

The fuzzy Kalman compensation of the GMVs in the log-polar plane was proposed by Kyriakoulis and Gasteratos [28]. Due to special features of the log-polar plane, each GMV was calculated as the average value of the four LMVs. Then the GMV displacements were imported in the fuzzy Kalman system. The fuzzy system was tested with several types of the Membership Functions (MFs), the different aggregation and the defuzzification methods.

For hand-held cameras and third-generation mobile phones, the unwanted motion is mainly caused by two independent motions: the camera motion (ego-motion) and the undesired hand jitter (high-frequency motion) [29]. The independent component

analysis searches the components that are both statistically independent and non-Gaussian. The authors assume that the estimated LMVs between the consecutive frames are the time-varying signals $x_j(t)$ and are a linear mixture of the ICs $s_i(t)$. The relative amplitude of each independent motion vector s_i at the estimated LMV is related to the selected frame region and can be defined as a weighting factor a_{ij} for each type of motion. The mixture $x_j, j = 2$ for two frame regions is represented by Eq. 9.8.

$$\begin{aligned}x_1 &= a_{11}s_1 + a_{12}s_2 \\x_2 &= a_{21}s_1 + a_{22}s_2\end{aligned}\tag{9.8}$$

Some original approaches can be found in the researches, dedicating to video stabilization by using a principal component analysis [30], an independent component analysis [31], a probabilistic global motion estimation based on Laplacian two-bit plane matching [32], wavelet transformations [33], a calculation of statistical functions, mean and variance of pixels in each block of the BMA [15], etc. An algorithm to estimate the global camera motion with Shift-Invariant Feature Transform (SIFT) features was proposed by Hu et al. [34]. These SIFT features have been proved to be affine invariant and used to remove the intentional camera motions.

Tanakian et al. [35] proposed the integrated system of the video stabilizer and the video encoder by using the BMA for the LMVs detection, the histogram analysis for the GMVs detection, and the low pass filtering for a Smooth Motion Vector (SMV) obtaining as the intentional motion correction. The authors suggested a low pass filtering to remove a high frequency component of intentional motion. They approximate the SMVs by the first-order auto-regression function (Eq. 9.9), where α is a smoothing factor, $0 \leq \alpha \leq 1$; n is a frame number.

$$|SMV_n| = \alpha|SMV_{n-1}| + (1 - \alpha)|GMV_n|\tag{9.9}$$

The authors proposed a rule to chose α value ($\alpha = 0.1$ or $\alpha = 0.95$) in dependence of the GMVs and the SMVs magnitudes in the previous frames. In their following research, a fuzzy system for tuning of smoothing factor α was suggested according to noise and camera motion acceleration [36]. The trapezoidal and triangular the MFs were used for adaptive filtering of horizontal and vertical motion components between $(n - 3)$, $(n - 2)$, $(n - 1)$, and n frames.

To product the full-frame stabilized video sequence with a good visibility is the final stage of the DVS. The direct pixel based on the full-frame video stabilization approach was proposed by Matsushita et al. [37]. This approach uses a technique of image mosaics by accumulating neighboring frames with natural stitching of multiple images and a motion deblurring method to reduce the motion blur caused by the original camera motion. The propagated motion field, based on a pyramidal version of the Lucas-Kanade optical flow computation, is used to help naturally fill up the missing image areas even for scene regions that are non-planar and dynamic. Also in order to sharpen the blurry frames by a novel interpolation-based deblurring

method was developed. The crucial idea is to transfer the blurry pixels in a current frame by the corresponding sharper pixels from neighbor frames. This method works well in most videos except the cases with a large area cover of moving object and impossibility of correct the GMVs detection.

The method of dual-tree complex wavelet transform for video stabilization was developed by Pang et al. [38]. This method considers the dependence between the phase changes of wavelet transform and shift invariant feature displacement in a spatial domain. The smoothness of motion jitters is achieved by the optimal Gaussian kernel filtering. This phase-based method is invariant to lighting changes, but has a high computational cost. Usually the technologies of image warping are used, which reduce an area of stabilized frame. The motion inpainting of frame boundaries (the reconstruction of frame boundaries) can be successfully applied for static scenes. However, this may be impossible in dynamic scenes, when the reconstruction data are absent in neighbor frames.

Liu et al. [8, 39] proposed a content-preserving warping based on two objectives: to displace all tracked feature coordinates to their regularized re-projected locations and, at the same time, to minimize the warping distortion in the content-rich regions with a minimum computational cost.

A novel method to stabilize video sequences based on a 3D perspective camera model without recovering the dense depth maps was proposed in the research of Zhang et al. [40]. By balancing the smoothness and similarity, a video stability was optimized related to rotation, zooming, and translation components with suitable weights. Based on a 3D perspective camera model, the depth relative motion (camera translation) and the depth irrelative motion (camera rotation and zooming) were separated. The corresponding SIFT features are constrained frame by frame according to the epipolar geometry theory with application of RANDOM SAMPLE Consensus (RANSAC) algorithm.

The literature survey shows a great variability of existing stabilization methods. Often the authors solve the DVS task for a static scene and prefer to use more simple decisions in order to provide the pseudo real-time applications.

9.4 Video Stabilization in Complex Static Scenes

The DVS in static scenes is a particular case of the DVS in dynamic scenes. The both approaches have some common procedures and a similar logic of realization but the differences are also essential, especially on the stage of motion inpainting. For static scenes, it is required to find vectors of unwanted motion, which are enough uniform distributed in each original frame and have the similar magnitudes and directions.

The motion estimation of cluttered background is the task with a high computational cost. The main goal of the current research was not only to develop the novel algorithms for separation of motion vectors but also to design the fast algorithms in order to make the DVS in a static scene as a pseudo real-time

application. These issues are performed in Sect. 9.4.1. Motion compensation by the smoothing of global motion in a static scene and camera path estimation is situated in Sect. 9.4.2. Section 9.4.3 provides the static scene alignment with using two main techniques: the stabilized frame scaling with the reduced stabilized frame area and the frame borders restoration with the non-changeable frame sizes.

9.4.1 Motion Estimation in Static Scene

According to the problem statement representing in Sect. 9.2, the motion estimation in static scenes can be provided by fast comparative methods. The small displacements of objects in a scene between two sequential frames may be roughly performed as the parallel transitions from frame to frame. Usually a motion of objects in static scene satisfies the assumption that a motion is described by the almost continuous function. The proposed modification of the BMA based on a statistical model of background is concerned to fast realization of the basic BMA. Let us consider the enhanced statistical model of a background.

The enhanced statistical model of background in a static scene is based on the following parameters: an average of frames I_{med} (medium values), a mean value $\mu(x, y)$, and a variance $\sigma^2(x, y)$ for K frames, which are described by the intensity function $I_t(x, y)$, where x and y are coordinates of a current pixel, t is a number of frame at moment t . A mean value $\mu(x, y)$ and a variance $\sigma^2(x, y)$ for K frames are calculated by Eqs. 9.10–9.11, where $w_t(x, y)$ are the weighting coefficients.

$$\mu(x, y) = \left(\sum_{t=1}^K w_t(x, y) \cdot I_t(x, y) \right) / \left(\sum_{t=1}^K w_t(x, y) \right) \quad (9.10)$$

$$\sigma^2(x, y) = \frac{1}{K-1} \left(\sum_{t=1}^K (w_t(x, y))^2 \cdot (I_t(x, y) - \mu(x, y))^2 \right) / \left(\sum_{t=1}^K w_t(x, y) \right)^2 \quad (9.11)$$

Let us notice, that Eq. 9.11 gives a unbiased estimator of variance $\sigma^2(x, y)$ for small K value, $K < 30$.

The weighting coefficients are used for minimization of spikes, which are maximally removed from the average of frames I_{med} (a noisy compensation) and are normally distributed. Equation 9.12 provides the estimations for $w_t(x, y)$, where standard deviation σ_{ex} is calculated from K neighbor frames in a spatial domain. The use of weighting coefficients $w_t(x, y)$ in the statistical model of background permits to build a robust background model without any training video sequences.

$$w_t(x, y) = \frac{1}{\sqrt{2\pi\sigma_{ex}^2}} \exp\left(-\frac{(I_t(x, y) - I_{med}(x, y))^2}{2\sigma_{ex}^2}\right) \quad (9.12)$$

The statistical model of background (Eqs. 9.10–9.12) is recalculated periodically because of lighting or meteorological changes. The enhanced model of background separates the background and the foreground moving objects. If Eq. 9.13 is executed, then this pixel is a foreground pixel, and otherwise. Value λ is equaled to 3 according to the rule of three sigma: all values of normal distributed random variable lay in the interval $\pm 3\sigma$ with not less reliability than 99.7 %.

$$|\mu(x, y) - I(x, y)|^2 \geq \lambda^2 \sigma^2(x, y) \quad (9.13)$$

Such statistical model is especially effective, when a set of sequential frames without moving objects can be provided. This requirement is ordinary for the outdoor and the indoor surveillance tasks. The worse results are received, when the moving objects with a small area (less 5–8 % of frame area) appear in static scene. The most problematic case connects with the moving object with a large sizes, when only the probable regions with or without motion are determined.

The background subtraction and the BMA are the main comparative methods of fast motion estimation. The background subtraction is the simplest motion estimation technique. For each current frame, the intensity values and the color components of each pixel are compared with the corresponding values of pixels in an initial averaged (sampling) frame of video sequence. As a result, the binary masks of moving foreground objects in a scene will be received. Such method is a noise-dependent. Therefore, a median filter or the mathematical morphological operators are applied for binary masks improvement. The filter parameters determine the sensitivity and the reliability of background subtraction method. The simplicity and the high computational speed are its main advantages. However, shadows, dynamic background, lighting change, and camera inaccuracy make this approach the non-used in practice.

More appropriate decision connects with the BMA application for a set of sequential frames. Experiments show that 25 frames processing (near 1 s) is a good decision that provides a delay for receiving of stabilized video sequence in 2–4 s. In surveillance systems as the urban surveillance or computer vision in the outdoor environment, such results are satisfied. The hardware realization by CUDA technology will reduce the processing duration in times.

The proposed procedure for motion estimation includes three steps:

- The local motion estimations by fast BMA modification.
- The accuracy improvement by using the Takagi-Sugeno-Kang model.
- The global motion estimation in a frame.

Let us consider local motion estimation by the BMA. The basic BMA has various interpretations such as full search, pattern search, and recursive search, among others [12]. The full search strategy provides the best results with the highest

computational cost. There are known its modifications such as three step search, four step search, block based gradient descent search, diamond search algorithm, adaptive rood pattern search, etc., which were developed to reduce the computational cost with a non-essential quality loss. All these fast search algorithms are based on the assumption that a block distortion measure increases monotonically around the global minimum. However, the search process can be easily trapped in one of the local minimums, a lot of which are included in any video sequence as noises, lighting changes, or dynamic textures. To avoid such problem, the Markov model with three states was proposed by Chen et al. [41] to provide an acceptance probability of being able to jump out of a local minimum. Also many BMA modifications were developed for various cases of motion estimation, for example, a motion estimation in noisy video sequences [42], fast BMA [43], Gaussian mixture model [44], a motion estimation by using Lie operators [45], a bilinear deformable BMA [46], a fuzzy logic BMA [47], etc.

A motion vector $\mathbf{MV}(d_x, d_y)$, for which an error functional e according one of metrics (Eq. 9.4) has the minimum value, is considered as a displacement vector for the given block. It shows the displacement of the left top corner in the marked block from a previous frame ($t - 1$) to a current frame t . The proposed BMA modification uses the transparent masks for moving objects and the opaque mask for a static background, which often involves periodical motions of textons. Usually such motion is not interesting for estimation and ignored. In the case of static scene, all values of background pixels can be set to a constant value, for example, -1 . When a motion is detected, the transparent mask is put on a visual object. Therefore, the intensity function describing a moving object will be available for estimation especially during the overlapping of visual objects. Forcibly maintained to negative constant values, the opaque background masks permit to reduce BMA calculations due to only the analysis in a neighbor region. The procedure of basic BMA is reactivated periodically (with interval 1 s) as an additional search of other moving objects, appearing in static scene. As a result, a set of the LMVs fields will be built for the chosen frames; the LMVs field shows the motion vectors between two neighbor frames. Such LMVs field is enough chaotic, and the following procedure is to separate of the LVMs as “good” and as “bad” motion vectors.

The field of LMVs includes the motion vectors, which describe an unwanted camera motion and objects motion in a scene. For such clustering, a novel fast method for detection of unwanted camera motion was developed based on Takagi-Sugeno-Kang (TSK) model. A fuzzy zero-order TSK model is adopted to infer the quality index: four different output fuzzy sets are defined to describe the quality of the matching, named as excellent, good, medium, and bad [48]. A zero-order TSK model is very simple, compact, and computationally efficient model, which permits to use the adaptive techniques. These adaptive techniques customize the MFs in such manner that the input data are modeled by a fuzzy system in the best way. Also a quite complex data behavior can be interpreted by using the “IF-THEN” fuzzy rules. In our experiments, the triangle, trapezoidal, and S -shape MFs to partitioning the LMVs were used. A view of these MFs is represented in Fig. 9.1, parameters a and b of S -shape membership are fitted empirically. Our recommendations are to use

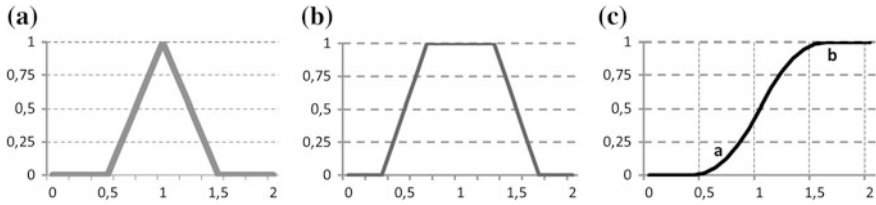


Fig. 9.1 A view of the MFs in the TSK model: **a** triangle, **b** trapezoidal, **c** S-shape

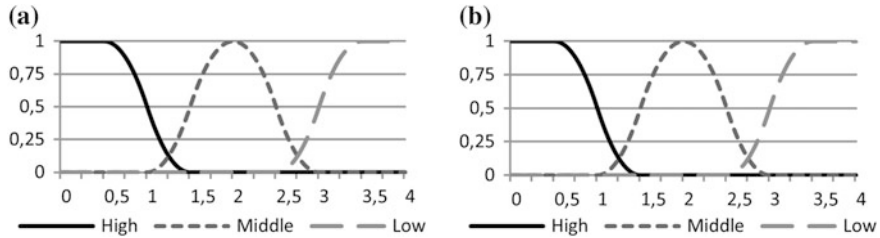


Fig. 9.2 A view of S-shape MFs: **a** for the non-noisy video sequence, **b** for the noisy video sequence

$a = 0.5$ and $b = 1.5$ for the non-noisy video sequence and $a = 0.75$ and $b = 1.75$ for the noisy video sequence. The recommended S-shape functions are situated in Fig. 9.2.

The inputs of fuzzy logic model are two error measures: an Euclidean distance e_i between expected and the real LMVs calculated in one of SAD, SSD, or MSD metrics (magnitude of vectors) $\mathbf{E}' = (e_1, e_2, \dots, e_i, \dots, e_n)$ and an angle between these LMVs c_i , $\mathbf{C}' = (c_1, c_2, \dots, c_i, \dots, c_n)$, where $i = 1 \dots n$. In this research, the similar approach from [49] to find error deviations d_i^e and d_i^c was used. The median values M_E and M_C of sets $\mathbf{E}' = (e_1, e_2, \dots, e_i, \dots, e_n)$ and $\mathbf{C}' = (c_1, c_2, \dots, c_i, \dots, c_n)$, $i = 1 \dots n$, respectively, are provided by Eq. 9.14.

$$d_i^e = e_i/M_E \quad d_i^c = c_i/M_C \tag{9.14}$$

Values of error deviations d_i^e and d_i^c from Eq. 9.14 are mapped in three different classes of accuracy: high, medium, and low. The lower values of error deviations are mapped to the best class, and otherwise. If the MFs are overlapped, then better class of the input fuzzy set is chosen.

The output of fuzzy logic model indicates a final reliability of estimation for a quality of the matching by using the TSK model. The quality index is a value in the range [0, 1]. It shows the quality of the LMVs, which are clustered in four classes: excellent, good, medium, and bad. The “IF–THEN” fuzzy rules defined for two inputs (error deviations d_i^e and d_i^c) are the following:

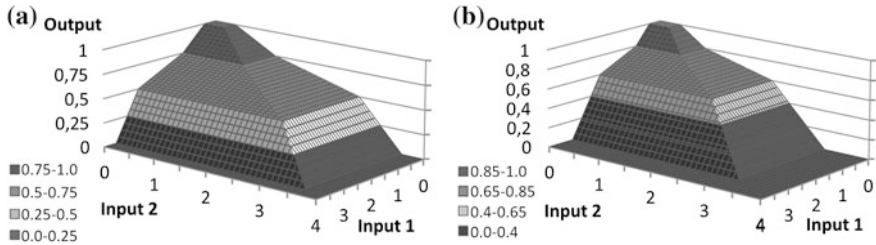


Fig. 9.3 The interpretation of the TSK model: **a** for the non-noisy video sequence, **b** for the noisy video sequence

- IF (both inputs = “high”) THEN (quality = “excellent”).
- IF ((one input = “high”) AND (other input = “medium”)) THEN (quality = “good”).
- IF (both inputs = “medium”) THEN (quality = “medium”).
- IF (at least one input = “low”) THEN (quality = “bad”).

Each of these four classes is mapped in a set of the constant values (1.0, 0.75, 0.5, 0.25, 0.0) [48]. During our experiments, the results for noisy video sequences were received with a set of the constant values (1.0, 0.85, 0.65, 0.4, 0.0). The TSK models for non-noisy and noisy video sequences with the sets of constant values (1.0, 0.75, 0.5, 0.25, 0.0) and (1.0, 0.85, 0.65, 0.4, 0.0), respectively, are show in Fig. 9.3. The TSK model permits to discriminate the LMVs with excellent and good quality and detect the best LMVs (with excellent and good values of indexes) in order to improve the final result.

Our following researches permitted to speed the LMVs calculation for both types of video sequences in the static scenes. Let us introduce the initial procedure, which will put an invisible grid on each frame adaptively to the frame sizes with 40–60 cells. The sizes of such grid are less than the frame sizes in order to reject the boundary areas of frame, which are more stressed to artifacts of instability. For five first frames in a scene, the LMVs estimations and their improvements by TSK model are calculated for all cells of this grid. For each cell, the information of reliable LMVs is accumulated under the condition, that 4–16 reliable LMVs are determined in a cell. According to the scene background, some of such cells are selected for the following analysis. Therefore, the LMVs of unwanted motion are calculated only in the selected cells, that permits to avoid the challenges of lighting change or moving foreground objects and reduce the number of analyzing cells in 1.5–3 times. Figure 9.4 provides such adaptive and fast technique for frame number 69 from video sequence “EllenPage_Juggling.avi”. Figure 9.5 illustrates several frames from the same video sequence with the imposed grid.

The selected cells include information only about unwanted motion that also increases an accuracy of the GMV in a frame. Let us consider the technique of the global motion estimation in static scene. The global motion caused by a camera movement is estimated for each frame by using a clustering model. On the one

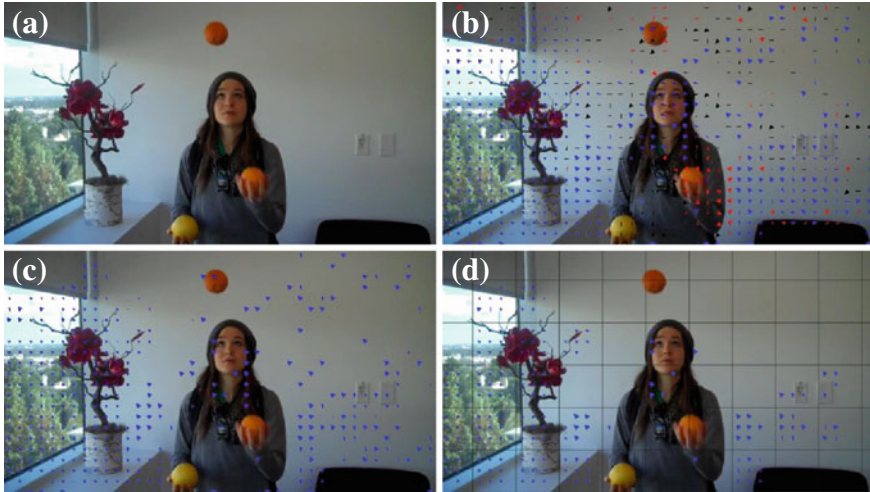


Fig. 9.4 The adaptive technique for LMVs estimation in static scene, video sequence “EllenPage_Juggling.avi”: **a** the initial frame 69; **b** all calculated LMVs; **c** the reliable LMVs based on the TSK model; **d** the reliable LMVs in the selected cells of imposed grid

hand, the LMVs of background are very similar in magnitudes and directions. On the other hand, they are essentially different from the motion vectors of foreground objects. The procedure, classifying the LMVs in two clusters – background and foreground, has the following steps:

- Step 1. The histogram H is built, which includes only valid LMVs with excellent and good values of indexes (near 30 % from all detected LMVs).
- Step 2. The LMVs are clustered by a criterion of the similar magnitudes.
- Step 3. The LMV with a maximum magnitude from background motion cluster is chosen as the GMV for a current frame.

The example of a histogram with valid LMVs is presented in Fig. 9.6.

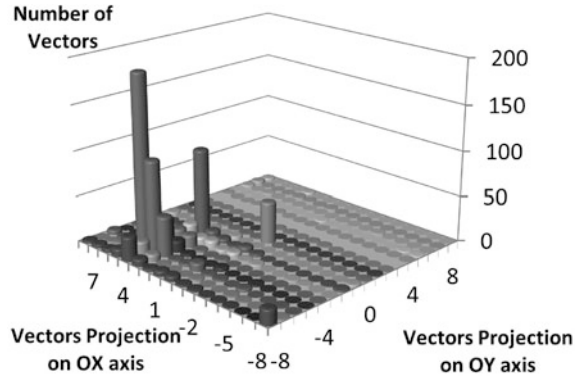
For the GMV detection, the TSK model can be also applied. The global motion includes two major components: a real motion (for example, a panning) and an unwanted motion caused by camera jitters. Usually an unwanted motion corresponds to a high frequency signal. Therefore, the low-frequency filtering can remove the unwanted motion.

The model, proposed in the research of Kyriakoulis and Gasteratos [28], was used to create a Smooth Motion Vector (SMV) calculated by Eq. (9.14). The low-pass filter of the first order requires the low computational cost and can be used in a real-time application.



Fig. 9.5 The LMVs estimation in static scene from video sequence “EllenPage_Juggling.avi”: all calculated LMVs are in the left column; the reliable LMVs based on the TSK model are in the middle column; the reliable LMVs in the selected cells of imposed grid are in the right column

Fig. 9.6 The example of a histogram with valid LMVs



9.4.2 Unwanted Motion Compensation

The unwanted motion compensation in static scenes is based on a smoothing of the GMVs. Any moving rigid object has the following states [50]:

- Appearance. The object is appearing in a scene.
- Mature. The object has been continuously tracked in some interval for approval that it is a foreground object, not a noise.
- Temporarily unavailable. The object temporarily loses its track because of being hidden, noised, or exited.
- Occlusion. The object is partially or totally hidden by other objects in a scene.
- Disappearance. The object may either already exit from a scene or be hidden by background objects such as buildings or trees.
- Reappearance. The object appears again after disappearing. The surveillance is restarted.
- Out of scene. The object has indeed moved away in a scene and its track considered terminated.

Only “appearance”, “mature”, “occlusion”, and “reappearance” are considered in the DVS task. For these purposes, the well-designed techniques are developed, including shadow compensation, lighting enhancement, and others challenges [51, 52].

The tuning procedure of a smoothing factor α , based on the analysis of previous 25 frames, was proposed. First, for k -sampling of frames, a global difference $GDiff_k$ is calculated by Eq. 9.15, where $|GMV_i|$ is a magnitude of global motion vector in frame I , $k > 25$.

$$GDiff_k = \sum_{i=k-25}^k \left| |GMV_i| - |GMV_{i-1}| \right| \tag{9.15}$$

Second, a value of α is chosen by Eq. 9.16, where $\alpha_{\max} = 0.95$ and $\alpha_{\min} = 0.5$ are maximum and minimum empirical values. In any case, the result from Eq. 9.16 is rounded to α_{\max} .

$$\alpha_k = \begin{cases} \frac{GDiff_k}{|GMV_{\max}|} \times \frac{\alpha_{\max}}{\alpha_{\min}} & \text{if } GDiff_k < |GMV_{\max}| \\ \alpha_{\max} & \text{if } GDiff_k \geq |GMV_{\max}| \end{cases} \quad (9.16)$$

The GMV in a frame is selected from the LMVs with maximum magnitude. It is easy may be done from a histogram with valid LMVs (Fig. 9.6). The total amount of LMVs with the similar magnitudes and directions are calculated. The LMV with the maximum value is considered as the GMV in a current frame.

9.4.3 Static Scene Alignment

In most algorithms for static scene stabilization, a motion inpainting is often considered as a scene alignment task [27]. At present time, two approaches of scene alignment are known. They are based on a frame scaling with reduction of frame sizes and a restoration of frame borders with conservation of frame sizes. For dynamic scenes, this is more complex task in comparison of static scenes.

For simple static scene, a scene alignment procedure can be simplified by a forcibly replacement of background from the statistical background model discussing in Sect. 9.4.1. In this case, only motion of moving objects ought to be compensated, and static scene alignment does not necessary. To make such “replaced” static scene more realistic, the procedures for rendering natural dynamic textures or other artifacts are required.

For each frame after calculation of smooth factor α , a module of smooth motion vector SMV_n is determined using Eq. 9.9. A magnitude of Undesirable Motion Vector (UMV) UMV_n is calculated by Eq. 9.17.

$$|UMV_n| = |GMV_n| - |SMV_n| \quad (9.17)$$

In the development of scene alignment method, a direction of vector SMV_n is normalized up to 8 directions with 45° step. For restoration of current frame, pixels are shifted on a value of Accumulated Motion Vector (AMV) AMV_n of unwanted motion by using Eq. 9.18. The stabilized location in a frame is determined from previous frames, beginning from a key frame, where m is a number of key frames in a video sequence.

$$AMV_n = \sum_{i=m}^n |UMV_i| \quad (9.18)$$

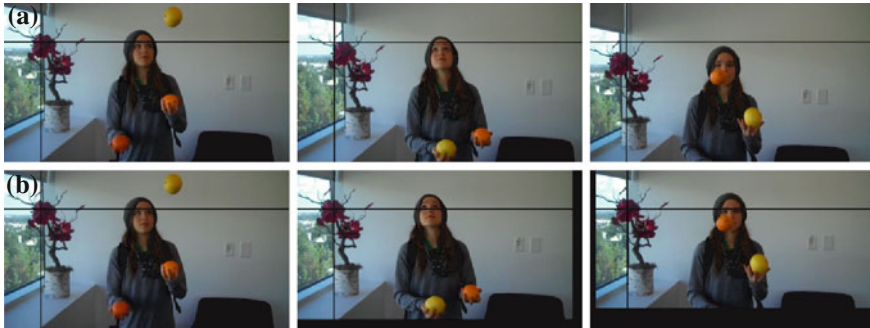


Fig. 9.7 The scaling resume for frames 540, 555, and 570 from video sequence “EllenPage_Juggling.avi”: **a** a non-stabilized video sequence, **b** the stabilized video sequence by changing of frame boundaries

The role of key frame is to represent the homogeneous regions in a following set of frames. The technique of key frame detection has been widely studied in [53, 54]. For static scenes, it is necessary to find the frames with the significant inter-frame difference. It means that the important motion changing occurs in a scene. Our recommendations connect with the limited power of such set, not more 25 frames. Then the following key frame ought to be chosen. The using of such technique permits to stabilize a video sequence with scaling. The sizes of such stabilized frames become less on 10–20 % relative the original video sequence that Fig. 9.7 demonstrates. Experiments show that the application of frame scaling for dynamic scenes may reduce the stabilized frame area up to 50–60 % that is not acceptable for the user.

All existing video inpainting algorithms can be broadly classified in two categories: Partial Differential Equation (PDE)-based methods and Texture Synthesis (TS)-based methods. The PDE-based image inpainting reconstructs the missing data spatially by extending the edges and filling the hole with smoothed color information by a diffusion process. In this case, a temporary nature of video sequence is ignored, and each frame is considered as an individual image. Such approach does not reproduce the texture and suffer from the blurred artifacts. This method is effective for restoring of small scratches or spots in archival footage.

The non-parametric sampling is an important class of TS-based methods, which uses the spatio-temporal patches extracted from the neighbor frames. The space-time patches called epitomes are created by a probabilistic learning of large number space-time patches taken from input video sequence [55].

The simple technique for static background frame borders restoration was applied in this research. Any static scene can be represented as two layers – foreground and background layers according to the affine model and as several layers – foreground and 2–3 background layers according to the perspective model of scene. The background layer (layers) is delivered by the statistical background model.

The foreground layer includes moving objects tracking by Kalman filter with following processing by the de-blocking filter with sizes 5×5 pixels to smooth the boundaries of the moving regions.

9.5 Video Stabilization Method in Dynamic Scenes

The analysis of dynamic scenes is the most complex issue in video processing. In literature, the researches devoting to DVS in dynamic scenes are sparsely represented. Before motion estimation, compensation, and inpainting, it is required to separate a video sequence on the relatively static scenes. The DVS task is solved for each separated scene, and then stabilized fragments ought to be collected in a whole stabilized video sequence. At this stage, the local smooth estimations of unwanted camera motion can be applied to smooth transitions between scenes. A valid scene transition does not require the additional processing. A sharp scene transition is needed in a future algorithm development or is executed by the user.

Section 9.5.1 provides a scene separation model. The recommended methods for motion estimation of background and moving objects are discussed in Sect. 9.5.2. The deblurring methods for the DVS task are proposed in Sect. 9.5.3. The unwanted motion compensation for dynamic scenes is represented in Sect. 9.5.4. At last, Sect. 9.5.5 includes the issues of motion inpainting in dynamic scenes by frames interpolation.

9.5.1 Scene Separation

For accurate DVS in dynamic scenes, a video sequence ought to be cut in separate relatively static scenes. Then the DVS task is partly added up to the DVS in static scenes discussed in Sect. 9.4, but with a dynamic background. The continuous outdoor shooting can be concerned to the complex cases, when the explicit cut of scenes is impossible, and the user's help is needed. All methods and algorithms of scenes cut can be divided in two categories. Methods from the first category use information from a service recording of video sequence, In other words, labels of beginning and ending of video scenes are located during a shooting by operator manually or by a video camera automatically in compliance with turn-on and turn-off modes of shooting [56].

Methods from the second category are based on the inner information of video sequence, and usually include two stages. On the first stage, frames histograms, configuration and number of feature points, color areas location, and other parameters are estimated [57]. On the second stage, the adaptive threshold values are determined, according to which a procedure of scenes cut is executed. The proposed algorithm, using the configuration of feature points, is realized cyclically from frame to frame and involves three steps:

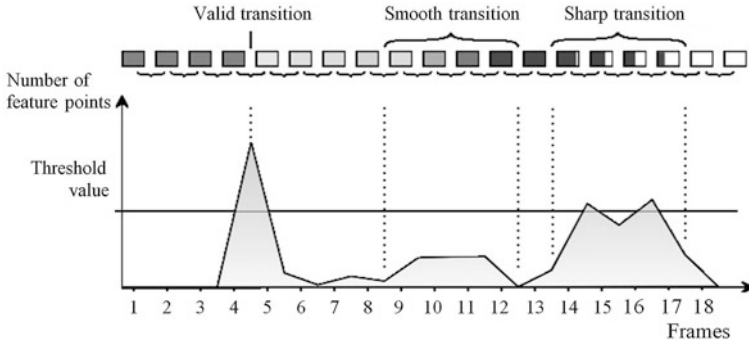


Fig. 9.8 Examples of scenes cut depending from a number of feature points, which have large displacement values

Step 1. The calculation of distance from each feature point to the center of frame j by Eq. 9.19, where x_{FPi} , y_{FPi} are coordinates of feature point i , x_{cj} , y_{cj} are coordinates of a center point of frame j .

$$D_{ij} = \sqrt{(x_{FPij} - x_{cj})^2 + (y_{FPij} - y_{cj})^2} \tag{9.19}$$

Step 2. The calculation of feature point displacement D_{ij} by using Eq. 9.20, where TH_j is a threshold for frame j for each feature point cyclically.

$$|D_{ij} - D_{ij-1}| < TH_j \tag{9.20}$$

Step 3. The calculation a number of feature points, which have large shifts in frame j by Eq. 9.21, where TH_{av} is an average threshold.

$$f(D, TH_j, j) = \text{count}(TH_j > TH_{av}) \tag{9.21}$$

If the function $f(\cdot)$ from Eq. 9.21 is in a local maximum in a current frame, then the previous and the following frames are the scene boundaries in video sequence. Such function displaces three types of dynamic scenes performance: valid transition, smooth transition, and sharp transition as Fig. 9.8 shows.

The smooth transition is the most complex case for DVS. Our recommendations connect with the sub-dividing of such transition in several sub-scenes for better visibility receiving.

9.5.2 Deblurring for Visual Objects with Complex Motion

The blur is a usual effect during a hand-held shooting. Unfortunately, the classical DVS approach based on the smoothness of camera motion leaves the blurriness artifacts untouched [39]. On the one hand, it is caused by camera jitters or both camera jitters and motion of objects. Such blur ought to be compensated before motion estimations, because the tracking of feature points or a motion field building are not reliable in the blurred frames. On the other hand, it may be assumed that only small amount of frames in the original video sequence are blurred frames. Such specialties permit to restore the original video sequence by a deblurring procedure more successful in comparison with other scopes of video restoration. The common strategy is to find the blurred frames, create the spatio-temporal blur kernel, and restore the blurred frames. Let us notice that a deblurring procedure can be applied only in the unreal-time applications.

The deblurring as a fundamental problem has been extensively studied in image processing and computer vision [58]. Its causes may be a high speed of moving objects or a directed high speed of moving camera. The most of existing methods are based only on the spatial blur kernel. The reasonable way of the classical approach is based on the single frames deblurring with following generation a temporal result. Through the non-aligned frames and a temporal coherence, even the classical multi-frame deblurring approaches are useless [59, 60].

The proposed approach removes the blurs caused only by a hand-held shooting. If a video sequence contains the blurring objects in a scene, then one of existing deblurring techniques may be applied, for example, the single image deblurring [61], the multi-image deblurring [62], the video deblurring by interpolation [37].

Let us suppose that an original video sequence is separated in scenes, and the following discussion will be concern to a single scene, which includes non-significant displacement describing by the simplest translation model because of neighbor frames similarity. For detection a blurred frame, the simple procedure of neighbor frames subtraction in each pixels with following total sum calculation of their absolute differences is used. Preliminary, the frames ought to be transformed to YUV-color space, and the brightness component Y is analyzed. A blur frame is characterized by a higher homogeneity distribution, and experiments show that such differences between normal and blur frames will be less than between two non-blurred frames in a non-stabilized video sequence.

A motion blurring due to camera jitters can be modeled as a spatial-temporal invariant convolution process described by Eq. 9.22, where f is a blurred frame, g is a non-blurred frame, p is the blur kernel (point spread function), n is the noise, and a $\text{sign} *$ means the convolution operator.

$$f = g * p + n \quad (9.22)$$

The recovery of frame g from a blurred frame f calls the image deconvolution problem. Two cases of image deconvolution: a non-blind deconvolution and a blind

deconvolution (the last one is actual for the DVS task), always varies in the different frames. In the case of a blind deconvolution, both a blur kernel p and a frame g are unknown. However, the prior assumptions of a kernel p and a frame g have to be made. The Gaussian-type optical blurring may be accepted, and also a frame g can be replaced by an adjacent non-blurred frame. In general, the motion blur kernel is expressed by Eq. 9.23, where C is a continuous curve of finite length in dimension R^2 denoting a camera trajectory, $v(x, y)$ is a speed function varying along a curve C . However, the estimation of camera trajectory and the definition of speed function is not a trivial task.

$$p = v(x, y)|_C \tag{9.23}$$

In particular case, when the blurred images are caused only by camera jitters, the simple procedure for blurred images detection is proposed. If small linear displacement in two neighbor frames is known, then the total sum of pixels differences will have less value between non-blurred and blurred frames or between two blurred frames in comparison with two non-blurred frames. The experimental results confirmed such assumption. Two variants of blurred frames restoration are possible, when a number of blurred frames is 1–2 and more. In the first case, the source for local blur kernel will be a decreased by the non-blurred frame to avoid the problems with shifts of boundaries. In the second case, the complex procedure of frames interpolation is required [63], which will replace the blurred frames by 2–5 interpolated frames. Let us specify the common expression for restoration of a single blurred frame. First, the local blur kernels are applied for each pixels as a weighed function $f_b(j, l)$ of a patch in a warping frame centered by pixel l . As a result, the deblur function $f_d(i, m)$ will be received in a spatial slicing window W_i , $i = n \times n$, $n = 7–11$ pixels centered by pixel m . Equation 9.24 provides the local patch deblurring, where $w(i, m, j, l)$ is a weight, determined by Eq. 9.25 as a Gaussian distributed value.

$$f_d(i, m) = \frac{\sum_{(j,l) \in W_{i,m}} w(i, m, j, l) f_b(j, l)}{\sum_{(j,l) \in W_{i,m}} w(i, m, j, l)} \tag{9.24}$$

$$w(i, m, j, l) = \exp\left(-\frac{(f_b(j, l) - f_d(i, m))^2}{2\sigma_{W_{i,m}}^2}\right) \tag{9.25}$$

Second, the deblurred frame is formed by the local deblurred patches, which can overlap each others. The more accurate approach connects with a patch-based texture synthesis [64]. To accelerate the deblurring process of overlapped patches, a sparse regular grid may be built, which also helps to avoid an over-smooth de-blurring effect caused by averaging of many patches.

Figure 9.7 shows the stabilization result by changing of frame boundaries in static scene. The similar effect is watched in dynamic scene. The using of

background model or/and a tile texture reconstruction of frame boundaries [65] permits to leave frame sizes non-changeable. More offensive problems appear, when a source of tile texture reconstruction is absent in the original video sequence. In this case, the texture synthesis using data from neighbor spatial regions can be applied to interpolate the missing pixels.

9.5.3 Motion Estimation of Background and Foreground

Usually the building of the background model without moving foreground objects is impossible. The local motion estimation can be done by using a multi-level motion model in affine or perspective scenes with depth. This is one of the main issues in the dynamic scene processing. For global motion estimation, a clustering procedure from Sect. 9.4.1 may be applied with more often re-calculations according to current camera speed. The motion estimation in a dynamic scene is implemented by a way including the following steps:

- The motion estimation of background.
- The motion estimation by feature tracking.
- The accurate estimation of moving objects by optical flow

The main idea of preliminary background motion estimation consists in detection such motion level (levels), which is owned to the background of dynamic scene with a high probability value. The basic model of multi-level motion is built on the following assumptions:

- Each pixel in a current frame is characterized by a motion vector, which connects it with a pixel in following frame.
- A set of various parametric motion levels exists. Each of levels uses the own probability model.
- A motion on each level is determined by a mixture Gaussian model.

In this model, a set of inner similar motion levels ml is determined, which correspond to the rigid objects situated on the different distances from a moving camera and regions rf in a frame. Let some motion structure $MS_i(x, y)$ in a point (x, y) of i th current frame corresponds to a level motion lm with θ_{lm} parameters. It means that in following $(i + 1)$ frame, a motion model $MS_{i+1}(x, y)$ will be shifted in a point $((x, y) + \mathbf{v}(x, y, \theta_{ml}))$ with an error of measurement $V_{xy, ml}$, which has value 1, if a structure owns to ml th motion level, and otherwise. Let us assume that any frame has a Gauss noise with standard deviation σ . Equation 9.26 provides a plausibility function, where $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{rf})$.

$$L(V, \theta) = - \sum_{xy, ml} V_{xy, ml} \frac{(MS_i(x, y) - MS_{i+1}(x + v_x(x, y, \theta_{ml}), y + v_y(x, y, \theta_{ml})))^2}{2\sigma^2} + C \quad (9.26)$$

The probabilities of motion values can be represented as the clustering maps of local motion estimations in each level and are provided by Eq. 9.27, where $V_{xy, ml}$ denotes a motion level. Each intensity level determines a motion level, and each motion level connects with own motion model of affine type usually.

$$P(V_{xy, ml} = 1 | MS_i, MS_{i+1}, \theta) \quad (9.27)$$

For background estimation, not all from a set of motion levels can be chosen but only those, which have better stabilization results; in other words the levels, where the moving objects are absent. After such pseudo-static levels extraction, a background motion for the LMVs estimation can be built. For each pixel, a Gaussian distribution $P(I_{xy, ml} | \theta)$ in RGB color space is determined by Eq. 9.28, where $I_{ml}(x, y)$ is the intensity value of pixel (x, y) on motion level ml , μ_i is a mean value in neighborhood, Σ_i is the covariance matrix, and $|\Sigma_i|$ is its determinant. Values μ_{ml} and Σ_{ml} are determined from a set of initial frames in dynamic scene.

$$P(I_{xy, ml} | \theta) = \frac{1}{(2\pi)^{3/2} |\Sigma_{ml}^{1/2}|} \exp\left(-\frac{1}{2} (I_{ml}(x, y) - \mu_{ml})^T \sum_{ml}^{-1} (I_{ml}(x, y) - \mu_{ml})\right) \quad (9.28)$$

The updating of such background model is not required because a scene is dynamic, and the background model will recalculated with a high frequency in comparison with a static scene. In general, the efficiency of background estimation is determined by a camera speed [66]. More speed value of camera means a less-successful background motion approximation.

The core of the DVS is the LMVs estimations of objects motions, and more acceptable decision for dynamic scenes will be a feature tracking approach. Two main strategies the SIFT and the SURF algorithms were investigated in dynamic scenes. The SIFT algorithm detects and describes the distinctive features based on difference of Gaussians of an image at different scales [67]. It detects the robust features, and builds a key point descriptor (for each feature), which is invariant to translation, rotation and scale. Such technique provides the accurate interframe key point matching and includes four steps, as mentioned below:

- The Difference-of-Gaussian (DoG) scale-space construction.
- The stable feature detection.
- The gradient orientation and magnitude assignment.
- The extraction of feature descriptor.

The first three items can be regarded as the step of feature detection, and the fourth one as the step of feature description. The SIFT feature is invariant to translation, scaling, and rotation, while at the same time it is quite robust to lighting change. A number of key points are restricted according to data interpolation, removal of low-contrast feature points, and feature points with high edge responses, using a threshold value, affine or 3D projection parameters. The detection of orientation is based on the local image gradient directions. In the neighborhood of feature point, Eqs. 9.29–9.30 calculate the gradient magnitude $M(x, y, \sigma)$ and the orientation $\theta(x, y, \sigma)$, where $L(x, y, \sigma)$ is a Gaussian-smoothed image at required scale; σ is a standard deviation; (x, y) is coordinates of pixel [67].

$$M(x, y, \sigma) = \sqrt{(L(x+1, y, \sigma) + L(x-1, y, \sigma))^2 + (L(x, y+1, \sigma) + L(x, y-1, \sigma))^2} \quad (9.29)$$

$$\theta(x, y, \sigma) = \arctg \frac{L(x, y+1, \sigma) + L(x, y-1, \sigma)}{L(x+1, y, \sigma) + L(x-1, y, \sigma)} \quad (9.30)$$

Then the orientation histogram with 36 bins (10° on an each bin) is created by the gradient magnitudes and the Gaussian-weighted circles with $\sigma = 1.5 \sigma$. The orientations, corresponding to local peaks (more 80 %), are assigned with the feature points. During a step of descriptor extraction, the local gradient data is used to create the future point descriptors (a set of 16 histograms aligned in a 4×4 grid, each with 8 orientation bins). Thus, the resulting descriptor contains 128 elements.

However, it is very difficult to achieve the software-based real-time computing of SIFT features due to its computational complexity. At present, the hardware architectures are designed. One of such architectures is implemented in a fully parallel hardware based on Field Programmable Gate Array (FPGA) with the SIFT feature description by a high-performance fixed-point Digital Signal Processor (DSP) chip. Such FPGA + DSP hardware module designed by Zhong et al. can be directly driven by the output of a regular video camera [68]. The system is able to detect the SIFT features in the images with sizes 320×256 pixels within 10 ms and takes merely about 80 μ s per a SIFT feature descriptor.

The SURF descriptor is based on convolutions and uses the Hessian matrix-based measure for a distribution-based detector [69]. The Hessian matrixes in continuous and discrete variants are presented in Fig. 9.9.

The Hessian matrix $\mathbf{H}(P, \sigma)$ in a point P is determined by Eq. 9.31, where $L_{xx}(P, \sigma)$, $L_{yy}(P, \sigma)$, $L_{yx}(P, \sigma)$, and $L_{xy}(P, \sigma)$ are convolutions the second derivative of Gaussian $G(P)$ with a function describing a frame I_p in a point P along OX direction, diagonal in the first quadrant, OY direction, and diagonal in the second quadrant respectively.

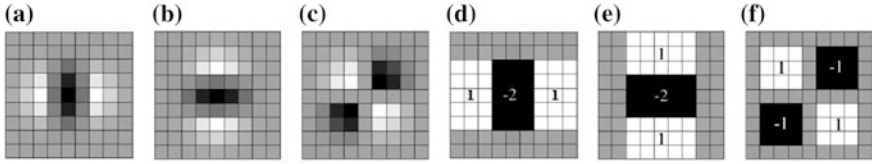


Fig. 9.9 A view of Hessian matrix: **a** a continuous variant along OX, **b** a continuous variant along OY, **c** a continuous variant along XY, **d** a discrete approximation along OX, **e** a discrete approximation along OY; **f** a discrete approximation along XY

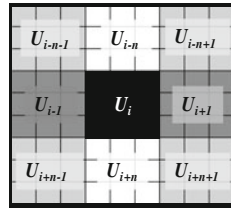


Fig. 9.10 A view of FELF detector

$$\mathbf{H}(P, \sigma) = \begin{bmatrix} L_{xx}(P, \sigma) & L_{xy}(P, \sigma) \\ L_{yx}(P, \sigma) & L_{yy}(P, \sigma) \end{bmatrix} \tag{9.31}$$

Equation 9.32 calculates the convolution along OX direction.

$$L_{xx}(P, \sigma) = \frac{\partial^2}{\partial x^2} G(P) * I_P \tag{9.32}$$

The SURF is based on the Haar wavelet response in the selected direction. It constructs a square region aligned to it, and extracts the SURF descriptor, which is invariant to rotation. The Haar wavelets are easy computed by integral images, if the window location in a point of interest is split up in 4×4 sub-regions. An underlying intensity pattern (first derivatives) of each sub-region is described by vector $\mathbf{V}_H = (\Sigma d_x, \Sigma d_y, \Sigma |d_x|, \Sigma |d_y|)$, where d_x and d_y are the Haar wavelet responses in horizontal and vertical directions, $|d_x|$ and $|d_y|$ are absolute values of corresponding responses. The overall vector will contain 64 elements. The resulting SURF descriptor is invariant to rotation, scaling, and lighting change. The SURF detector has a similar performance in comparison with other descriptors being at the same time faster.

In some applications, the DoGs detector (the Laplacian of Gaussian detector) is used, which shows the difference between two Gaussian smoothed images. Such approach is applied for the SIFT detector to build a scale space pyramid by sub-sampling images and convolving with differently sized kernels. Maxima and minima values are determined to find the response from the Difference of Gaussian

function in the 9-pixel neighborhood on the same scale level, at the scale level above, and the scale level below. The similar approach is used in a Finite Element Laplacian Feature (FELF) detector, where second order Gaussian smoothed image derivatives are used to compute the Hessian matrix [70]. Figure 9.10 provides a view of such detector.

The following feature points tracking by Lucas-Kanade algorithm [71] is a well-designed procedure, discussing in literature [12, 72].

The optical flow is a widely distributed method for accurate motion estimation in video sequences [73, 74]. It builds the motion vectors between two neighbor frames at time instants t and $(t + \delta t)$ in every pixel position with coordinates (x, y) . The intensity function $I(x, y, t)$ moves by δx , δy , and δt between two neighbor frames. Under the assumption that the intensity function, which describes a visual object, remains constant, the main equation of optical flow for motion estimation in video sequences, can be written in a view of Eq. 9.33.

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (9.33)$$

On account of the motion is small enough, Eq. 9.33 may be performed by expanding function $I(x, y, t)$ in a Taylor series by Eq. 9.34, where H.O.T. means high order terms, which are small enough and ignored.

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + \text{H.O.T.} \quad (9.34)$$

From Eq. 9.34, the Eq. 9.35 follows, where v_x and v_y are the (x, y) speed components, $\delta x/\delta t$, $\delta y/\delta t$, $\delta t/\delta t$ are the partial derivatives in coordinates (x, y, t) .

$$\begin{aligned} \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t &= 0 \\ \frac{\partial I}{\partial x} \frac{\delta x}{\delta t} + \frac{\partial I}{\partial y} \frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} \frac{\delta t}{\delta t} &= 0 \\ \frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} &= 0 \end{aligned} \quad (9.35)$$

The Eq. 9.35 applied to the gradient of intensity function is represented as symmetric covariance 3D structure tensor \mathbf{J}_S [12]. The eigenvalues $\Lambda = \{\lambda_k\}$, ($k = 1, 2, 3$) in neighborhood 3×3 pixels are characterized the local intensity displacements along two spatial axes OX and OY in Euclid space and a temporal axis. The intensity maps $\lambda_1(I)$, $\lambda_2(I)$, $\lambda_3(I)$, based on eigenvalues $\lambda_1(x, y, t)$, $\lambda_2(x, y, t)$, $\lambda_3(x, y, t)$ of local 3D structure tensor, provide the motion estimations in three dimensions.

To other known estimations of optical flow, the Bouguet approach [75] and the Horn-Schunck approach [76] can be mentioned. The Bouguet approach implements a sparse iterative version of Lucas-Kanade tracking feature points in the pyramids.

In the classical Horn-Schunck method, the optical flow estimations are performed as a variational problem, when a vector field is defined as a minimum of certain energy functional J under the assumption that the frame noise and the optical flow derivatives have a Gaussian distribution. This functional includes two terms: a data term provided by the optical flow constraint and a regularity term based on the gradient of the optical flow. The Eq. 9.36 shows a common view of functional J , where a weight parameter α shows a smoothness degree in a regularity term, Ω is a vector field.

$$J = \int_{\Omega} (I_x v_x + I_y v_y + I_t)^2 dt + \alpha^2 (|\nabla v_x|^2 + |\nabla v_y|^2) \quad (9.36)$$

The optical flow separates the spatio-temporal set of pixels in a set of “moving” points and a set of “static” points. For more accurate separation of such sets especially in noisy (type “salt-pepper”) video sequence, additional procedures can be recommended based on Lorentzian estimator, Tukey’s bi-weight estimator, German-McClure estimator, or Leclerc estimator. For Gaussian noisy video sequences, the improvement can be achieved by applying the higher-order statistics. The optical flow values are caused by an additive noise, which is often modeled by a Gaussian distribution (hypothesis H_1) or by a true motion in each pixel (hypothesis H_0) in Eq. 9.37, where $v_k(\bar{r})$ is a flow estimation in a frame k , $z_k(\bar{r})$ is an additive Gaussian noise, and $u_k(\bar{r})$ is the lighting variation caused by true motion [77].

$$\begin{aligned} H_0 : v_k^0(\bar{r}) &= z_k(\bar{r}) \\ H_1 : v_k^1(\bar{r}) &= u_k(\bar{r}) + z_k(\bar{r}) \end{aligned} \quad (9.37)$$

The computer cost of pixels separation on moving (active) and static ones is reduced, if the Gaussianity in the optical flow estimations will be detected. The classical measure of Gaussianity is the kurtosis, which is equaled to 0 for a Gaussian random variable. The kurtosis is determined by Eq. 9.38, where y is a random value, $E[\cdot]$ is an expectation.

$$\text{kurt}(y) = E[y^4] - 3(E[y^2])^2 \quad (9.38)$$

The kurtosis values for active pixels are significantly higher than those for static pixels with noise-induced optical flow values. The binary mask, which demonstrates a pixel activity, can separate pixels on moving and static more accurate that permits better motion compensation caused by camera jitters.

9.5.4 *Unwanted Motion Compensation*

The unwanted motion compensation can be realized by different ways: a smoothing of the GMVs, camera path estimation, and one of surveillance mode – a frame retargeting. Let us consider these cases particularly.

The challenge of the GMVs smoothing is closely tied with a key frames extraction in dynamic scene. The intervals between key frames can have different values. If key frames are extracted, then the task is transformed for the GMVs smoothing in static scene, as this was considered in Sect. 9.4.1.

For dynamic scenes, the extraction of key frame is more difficult task in comparison to the static scene. The accurate inter-frame differences can be calculated as the correlation of RGB color channels, color histogram, moments of inertia, or descriptors based on feature points [78–80]. Also the difference between the current frame and the follow key frame ought to have a significant value. The mentioned characteristics have high computational cost, and do not propose a real-time application. In this research, a way for calculation of feature points was chosen to detect the key frames in dynamic scene. If a number of feature points is essentially changed in local regions of frame, then this frame is marked as a key frame.

The procedure of camera path estimation requires a pre-segmentation of background, feature descriptors extraction, and background objects recognition in adjacent key frames. The main idea is to find “good” corresponding points and track their displacements in all frames between key frames. As a result, two envelope curves from points with high and low ordinate values will be built because of jump camera jitters. Then the curve with middle ordinate value can be interpolated by linearization (if an angle of shooting does not change, and only transitions are available) or by bilinear or bi-cubic functions (if an angle of shooting changes, and transitions/rotations are available) under the assumption that the ideal trajectory of camera is a blend curve. Such points compulsory own to background. Several pairs of such corresponding points can be determined in adjacent key frames to increase a reliability of camera path estimation.

If displacements between real and interpolated trajectories are calculated for each frame, then the locations of moving foreground object may be recalculated by the displacement values under assumption that a motion is defined by the affine motion model. This approach is enough complex: to find the corresponding points with the following interpolation is a separate high computational task. It can be simplified, if the calibrated video camera will be used. However, the camera calibration for dynamic scenes is too difficult and continuous procedure.

A video retargeting is one of ways for the DVS, when the object of interest is held in the center of a frame in the stabilized position. Such surveillance can be realized by some techniques beginning from the feature points tracking to Kalman filter application. In this section, let us discuss the application of Kalman filter and particle filter for the DVS tasks. Also the interesting issue connects with the re-targeting of non-rigid objects.

The Kalman filter provides a recursive solution to the linear optimal filtering and applies in static and dynamic environment [81]. Feature evaluation by Kalman filter during a tracking process is under the following constrains [82]:

- The confidence estimations and discriminative ability of a feature has the Gaussian distribution.
- Features with higher discriminative ability should have larger confidence estimation, and vice versa.

First, the state of Kalman filter is represented as the combination of confidence estimations $W_t = \{w_t(1), w_t(2), \dots, w_t(N)\}$ and variation “speed” $\Delta W_t = W_t - W_{t-1}$ of each feature, where $w_t(i)$ is a confidence estimation of i th feature. Second, the measurements of the filter $S_t = \{S_t(1), S_t(2), \dots, S_t(N)\}$ are provided by a frame in a time instant t . The predicted equation and the measurement equation of Kalman filter are calculated by Eq. 9.39, where $I_{N \times N}$ is an identity matrix describing a targeting object, u_t is a displacement and v_t is a speed, u_t and v_t are both Gaussian noised functions.

$$\begin{cases} \begin{pmatrix} W_{t+1} \\ t+1 \end{pmatrix} = \begin{pmatrix} I_{N \times N} & I_{N \times N} \\ 0 & I_{N \times N} \end{pmatrix} \begin{pmatrix} W_t \\ \Delta W_t \end{pmatrix} + u_t \\ (S_t) = (I_{N \times N} \ 0) \begin{pmatrix} W_t \\ \Delta W_t \end{pmatrix} + v_t \end{cases} \quad (9.39)$$

The particle filter is an estimation algorithm for implementing a recursive temporal Bayesian filter by Monte Carlo simulations. It represents a posterior state of moving object by a set of random samples with associated confidence estimations. The feature evaluation by using the particle filter can calculate the confidence estimations or discriminative abilities for non-Gaussian and non-linear distribution [83]. The core of this procedure is to define such feature set, each feature in which is seen as a particle. As a result, a weighted sample of particles at frame t , $\{(i, w_t(i))\}$, $i = 1, 2, \dots, N$, where i denotes the i th feature (particle), is created. The iterations of this process evaluate the temporal consistency even a variation of features is a non-linear and a non-Gaussian [84].

The challenges of non-rigid deformations, rotations, appearance, occlusions, and drifting are known as the template update problem or stability-plasticity dilemma. However, most of approaches such as robust learning algorithms [85], different learning paradigm [86], multiple different classifiers [87], or a conservative learning framework [88] are limited to a bounding-box-based representation.

To avoid of inaccuracy, the segmentation of non-rigid objects can be represented by the deformable parts model [89] or models obtained via the generalized Hough-transform [90, 91]. These methods need large amount of labeled training data, which cannot be provided during a tracking of unknown objects.

In the research of Godec et al. [92], a novel tracking-by-detection approach is proposed. It is based on the online Hough ferns and a couple of procedures: the

voting-based detection and back-projection with a rough GrabCut segmentation [93]. The randomized Hough ferns use simple pixel comparisons as splitting tests that allows the robustly detection of the non-rigid objects. The voting-based detection procedure has a valid geometric relation. The back-projection procedure roughly separates the object from the background pixel-wise. Such approach is very perspective for non-rigid natural objects and blurred objects segmentation.

The random ferns are such modification of random forest, which is based on independent flat test structures instead of tree-like structures [94, 95]. For S binary tests, the best matching class c for a given image sample \mathbf{v} is estimated by Eq. 9.40 assuming a uniform prior distribution over all classes.

$$\begin{aligned} c &= \arg \max_c P(c|\mathbf{v}) = \arg \max_c P(c|x_1, x_2, \dots, x_S) \\ &= \arg \max_c P(x_1, x_2, \dots, x_S|c) \end{aligned} \quad (9.40)$$

However, the joint features distribution over all tests cannot be modeled in practice. The features are grouped in several independent sets x_S . If a power of set is equaled 1, then a well-known Naïve-Bayes formulation is acceptable (Eq. 9.41).

$$P(x_1, x_2, \dots, x_S|c) = \prod_{c=1}^S P(x_S|c) \quad (9.41)$$

The Random Ferns is based on a semi-Naïve-Bayes formulation of Eq. 9.41 using larger feature sets and expressed by Eq. 9.42, where \tilde{x}_m denotes a set test and M is a number of used groups.

$$P(x_1, x_2, \dots, x_S|c) = \prod_{m=1}^M P(\tilde{x}_m|c) \quad (9.42)$$

Godec et al. show that Naïve formulation can be used for tracking, if $P(x_n - c)$ is modeled by using histograms instead of binary features [96]. However, binary tests can be also interpreted as a semi-Naïve formulation. For tracking of unknown objects, a node optimization is difficult because of the limited training data in several frames. A special tree-growing scheme was proposed by Saffari et al. [85]. The more complex statistics use the Hough-transform.

9.5.5 Motion Inpainting in Dynamic Scenes

The motion inpainting in dynamic scenes can be considered as the task of missing data restoration. Many interesting methods are designed for texture reconstruction, for example, image inpainting by contourlet transform [97], regularized image restoration [98], etc. More reasonable approach is the information extraction from

the sharp neighbor frames. In this research, two cases are applied: the frame boundaries restoration and the frames interpolation between key frames.

Frame boundaries restoration in dynamic scenes includes two cases, when the missing pixels can be taken from the previous frame and when it is impossible. First, the assumption is declared that the missing pixels can be founded. A pseudo-panoramic background may be built from neighbor stabilized frames, and it will be the source of tile texture reconstruction by a line or a field of textons with following inpainting of moving foreground objects in the reconstruction area [65]. Second, if the reconstruction information is absent (very sharp jitters respect to the whole video sequence or the sequential blurred frames), the decision is accepted about removal such frames and adding the interpolated frames under the assumption of smooth camera motion.

For the DVS task, it is enough to apply the simplest linear model of background and foreground objects motion. In this case, the point coordinates (x_n, y_n) in reconstruction frame n are calculated by Eq. 9.43.

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = (n - 1) \cdot \begin{bmatrix} x_i - x_{i-1} \\ y_i - y_{i-1} \end{bmatrix} \quad (9.43)$$

By using a normalized correlation function, the location of slicing window 11×11 or 15×15 pixels from stabilized area of a recoverable frame and a pseudo-panoramic background is determined. Such value shows the background place suitable for restoration of frame boundaries. Then a field of textons is replaced in a missing area of frame with corresponding stitching procedures [99, 100]. If a foreground moving object is in this part of scene, then its image is also restore according to linear motion model but with the own shifts. Let us notice that the proposed frame boundaries restoration is out from border of real-time application because the analysis of sequential frames can be required, and also the work with texture is usually a durable process.

The task of missing frames interpolation is a separate complex issue in digital video processing. In the current research, this is an additional aspect of motion inpainting in dynamic scenes and concerns only to the blurred frames. The goal is to improve a video sequence, when several sequential frames are blurred very strongly, and the procedure proposed in Sect. 9.5.2 is a non-useful processing.

In such particular case, the blurred frames are removed. Then very complex and computer high cost analysis is initiated for such pseudo-static scene. The main idea is to build the previous and the following trajectories of moving objects in a missing interval. The procedure is based on interpolation of two data types: background and foreground. In literature, the main attention gives consideration for interpolation of moving foreground objects in a scene. Three main approaches, among others, are used:

- The interpolation based on functions [101–104].
- The interpolation based on autoregressive modeling [105, 106].
- The interpolation using Markov random fields [107–112].

Objectively, if more blurred frames are in a video sequence, then the results of interpolation will be worse especially in the dynamic scenes.

9.6 Discussion of Experimental Results

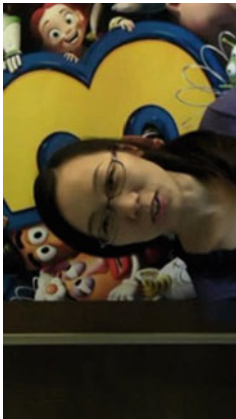

Two main directions of the experimental work were determined: the motion estimations based on the proposed methods (Sect. 9.6.1) and the stabilization estimations (Sect. 9.6.2). Six video sequences received from the static camera and eight video sequences received from the moving camera were used in this research. The titles, URL, and snapshots of some investigated static and dynamic scenes are presented in Tables 9.2 and 9.3, respectively.

All experiments were executed by the designed software tool “DVS Analyzer”, v. 2.04, which was developed in Laboratory of Image and Videos Processing (Department of Informatics and Computer Technique, Siberian State Aerospace University). The software tool “DVS Analyzer” has two modes: the pseudo real-time stabilization of video sequences, which are broadcasted from the surveillance cameras (the simplified processing) and the unreal-time stabilization of available video sequences (the intelligent processing). The architecture of the software tool includes the extended set of program modules, which can be designed or developed independently each from others. The Pre-processing Module, the Motion Estimation Module, the Motion Compensation Module, the Motion Inpainting Module, the Module of Quality Estimation, the Core Module, and the Interface Module are the main components of the software tool “DVS Analyzer”. Let us briefly discuss the functionality of each program module.

The Pre-processing Module involves various spatio-temporal filters such as the auto-contrast filter, the temporal 2D_cleaner filter, and the adaptive Gauss filter, which are applied to frames representing in color RGB-, HSV-, and YUV-spaces. Also this module divides a video sequence in the scenes. The Motion Estimation Module calculates the LMVs by the BMA and feature correspondences. It builds the GMVs in frames by using of fuzzy TSK model. The Motion Compensation Module determines the SMVs in each frame based on the GMVs. The Motion Inpainting Module realizes the frame stabilization by a re-calculating of an original frame according to the U MVs in scenes. The procedure for restoration of frame boundaries and the deblurring procedure are executed in this module. The Module of Quality Estimation provides the quality comparison of original and stabilized video sequences according to PSNR and ITF metrics. The Core Module controls and coordinates the work of all other Modules and includes program codec for a pre-processing and saving of the stabilized video sequences. The Interface Module displays the received results in file/monitor/printed version.

The software tool “DVS Analyzer”, v. 2.04 was designed in the Rapid Application Development Embracadero RAD Studio 2010. Some external software tools were used: the libraries “Video for Windows” for initial processing and “Alpha-Controls 2010”, v. 7.3 for enhanced user interface and a video codec “K-Lite Codec

Table 9.2 The description of investigated static scenes

Title, URL	Snapshot	Resolution	Frames	Motion type
<p>“SANY0025_xvid.avi” http://cpl.cc.gatech.edu/projects/videostabilization/</p>		<p>640 × 360</p>	<p>445</p>	<p>Slow motion of camera, motion of large object</p>
<p>“If_juggle.avi” http://cpl.cc.gatech.edu/projects/videostabilization/</p>		<p>480 × 360</p>	<p>460</p>	<p>Stable camera, fast motion of small objects</p>

(continued)

Table 9.2 (continued)


Title, URL	Snapshot	Resolution	Frames	Motion type
"akiyo.avi" http://see.xidian.edu.cn/vips/database_Video.html		352 × 288	300	Stable camera, motion of large object

Table 9.3 The description of investigated dynamic scenes

Title, URL	Snapshot	Resolution	Frames	Motion type
"Cat_orig.avi" http://www.youtube.com/watch?v=4XOCRIDA4w4		640 × 360	1582	Camera movement, fast motion of large object, lighting change
"Gleicher1.avi" http://cpl.cc.gatech.edu/projects/videostabilization		640 × 360	495	Camera movement, complex scene, low-contrast background
"Sam_1.avi" http://cpl.cc.gatech.edu/projects/videostabilization		640 × 360	278	Camera movement, fast motion of object, irregular motion

Pack”, v. 8.0. The experiments were performed on a computer with the following configuration: CPU Intel Core I5.760, 4 Gb RAM, Nvidia GeForce 460GTX, Windows 7 64 bit.

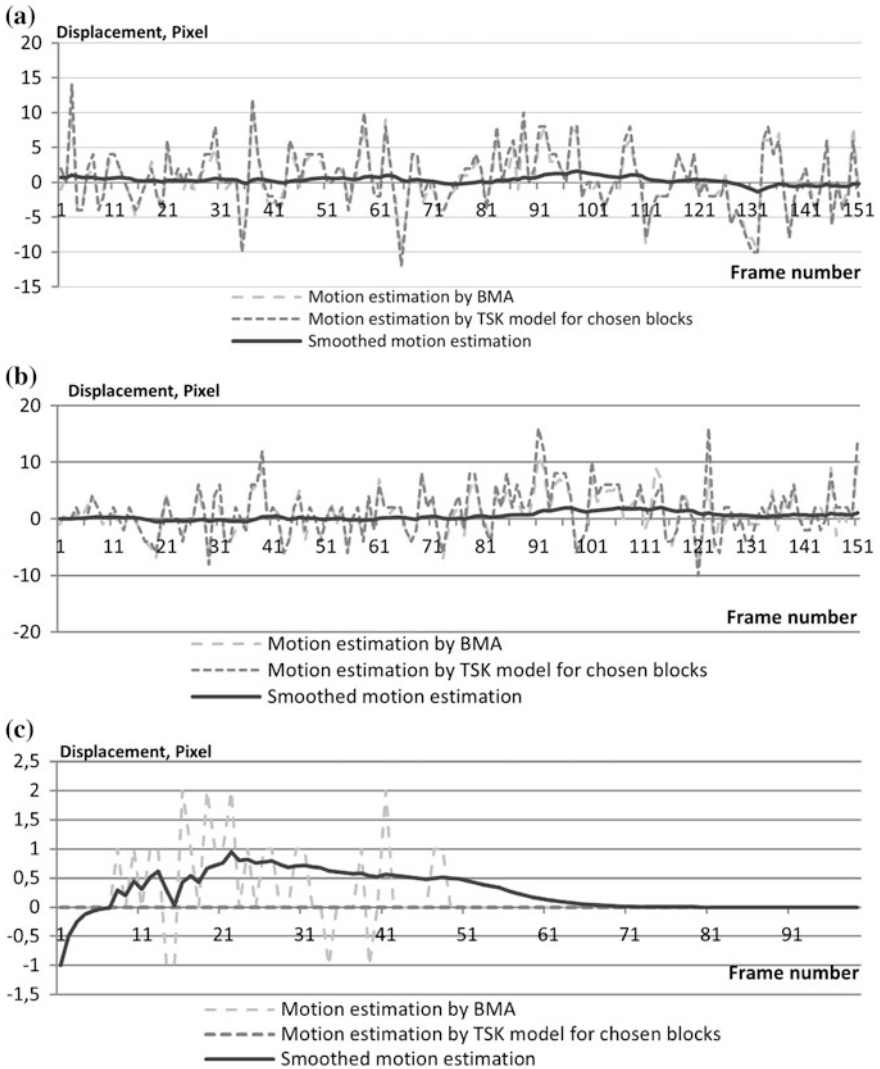


Fig. 9.11 Graphics of motion estimation and compensation results in static scenes: **a** “SANY0025_xvid.avi”, **b** “lf_juggle.avi”, **c** “akiyo.avi”

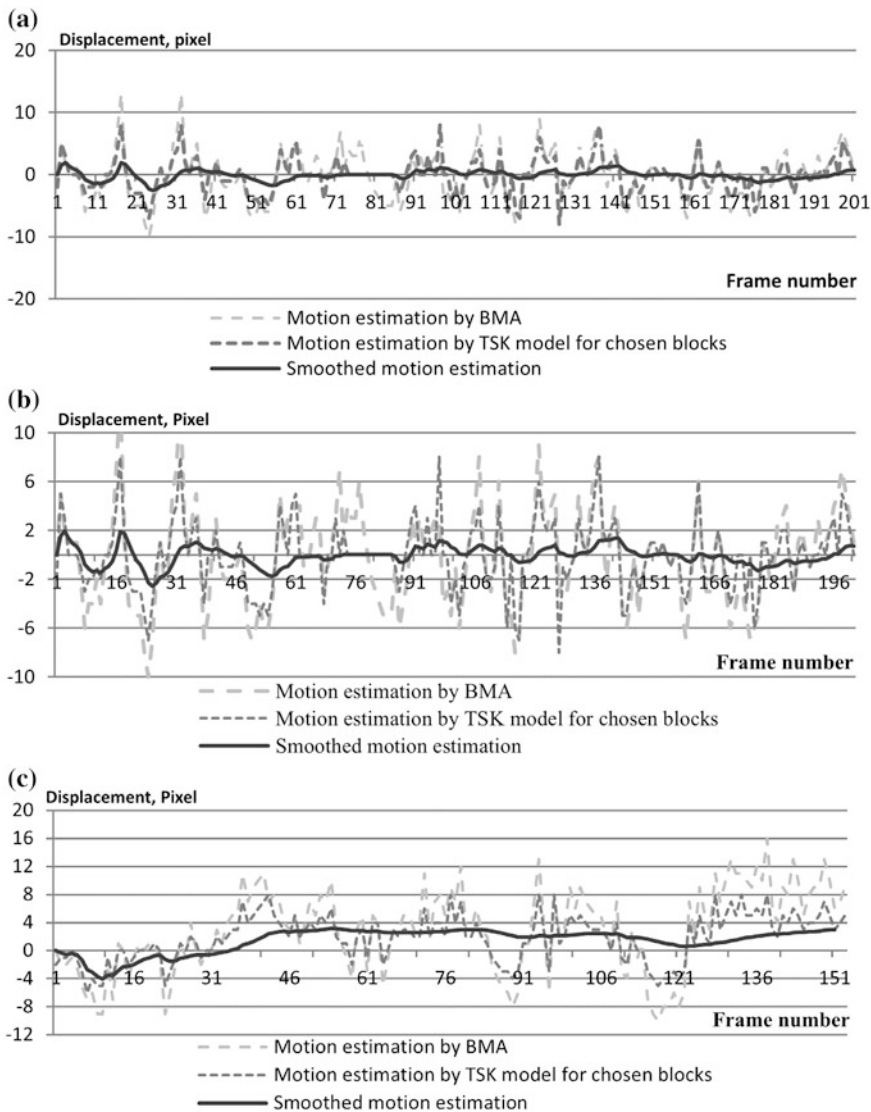


Fig. 9.12 Graphics of motion estimation and compensation results in dynamic scenes: a “Cat_orig.avi”, b “Gleicher4.avi”, c “Sam_1.avi”

9.6.1 Experimental Results for Motion Estimations

The experimental graphics for motion estimation and compensation are represented in Figs. 9.11 and 9.12 for static and dynamic scenes, respectively.

The motion estimations in static scenes by using the TSK model provide more accurate estimations of the global motion due to the fact that the motion of

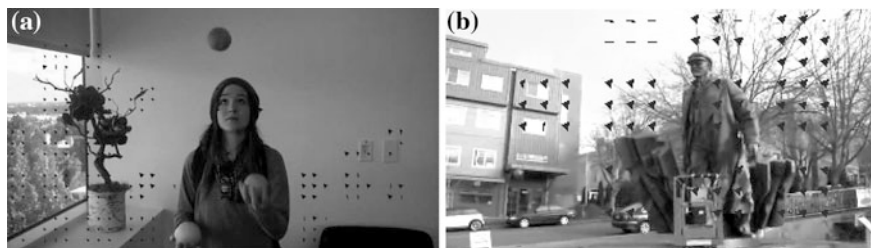


Fig. 9.13 The frames with “rejected” regions for motion estimations: **a** “EllenPage_juggling.avi”, **b** “Gleicher1.avi”

foreground objects is not considered. Such algorithmic specialty is well demonstrated in video sequences (“road_cars_krasnoyarsk_synthetic.avi”; “akiyo.avi”, “Butovo_synthetic.avi”). In original video sequences with static scenes, these differences are less and appear only in the GMV estimations. For video sequences including many small sizes objects (“lf_juggle.avi”, “Butovo_synthetic.avi”), the algorithm for motion estimations by using the TSK model shows the best results increasing ITF value up on 3 dB. For the dynamic scenes stabilization, the detection of the SMVs is the most important step. Such SMVs detection is executed based on the GMVs in a frame.

The non-contrast regions in frames decrease the quality of stabilization (“EllenPage_Juggling.avi”, “Gleicher4.avi”, “Gleicher1.avi”) and show the unpredicted results. Therefore, the non-contrast regions do not process in the most of frames. The frames with such “rejected” regions are represented in Fig. 9.13.

If the GMVs are estimated inaccurately, then the quality of following stabilized video sequence will decrease. The developed algorithm of motion estimation is non-sensitive to the large moving foreground moving in video sequences “EllenPage_Juggling.avi”, “Sam_1.avi”, “Cat_orig.avi”, “Cleicher3.avi”. The best global motion estimations were received for video sequence “EllenPage_Juggling.avi” due to ignoring the fast moving of foreground objects.

9.6.2 Experimental Results for Stabilization Estimations

The objective estimations of the DVS quality were calculated by Mean-Square Error (MSE) and PSNR metrics between a current frame I^{cur} and a key frame I^{key} , which are expressed in Eqs. 9.44–9.45, where I_{max} is a maximum of pixel intensity, m and n are the frame sizes along OX and OY axes.

$$MSE = \frac{1}{m \times n} \sum_{y=1}^m \sum_{x=1}^n (I^{cur}(x, y) - I^{key}(x, y))^2 \quad (9.44)$$

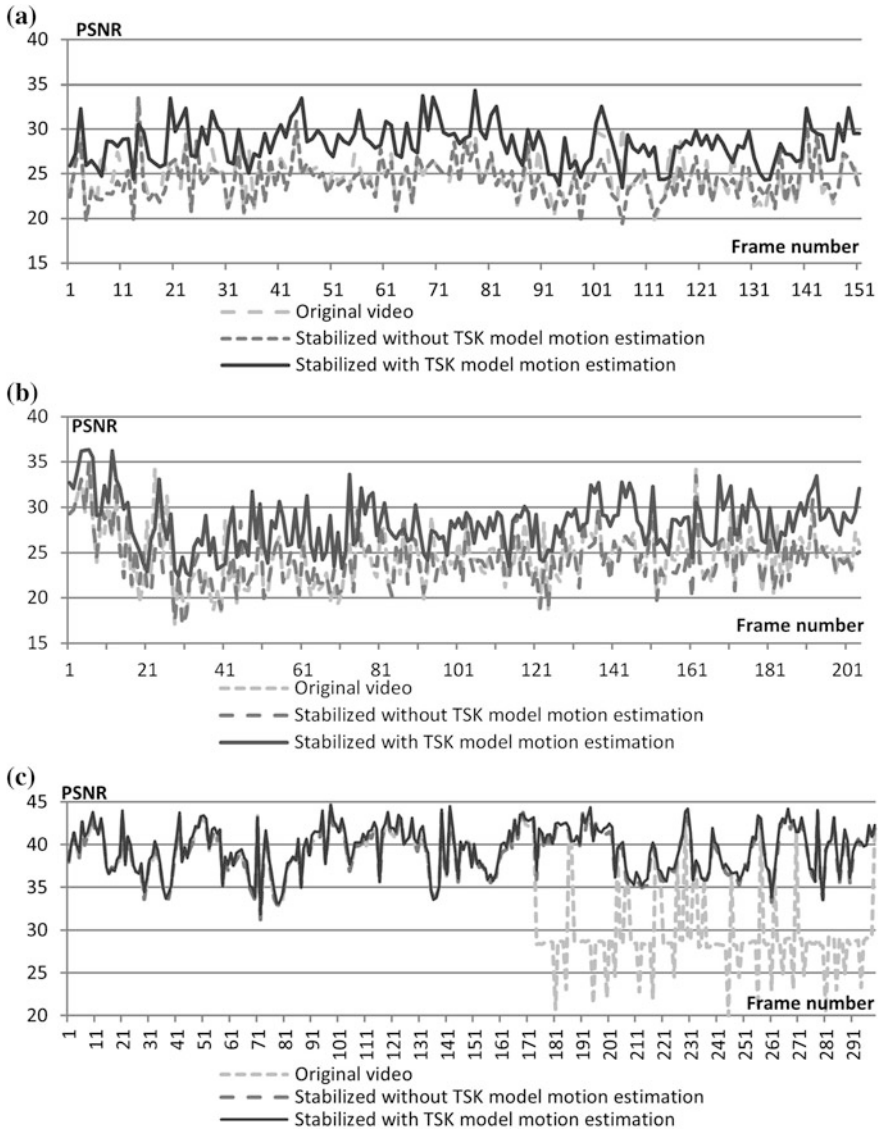


Fig. 9.14 Graphics of stabilization quality in static scenes: **a** “SANY0025_xvid.avi”, **b** “lf_juggle.avi”, **c** “akiyo.avi”

$$PSNR = 10\log_{10}\left(\frac{I_{\max}^2}{MSE}\right) \tag{9.45}$$

The PSNR metric is useful for estimations between neighbor frames. The quality of the ITF estimations provides the objective estimation in whole video sequence.

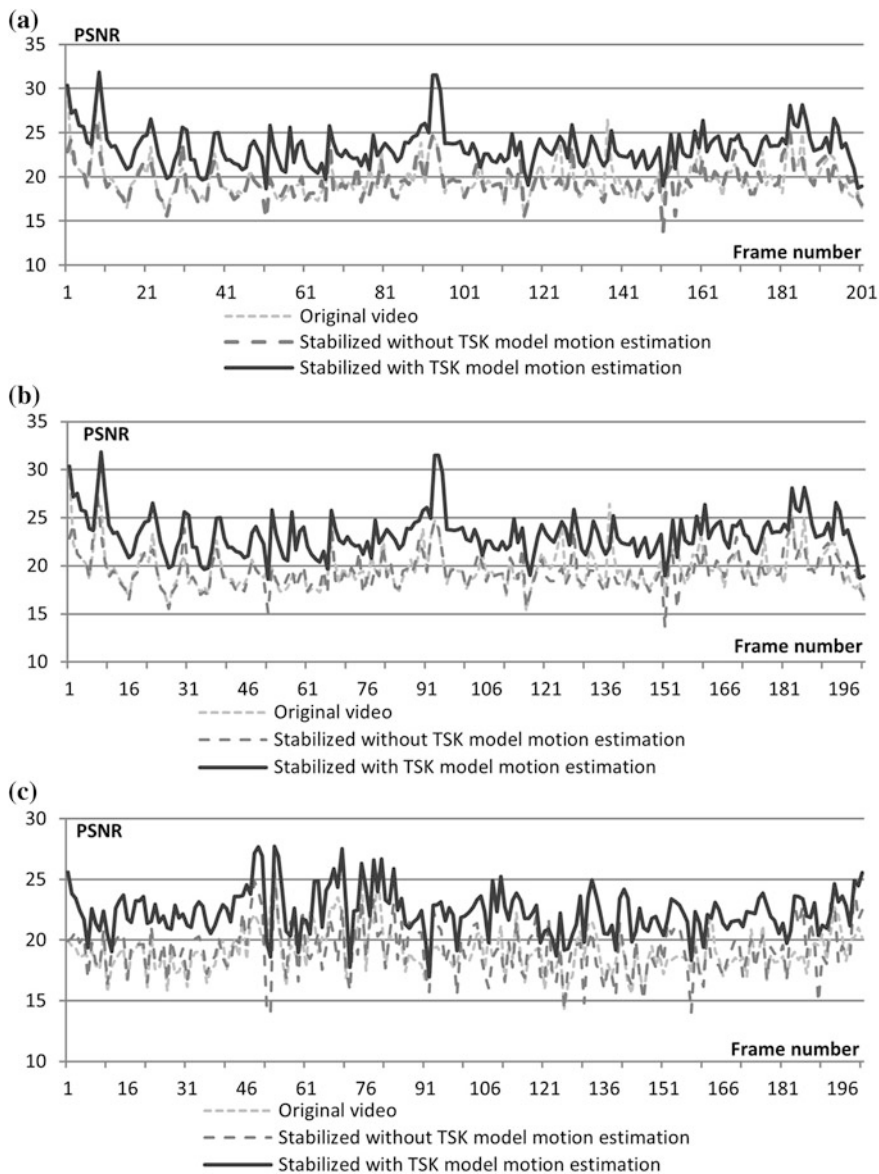


Fig. 9.15 Graphics of stabilization quality in dynamic scenes: **a** “Cat_orig.avi”, **b** “Gleicher4.avi”, **c** “Sam_1.avi”

The ITF of the stabilized video sequence is almost higher than the ITF of the original video sequence. This parameter is calculated by Eq. 9.46, where N_{fr} is a frame amount in a video sequence.

Table 9.4 ITF estimations for static scenes

Video sequence	ITF estimations (dB)		
	Original	Without TSK model	With TSK model
road_cars_krasnoyarsk.avi	22.70482	22.80707	25.91258
SANY0025_xvid.avi	20.5389	21.09076	23.79189
If_juggle.avi	24.30286	24.37177	28.06012
akiyo.avi	35.92952	39.14661	39.53257
EllenPage_Juggling.avi	24.65855	25.23049	28.58255
Butovo_synthetic.avi	22.26415	27.19789	27.20789

Table 9.5 ITF estimations for dynamic scenes

Video sequence	ITF estimations (dB)		
	Original	Without TSK model	With TSK model
Cat_orig.avi	25.07131	26.47094	28.14086
Gleicher4.avi	19.29703	19.50634	23.18371
Sam_1.avi	19.09737	19.28141	22.20112
Gleicher1.avi	18.86996	19.48223	22.78846
Gleicher2.avi	19.91954	20.36718	24.56673
Gleicher3.avi	16.55214	16.71899	20.12285
new_gleicher.avi	17.28921	17.81638	21.70575
yuna_long_original.avi	17.84131	18.94389	21.46971

$$ITF = \frac{1}{N_{fr}} \sum_{k=0}^{N_{fr}} PSNR_k \tag{9.46}$$

The experimental graphics for stabilization quality are represented in Figs. 9.14 and 9.15 for static and dynamic scenes respectively. As shown from Figs. 9.14 and 9.15, the PSNR estimations of the stabilized video sequences are always higher than the PSNR estimations of the original video sequences.

Tables 9.4 and 9.5 contain the ITF estimations for all (used in experiments) whole video sequences with static scenes and dynamic scenes, respectively.

As it seems from Tables 9.4 and 9.5, the video stabilization results are different for various video sequences because of varied foreground and background content, moving objects, lighting, noisy, and the shooting condition. The using of the TSK model provides the increment of the ITF estimations up on 3–4 dB or 15–20 %.

The stabilization and temporal results of “Deshaker”, “WarpStabilizer”, “Video Stabilization with Robust L1 Optimal Camera Paths”, and our developed “DVS Analyzer” are situated in Table 9.6.

Table 9.6 Comparison of stabilization algorithms for static and dynamic scenes

Video sequence	Algorithm							
	Deshaker		WarpStabilizer		Video Stabilization with robust L1 optimal camera paths		DVS analyzer	
	ITF (dB)	Time (s)	ITF (dB)	Time (s)	ITF (dB)	Time (s)	ITF (dB)	Time (s)
EllenPage_Juggling.avi	25.61	3.53	26.68	4.53	27.33	3.17	28.58	3.54
Gleicher4.avi	20.33	1.89	19.15	2.78	20.45	1.44	23.18	1.66
Sam_1.avi	20.09	1.22	20.27	2.65	20.58	1.01	22.20	1.23
road_cars_krasnoyarsk	22.31	1.45	21.48	2.15	25.2	1.29	25.91	0.24
SANY0025xvid.avi	23.53	1.33	22.7	1.87	22.74	1.34	23.79	0.17
lf_juggle.avi	26.65	1.22	24.41	1.64	26.15	1.18	28.06	0.15

The ITF estimations of the proposed software tool “DVS Analyzer” provides better results (at average 1–3 dB or 5–15 %) with the lower processing time relatively the existing software tools.

9.7 Conclusion and Future Development

In this chapter, the intelligent methods for digital video stabilization in static and dynamic scenes were developed. The extended literature review represents the state-of-art in the DVS till the present time. In this research, many novel reasonable methods were proposed and realized including the statistical background model, the scene separation in video sequences received from the moving camera, the fuzzy TSK model application, the detection of “good” regions in frames, which contain only camera jitters, frames boundaries restoration in static and dynamic scenes, etc. All methods and algorithms were realized by the designed software tool “DVS Analyzer”, v. 2.04.

For experiments, six video sequences received from the static camera and eight video sequences received from the moving camera were used. Graphics of the motion estimation and compensation and the stabilization quality demonstrate the improvements relative to the original video sequences. The PSNR and ITF metrics were used to estimate the received results. The stabilized results directly depend from varied foreground and background content, the moving objects, lighting, noisy, and the shooting condition for each video sequence. The ITF estimations increase up on 3–4 dB or 15–20 % relative to the original video sequences. The ITF estimations of the proposed software tool “DVS Analyzer” provides better results (at average 1–3 dB or 5–15 %) with the lower processing time relatively the existing software tools.

The future efforts are connected with the development of advanced motion inpainting methods and algorithms for the DVS task and also fast realization of algorithms without the essential accuracy reduction for pseudo real-time applications.

References

1. Battiato S, Lukac R (2008) Video stabilization techniques. In: Furht B (ed) *Encyclopedia of multimedia*. Springer Science+Business Media, New York
2. Chen H, Liang CK, Peng YC, Chang HA (2007) Integration of digital stabilizer with video codec for digital video cameras. *IEEE Trans Circuits Syst Video Technol* 17(7):801–813
3. Rawat P, Singhai J (2011) Review of motion estimation and video stabilization techniques for hand held mobile video. *Int J of Signal and Image Process* 2(2):159–168
4. Leghmizi S, Liu S (2011) A survey of fuzzy control for stabilized platforms. *Int J of Comput Sci Eng Surv (IJCSES)* 2(3):48–57
5. Battiato S, Bruna AR, Puglisi G (2010) A robust block based image/video registration approach for mobile imaging devices. *IEEE Trans Multimedia* 12(7):622–635
6. Bosco A, Bruna A, Battiato S, Bella G, Puglisi G (2008) Digital video stabilization through curve warping techniques. *J IEEE Trans Consum Electron* 54(2):220–224
7. Lee J, Park Y, Lee S, Paik J (2009) Statistical region selection for robust image stabilization using feature-histogram. In: *IEEE international conference on image processing (ICIP)*, pp 1553–1556
8. Liu F, Gleicher M, Jin H, Agarwala A (2009) Content-preserving warps for 3D video stabilization. *ACM Trans Graph (SIGGRAPH)* 28(3):44:1–44:9
9. Wang JM, Chou HP, Chen SW, Fuh CS (2009) Video stabilization for a hand-held camera based on 3D motion model. In: *IEEE international conference on image processing (ICIP)*, pp 3477–3481
10. Nestares O, Gat Y, Haussecker H, Kozinsev I (2010) Video stabilization to a global 3D frame of reference by fusing orientation sensor and image alignment data. In: *9th IEEE international symposium on mixed augmented reality (ISMAR)* 257–258
11. Hansen M, Anandan P, Dana K, van der Wal G, Burt PJ (1994) Real time scene stabilization and mosaic construction. *Image understanding Workshop, Defense Advanced Research Project Agency (DARPA)*, pp 457–465
12. Favorskaya M (2012) Motion estimation for object analysis and detection in videos. In: Kountchev R, Nakamatsu K (eds) *Advances in reasoning-based image processing, analysis and intelligent systems*. Springer, Berlin Heidelberg
13. Babagholami-Mohamadabadi B, Bagheri-Khaligh A, Hassanpour R (2012) Digital video stabilization using radon transform. *Digital Image Comput: Tech Appl (DICTA)* 12:1–8
14. Favorskaya M, Levitin K (2014) Early smoke detection in outdoor space by spatio-temporal clustering using a single video camera. In: Tweedale JW, Jain LC (eds) *Recent advances in knowledge-based paradigms and applications*. Springer, Berlin Heidelberg
15. Shakoer MH, Moattari M (2011) Statistical digital image stabilization. *J Eng Technol Res* 3(5):161–167
16. Zhou M, Asari VK (2011) A fast video stabilization system based on speeded-up robust features advances. In: Zhou M, Asari VK (eds) *Adv Vis Comput., LNCSSpringer*, Berlin Heidelberg
17. Ertürk S (2002) Real-time digital image stabilization using Kalman filters. *Real Time Image* 8(4):317–328
18. Pinto B, Anurenjan PR (2011) Video stabilization using speeded up robust features. In: *International conference on communications and signal processing (ICCSP)*, pp 527–531

19. Chang HC, Lai SH, Lu KR (2006) A robust real-time video stabilization algorithm. *J Vis Commun Image Represent* 17(3):659–673
20. Alzoubi H, Pan WD (2008) Fast and accurate global motion estimation algorithm using pixel subsampling. *Inf Sci* 178:3415–3425
21. Del Blanco CR, Jaureguizar F, Salgado L, García N (2008) Automatic feature-based stabilization of video with intentional motion through a particle filter. In: del-Blanco CR, Jaureguizar F, Salgado L, García N (eds) *Advanced concepts for intelligent vision systems (ACIVS) LNCS*. Springer, Berlin Heidelberg
22. Yang J, Schonfeld D, Mohamed M (2009) Robust video stabilization based on particle filter tracking of projected camera motion. *IEEE Trans Circ Syst Video Technol* 19(7):945–954
23. Litvin A, Konrad J, Karl WC (2003) Probabilistic video stabilization using Kalman filtering and mosaicking. *Image and video communication and processing 2003*. SPIE-IS&T Electron Imaging SPIE 5022:663–674
24. Güllü MK, Ertürk S (2004) Membership function adaptive fuzzy filter for image sequence stabilization. *IEEE Trans Consum Electron* 50(1):1–7
25. Battiato S, Puglisi G, Bruna AR (2008) A robust video stabilization system by adaptive motion vectors filtering. In: *IEEE international conference on multimedia and expo*, pp 373–376
26. Tanakian MJ, Rezaei M, Mohanna F (2012) Digital video stabilizer by adaptive fuzzy filtering. *EURASIP J Image Video Process* 2012. SpringerOpen. doi:[10.1186/1687-5281-2012-21](https://doi.org/10.1186/1687-5281-2012-21)
27. Puglisi G, Battiato S (2011) A robust image alignment algorithm for video stabilization purposes. *IEEE Trans Circ Syst Video Technol* 21(10):1390–1400
28. Kyriakoulis N, Gasteratos A (2008) A recursive fuzzy system for efficient digital image stabilization. *J Adv Fuzzy Syst* 8:1–8
29. Amanatiadis AA, Andreadis I (2010) Digital image stabilization by independent component analysis. *IEEE Trans Instrum Measur* 59(7):1755–1763
30. Shen Y, Guturu P, Damarla T, Buckles BP, Namuduri KR (2009) Video stabilization using principal component analysis and scale invariant feature transform in particle filter framework. *IEEE Trans Consum Electron* 55(3):1714–1721
31. Tsai D, Lai S (2008) Defect detection in periodically patterned surfaces using independent component analysis. *Pattern Recogn* 41(9):2812–2832
32. Kim N, Lee H, Lee J (2008) Probabilistic global motion estimation based on Laplacian two-bit plane matching for fast digital image stabilization. *EURASIP J Adv Signal Process* 1–10
33. Pun CM, Lee MC (2003) Log-polar wavelet energy signatures for rotation and scale invariant texture classification. *IEEE Trans Pattern Anal Mach Intell* 25(5):590–603
34. Hu R, Shi R, Shen I, Chen W (2007) Video stabilization using scale invariant features. In: *IEEE 11th international conference on information visualization IV'07*, pp 871–877
35. Tanakian MJ, Rezaei M, Mohanna F (2010) Digital video stabilization system by adaptive motion vector validation and filtering. In: *International conference on communication engineering*, pp 165–183
36. Tanakian MJ, Rezaei M, Mohanna F (2011) Digital video stabilization system by adaptive fuzzy filtering. In: *19th European signal process conference*, pp 318–322
37. Matsushita Y, Ofek E, Ge W, Tang X, Shum HY (2006) Full-frame video stabilization with motion inpainting. *IEEE Trans Pattern Anal Mach Intell* 28(7):1150–1163
38. Pang D, Chen H, Halawa S (2010) Efficient video stabilization with dual-tree complex wavelet transform. *EE368 project report*
39. Liu F, Gleicher M, Wang J, Jin H, Agarwala A (2011) Subspace video stabilization. *ACM Trans Graph* 30(1):4:1–4:10
40. Zhang G, Hua W, Qin X, Shao Y, Bao H (2009) Video stabilization based on a 3D perspective camera model. *Vis Comput* 25(11):997–1008
41. Chen PH, Chen HM, Hung KJ, Fang WH, Shie MC, Lai F (2006) Markov model fuzzy-reasoning based algorithm for fast block motion estimation. *J Vis Commun Image Represent* 17(1):131–142

42. Benmoussat N, Belbachir MF, Benamar B (2007) Motion estimation and compensation from noisy image sequences: a new filtering scheme. *Image Vis Comput* 25(5):686–694
43. Liu X, Cong W (2010) Hybrid-template adaptive motion estimation algorithm based on block matching. *Int Conf Comput Commun Technol Agric Eng*. doi:[10.1109/CCTAE.2010.5543459](https://doi.org/10.1109/CCTAE.2010.5543459)
44. Boudlal A, Nsirir B, Aboutajdine D (2010) Modeling of video sequences by gaussian mixture: application in motion estimation by block matching method. *EURASIP J Adv Signal Process*. doi:[10.1155/2010/210937](https://doi.org/10.1155/2010/210937)
45. Jang SW, Pomplun M, Kim GY, Choi HI (2005) Adaptive robust estimation of affine parameters from block motion vectors. *Image Vis Comput* 23(14):1250–1263
46. Basarab A, Liebgott H, Morestin F, Lyshchik A, Higashi T, Asato R, Delachartre P (2008) A method for vector displacement estimation with ultrasound images and its application for thyroid nodular disease. *Med Image Anal* 12(3):259–274
47. Moreno-Garcia J, Rodriguez-Benitez L, Fernandez-Caballero A, Lopez MT (2010) Video sequence motion tracking by fuzzification techniques. *Appl Soft Comput* 10(1):318–331
48. Sugeno M (1985) *Industrial applications of fuzzy control*. Elsevier Science Inc., New York
49. Battiato S, Gallo G, Puglisi G, Scellato S (2009) Fuzzy-based motion estimation for video stabilization using SIFT interest points. In: *IS&T-SPIE electronic imaging symposium, digital photography V EI-7250*, pp 1–8
50. Pan JY, Hu B, Zhang JQ (2008) Robust and accurate object tracking under various types of occlusions. *IEEE Trans CSVT* 18:223–236
51. O’Hara S, Lui YM, Draper BA (2012) Using a product manifold distance for unsupervised action recognition. *Image Vis Comput* 30(3):206–216
52. Belardinelli A, Carbone A, Schneider WX (2013) Classification of multi-scale spatiotemporal energy features for video segmentation and dynamic objects prioritization. *Pattern Recogn Lett* 34(7):713–722
53. Panagiotakis C, Doulamis AD, Tziritis G (2009) Equivalent key frames selection based on iso-content principles. *IEEE Trans Circuits Syst Video Technol* 19(3):447–451
54. Chao GC, Tsai YP, Jeng SK (2010) Augmented keyframe. *J Vis Commun Image Represent* 21(7):682–692
55. Cheung V, Frey BJ, Jovic N (2005) Video epitomes. *IEEE Comput Soc Conf Comput Vis Pattern Recognit (CVPR)* 1:42–49
56. Yu G, Li Z, Suyu W, Lansun S (2009) A novel scene cut detection method in H.264/AVC compression domain. *Chin J Electron* 18(4):695–699
57. Jokovic J, Horevic D (2009) Scene cut detection in video by using combination of spatial-temporal video characteristics. In: *9th international conference on telecommunication in modern satellite, cable, and broadcasting services*, pp 479–482
58. Xiang S, Meng G, Wang Y, Pan C, Zhang C (2012) Image deblurring with matrix regression and gradient evolution. *Pattern Recogn* 45(6):2164–2179
59. Cho S, Wang J, Lee S (2011) Handling outliers in non-blind image deconvolution. In: *International conference on computer vision ICCV11*, pp 495–502
60. Xu YQ, Wang L, Hu XY, Peng SL (2012) Single-image blind deblurring for non-uniform camera-shake blur. In: *ACCV12*, vol 7726, pp 336–348
61. Whyte O, Sivic J, Zisserman A, Ponce J (2010) Non-uniform deblurring for shaken images. In: *Computer vision and pattern recognition CVPR 2010*, pp 491–498
62. Cai JF, Ji H, Liu C, Shen Z (2009) Blind motion deblurring using multiple images. *J Comput Phys* 228(14):5057–5071
63. Favorskaya M, Pyankov D, Popov A (2013) Motion estimations based on invariant moments for frames interpolation in stereovision. *Proc Comput Sci* 22:1102–1111
64. Kwatra V, Essa I, Bobick A, Kwatra N (2005) Texture optimization for example-based synthesis. *ACM Trans Graph* 24(3):795–802
65. Favorskaya M, Damov M, Zotin A (2013) Accurate spatio-temporal reconstruction of missing data in dynamic scenes. *Pattern Lett Recognit* 34(14):1694–1700

66. Lim T, Han B, Han JH (2012) Modeling and segmentation of floating foreground and background in videos. *Pattern Recognit* 45(4):1696–1706
67. Lowe D (2004) Distinctive image features from scale-invariant key points. *Int J Comput Vis* 60(2):91–110
68. Zhong S, Wang J, Yan L, Kang L, Cao Z (2013) A real-time embedded architecture for SIFT. *J Syst Archit* 59(1):16–29
69. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (surf). *Comput Vis Image Underst* 110(3):346–359
70. Kerr D, Coleman SA, Scotney BW (2011) Finite element laplacian feature detector. In: IAPR, conference on machine vision and applications (MVA 2011), pp 381–384
71. Lucas B, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: 7th international joint conference on artificial intelligence (IJCAI), vol 2, pp 674–679
72. Liao B, Du M, Hu J (2010) Color optical flow estimation based on gradient fields with extended constraints. In: International conference on networking and information technology, pp 279–283
73. Lee KJ, Yun ID, Lee SU (2013) Adaptive large window correlation for optical flow estimation with discrete optimization. *Image Vis Comput* 31(9):631–639
74. Nayak NM, Zhu Y, Roy-Chowdhury AK (2013) Vector field analysis for multi-object behavior modeling. *Image Vis Comput* 31(6–7):460–472
75. Kim KS, Jang DS, Choi HI (2007) Real time face tracking with pyramidal Lucas-Kanade feature tracker. In: International conference on computational science and its applications ICCSA'07 Part I, pp 1074–1082
76. Horn BKP, Schunck BG (1981) Determining optical flow. *Artif Intell* 17:185–203
77. Briassouli A, Kompatsiaris I (2009) Robust temporal activity templates using higher order statistics. *IEEE Trans Image Process* 18(12):2756–2768
78. Kim HH, Kim YH (2010) Toward a conceptual framework of key-frame extraction and storyboard display for video summarization. *J Am Soc Inf Sci Technol* 61(5):927–939
79. Ejaz N, Baik SW (2011) Weighting low level frame difference features for key frame extraction using fuzzy comprehensive evaluation and indirect feedback relevance mechanism. *Int J Phys Sci* 6(14):3377–3388
80. Ejaz N, Tariq TB, Baik SW (2012) Adaptive key frame extraction for video summarization using an aggregation mechanism. *J Vis Commun Image Represent* 23(7):1031–1040
81. Brown RG, Hwang PYC (1992) Introduction to random signals and applied Kalman Filtering. Wiley, New York
82. Han ZJ, Ye QX, Jiao JB (2008) Online feature evaluation for object tracking using Kalman Filter. In: 19th International conference on pattern recognition ICPR 2008, pp 1–4
83. Wang J, Chen X, Gao W (2005) Online selecting discriminative tracking features using particle filter. In: IEEE international conference on computer vision and pattern recognition CVPR'2005, vol 2, pp 1037–1042
84. Han Z, Ye Q, Jiao J (2011) Combined feature evaluation for adaptive visual object tracking. *Comput Vis Image Underst* 115(1):69–80
85. Saffari A, Leistner C, Santner J, Godec M, Bischof H (2009) On-line random forests. In: IEEE 12th international conference on computer vision workshops (ICCV Workshops), pp 1393–1400
86. Grabner H, Leistner C, Bischof H (2008) Semi-supervised on-line boosting for robust tracking. In: 10th European conference on computer vision ECCV '08 Part I, pp 234–247
87. Santner J, Leistner C, Saffari A, Pock T, Bischof H (2010) PROST parallel robust online simple tracking. In: IEEE conference on computer vision and pattern recognition, pp 723–730
88. Kalal Z, Matas J, Mikolajczyk K (2010) P–N learning: bootstrapping binary classifiers by structural constraints. IEEE conference on computer vision and pattern recognition, pp 49–56
89. Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part based models. *IEEE Trans Pattern Anal Mach Intell* 32(9):1627–1645

90. Gall J, Lempitsky V (2009) Class-specific Hough forests for object detection. In: IEEE conference on computer vision and pattern recognition CVPR 2009, pp 1022–1029
91. Maji S, Malik J (2009) Object detection using a max-margin Hough transform. In: IEEE conference on computer vision and pattern recognition, CVPR 2009, pp 1038–1045
92. Godec M, Roth PM, Bischof H (2011) Hough-based tracking of non-rigid objects. In: IEEE international conference on computer vision and image understanding computer vision (ICCV 2011), pp 81–88
93. Rother C, Kolmogorov V, Blake A (2004) GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Trans Graph* 23(3):309–314
94. Ozuysal M, Calonder M, Lepetit V, Fua P (2010) Fast keypoint recognition using random ferns. *IEEE Trans Pattern Anal Mach Intell* 32(3):448–461
95. Villamizar M, Moreno-Noguer F, Andrade-Cetto J, Sanfeliu A (2010) Efficient rotation invariant object detection using boosted Random Ferns. In: IEEE conference on computer vision and pattern recognition, CVPR 2010, pp 1038–1045
96. Godec M, Leistner C, Saffari A, Bischof H (2010) On-line random naive bayes for tracking. In: International conference on pattern recognition ICPR 2010, pp 3545–3548
97. Rodriguez-Sánchez R, García JA, Fdez-Valdivia J (2013) Image inpainting with nonsubsampling contourlet transform. *Pattern Recogn Lett* 34(13):1508–1518
98. Kim S, Shina J, Paik J (2003) Real-time iterative framework of regularized image restoration and its application to video enhancement. *Real-Time Imaging* 9(1):61–72
99. Szeliski R (2006) Image Alignment and Stitching: a tutorial. *Found Trends Comput Graph Vis* 2(1):1–104
100. Eden A, Uyttendaele M, Szeliski R (2006) Seamless image stitching of scenes with large motions and exposure differences. In: IEEE computer society conference on computer vision and pattern recognition, vol 2, pp 2498–2505
101. Ma LM, Wu ZM (2009) Approximation to the k-th derivatives by multiquadric quasi-interpolation method. *J Comput Appl Math* 231(2):925–932
102. Colonnese S, Randi R, Rinauro S, Scarano G (2010) Fast image interpolation using circular harmonic functions. *European workshop on visual information processing EUVIP 2010*, pp 114–118
103. Lee Y, Yoon J (2010) Nonlinear image upsampling method based on radial basis function interpolation. *IEEE Trans Image Process* 19(10):2682–2692
104. Gao Q, Wu Z, Zhang S (2011) Applying multiquadric quasi-interpolation for boundary detection. *Comput Math Appl* 62(12):4356–4361
105. Zhang X, Wu X (2008) Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation. *IEEE Trans Image Process* 17(6):887–896
106. Takeda H, van Beek P, Milanfar P (2008) Spatio-temporal video interpolation and denoising using motion-assisted steering kernel MASK regression. In: International conference on image processing ICIP 2008, pp 637–640
107. Ceccarelli M (2007) A finite Markov random field approach to fast edge-preserving image recovery. *Image Vis Comput* 25(6):792–804
108. Li M, Nguyen TQ (2008) Markov random field model based edge-directed image interpolation. *IEEE Trans Image Process* 17(7):1121–1128
109. Nemirovsky S, Porat M (2009) On texture and image interpolation using Markov models. *Sig Process Image Commun* 24(3):139–157
110. Simonyan K, Vatolin D (2009) Edge-directed interpolation in a bayesian frame-work. In: British machine vision conference, vol 10, pp 1521–1527
111. Colonnese S, Rinauro S, Scarano G (2011) Markov random fields using complex line process: an application to Bayesian image restoration. *European workshop on visual information processing EUVIP 2011*, pp 30–35
112. Colonnese S, Rinauro S, Scarano G (2013) Bayesian image interpolation using Markov random fields driven by visually relevant image features. *Sig Process Image Commun* 28(8):967–983

Chapter 10

Implementation of Hadamard Matrices for Image Processing

Leonid Mironovsky and Valery Slaev

Abstract The image quality influences the accuracy of obtained results. In the chapter, the application of the strip-method for noise-immune storage and transmission of images is analyzed. At the same time, before transmitting the matrix transformation of an original image has to be done, when the image fragments are mixed up and superimposed each other. The transformed image is transmitted over a communication channel, where it is distorted with a pulse noise, the latter being, for example, a possible reason for a complete loss of separate image fragments. After the signal transmission to the receiving end, an inverse transformation is performed. During this transformation, the reconstruction of the image takes place. If it is possible to provide a uniform distribution of the pulse noise over the whole area, which the image occupies without any changes of its energy, then a noticeable decrease of noise amplitude will take place and an acceptable quality of all fragments of the image are reconstructed. The tasks of the chapter are the consideration the versions of the two-sided strip-transformation of images and the choice of optimal transformation matrices. A great attention has been paid to the implementation of Hadamard matrices and matrices close to them such as Hadamard-Mersenne, Hadamard-Fermat, and Hadamard-Euler matrices.

Keywords Image quality • Strip-method • Image transmission • Matrix transformations • Pulse noise • Inverse transformation • Hadamard matrix • Two-levels M-matrix • Three-levels M-matrix • Multi-levels M-matrices

L. Mironovsky (✉)

St. Petersburg State University of Aerocosmic Instrumentation, 67 Bol.Morskaya st.,
St. Petersburg 19000, Russian Federation
e-mail: miron@aanet.ru

V. Slaev

D.I. Mendeleyev Research Institute for Metrology, 19 Moskovsky av.,
St. Petersburg 190005, Russian Federation
e-mail: v.a.slaev@vniim.ru

10.1 Introduction

Many problems of information transforming and data studying are connected with the images processing and transmitting. For example, it is possible to name the remote sensing of the Earth surface by satellites, rentgenography and its application in medicine, investigations of biological and chemical processes, among others. The accuracy of the obtained results depends from the quality of images.

In this chapter, the application of the strip-method [1–3] for storage and noise-immune transmission of images is analyzed [4]. At the same time, before transmitting the matrix transformation of an original image is fulfilled, when the image fragments are mixed up and superimposed each other. The transformed image is transmitted over a communication channel, where it is distorted by pulse noises, the latter being for example a possible reason for a complete loss of separate image fragments. After the signal transmission to the receiving end, an inverse transformation is performed. During this transformation, the reconstruction of the image takes place. If it is possible to provide a uniform distribution of the pulse noise over the whole area, which the image occupies without any changes of its energy, then a noticeable decrease of noise amplitude will take place and the acceptable quality of all fragments of the reconstructed image is achieved.

Section 10.2 provides the related work. Strip-method of image transformation is developed in Sect. 10.3. The Hadamard matrices and matrices closed to them are represented in Sect. 10.4. Conclusion is situated in Sect. 10.5.

10.2 Related Work

It is reasonable that the strip-method is merely one of the methods used for increasing the accuracy of signal and image transmission over communication channels. A great number of publications are devoted to issues of raising the noise-resistance of information transmission systems [3, 5–8], and others. It is also necessary to mention some works in the adjacent fields of activities such as the works in the cluster systems of message transmission and linear predistortion of signals, which were made by Russian researches Ageyev, Babanov, Lebedev, Marigodov, Suslonov, Tsibakov, Yaroslavsky, and others; the works connecting with the method of redundant variables; the works in the linear transformation and block coding of signals and images, which were done by American researches Costas, Lang, Leith, Pierce, Upatnieks [9–17], and others. Thus, the noise control based on introduction of pre-distortions at the stage of signal transmission and on optimal processing at the stage of signal reception is widely used in information transmission systems.

However, the majority of works deal with the pre-distortion methods and the correction by using a root-mean-square criterion, whereas the methods satisfying the requirements for optimizing the information transmission systems with the help

of a minimax criterion have been developed to a significantly lesser degree. Therefore, it would be more useful to develop and study new methods for blanking the pulse interference, which are supported by using the minimax criterion and modern computer processing for images.

10.3 Strip-Method of Image Transformation

In this section, the basis of two-dimensional strip-transformation (Sect. 10.3.1) and the choice of optimal transformation matrices (Sect. 10.3.2) will be discussed.

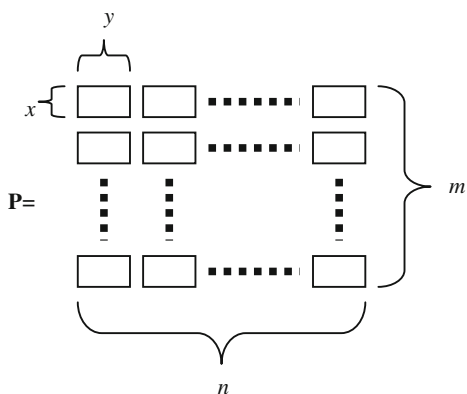
10.3.1 Two-Dimensional Strip-Transformation

The first stage of the strip-method for the transformation of one-dimensional signals consists in a “cutting” the original signal into n strips with equal duration and a forming from them the n -dimensional vector \mathbf{X} . At the second stage, this vector is fallen under the isometric transformation by its multiplying on the orthogonal matrix \mathbf{A} of the dimension $n \times n$

$$\mathbf{Y} = \mathbf{A}\mathbf{X}.$$

In the same way, the first stage of the strip-transformation of two-dimensional signals (images) consists in dividing the original image \mathbf{P} into N rectangular fragments similar in size as it is shown in Fig. 10.1. Let the number of horizontal and vertical stripes, into which the image is conditionally “cut”, be denoted as m and n ; then $N = mn$. Further, a linear combination of the fragments is made. At that, there are two approaches such as the vector and the matrix ones [18, 19].

Fig. 10.1 The image presentation as the block-matrix \mathbf{P}



According to the first (vector) approach, the obtained fragments are used to form an N -dimensional block-vector X that as in the one-dimensional case undergoes the isometric transformation by multiplying it on the orthogonal matrix \mathbf{A} of the dimension $N \times N$: $\mathbf{Y} = \mathbf{A}\mathbf{X}$. Let this version, entirely the same as in the one-dimensional case, be denoted as the one-sided strip-transformation. Its main weakness is too high dimensionality of the matrix \mathbf{A} and corresponding calculation costs.

According to the second (matrix) approach, the original image divided into fragments, is considered as a block-matrix \mathbf{X} of the dimension $m \times n$. Here three versions of isometric transformation of this matrix with the purpose to “mix” its fragments are possible [20]:

- The multiplication by the orthogonal $m \times m$ matrix \mathbf{B} on the left (the left-sided matrix transformation)

$$\mathbf{Z}_1 = \mathbf{B}\mathbf{X}.$$

- The multiplication by the orthogonal $n \times n$ matrix \mathbf{A} on the right (the right-sided matrix transformation)

$$\mathbf{Z}_2 = \mathbf{X}\mathbf{A}.$$

- The simultaneous multiplication by the matrix \mathbf{B} on the left and by the matrix \mathbf{A} on the right (the two-sided or bilateral matrix transformation)

$$\mathbf{Z}_3 = \mathbf{B}\mathbf{X}\mathbf{A}.$$

All versions listed above are shown in Fig. 10.2. It illustrates a chain of transformations of the original image \mathbf{P} , which results into an image being transmitted over the communication channel.

The first and the last versions of transformation are of the main interest, since they provide the most complete “mixing” of the image fragments. Each fragment of the transformed image contains information about all $N = mn$ fragments of the original image \mathbf{P} . In other two versions $\mathbf{Z}_1 = \mathbf{B}\mathbf{X}$ and $\mathbf{Z}_2 = \mathbf{X}\mathbf{A}$ only horizontal or only vertical stripes into which the original image has been “cut” are linearly combined.

Therefore, only two versions of transformation will be considered below:

- The one-sided strip-transformation provided by Eq. 10.1, where \mathbf{X} is the block-vector of the dimension $mn \times 1$, \mathbf{A} is the orthogonal matrix of the order mn .

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \tag{10.1}$$

- The two-sided strip-transformation by Eq. 10.2, where \mathbf{X} is the block-matrix of the dimension $m \times n$, \mathbf{B} and \mathbf{A} are the orthogonal matrices of the orders m and n .

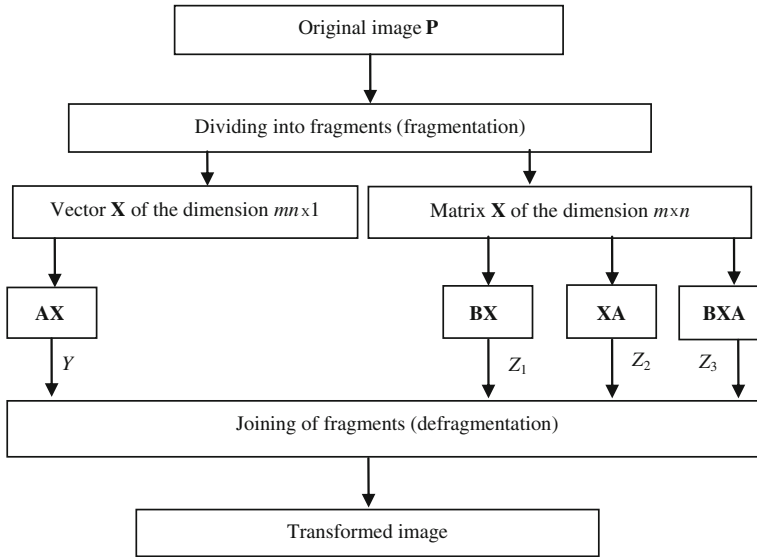


Fig. 10.2 The strip-transformation of two-dimensional signals

$$\mathbf{Z} = \mathbf{BXA} \tag{10.2}$$

Correspondingly, the inverse transformations, when the image is reconstructed at the receiving end of the communication channel, are described by Eq. 10.3 for the one-sided transformation and Eq. 10.4 for the two-sided transformation.

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{Y} \tag{10.3}$$

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{ZB}^{-1} \tag{10.4}$$

Let these two versions be described in more details.

The image transmission with using the one-sided strip-transformation. Let an original and reconstructed images be denoted as \mathbf{P} and \mathbf{P}' , and a straight and an inverse operators, realizing fragmentation and defragmentation of the image, as S_1 and $S_1' = S_1^{-1}$.

In the communication channel to the vector $\mathbf{Y} = \mathbf{AX}$, a pulse noise signal Δ is added. As a result, at the output of the channel we get an image-vector $\mathbf{Y}' = \mathbf{Y} + \Delta$. At the receiving end, the inverse one-sided strip-transformation is performed aimed at obtaining a vector \mathbf{X}' . This transformation is described by Eq. 10.5.

$$\mathbf{X}' = \mathbf{A}^{-1}\mathbf{Y}' = \mathbf{A}^{-1}(\mathbf{Y} + \Delta) = \mathbf{A}^{-1}\mathbf{Y} + \mathbf{A}^{-1}\Delta = \mathbf{X} + \mathbf{A}^{-1}\Delta \tag{10.5}$$

The obtained vector \mathbf{X}' is represented in the form of the sum of the vector \mathbf{X} and noise vector Δ , which have experienced the inverse transformation. At the last stage the vector \mathbf{X}' is transformed into the matrix $m \times n$, describing the reconstructed image \mathbf{P} with a noise $\Delta' = \mathbf{A}^{-1}\Delta$ added to it.

As it has already been shown, the main disadvantage of the one-sided strip-transformation is too large dimension of the matrix \mathbf{A} equal to $mn \times mn$ (the number of entries of this matrix is equal to the squared number of fragments, into which the image is divided). The matrices \mathbf{B} and \mathbf{A} used in the two-sided strip-transformation have significantly smaller dimensions (at $m = n$ a total number of their elements is equal to the doubled number of image fragments). This facilitates their formation and storage.

The image transmission with using of the two-sided transformation. The image $\mathbf{Z} = \mathbf{B}\mathbf{X}\mathbf{A}$ obtained as a result of the two-sided strip-transformation of the original image \mathbf{P} is transmitted through the communication channel. A pulse noise signal Δ (a block-matrix of the dimension $m \times n$) is added to the image transmitted into the channel. As a result at the output of the channel we have an image $\mathbf{Z}' = \mathbf{Z} + \Delta$. At the receiving end the image \mathbf{Z}' is affected by the inverse two-sided transformation for getting a matrix of a resulting image \mathbf{P}' . It is described by Eq. 10.6.

$$\mathbf{P}' = \mathbf{A}^{-1}\mathbf{Z}'\mathbf{B}^{-1} = \mathbf{A}^{-1}(\mathbf{Z} + \Delta)\mathbf{B}^{-1} = \mathbf{A}^{-1}\mathbf{Z}\mathbf{B}^{-1} + \mathbf{A}^{-1}\Delta\mathbf{B}^{-1} = \mathbf{P} + \mathbf{A}^{-1}\Delta\mathbf{B}^{-1} \quad (10.6)$$

In accordance with Eq. 10.6, the recipient will see the original image \mathbf{P} with the noise added to it in the channel and changed by the inverse two-sided transformation. To make this method more appropriate in practice, matrices \mathbf{A} and \mathbf{B} have the equal sizes that simplify calculations and save memory. Then Eq. 10.2 takes the form of Eq. 10.7, where \mathbf{A} is an orthonormal matrix.

$$\mathbf{Z} = \mathbf{A}\mathbf{X}\mathbf{A} \quad (10.7)$$

The Eq. 10.6 will simplify in a following manner (Eq. 10.8).

$$\mathbf{P}' = \mathbf{P} + \mathbf{A}^T\Delta\mathbf{A}^T \quad (10.8)$$

For further simplifying of the transformation, it is useful to apply a symmetrical matrix \mathbf{A} . In this case the inverse transformation will coincide with the straight one and the need to store and calculate separately the inverse matrix will disappear. The noise at the output of the system will be determined by the formula $\Delta' = \mathbf{A}\Delta\mathbf{A}$.

Mathematical formalism. In Eqs. 10.2–10.8 and other formulae of this part of the chapter, the multiplication of usual numeric matrices \mathbf{A} and \mathbf{B} by the block matrices \mathbf{X} , \mathbf{Y} , \mathbf{Z} , the elements of which are the image fragments, is performed. At the same time the following rules are used.

Fig. 10.3 Original image

Summation of blocks (fragments). Separate blocks (fragments) of image matrices are summed up by adding the corresponding block elements. This operation is similar to summation of two matrices of the same sizes.

Multiplication of a fragment by a number. The operation is performed by multiplying each fragment pixel by a number. At the same time the brightness of the fragment changes as a whole. The operation is similar to the multiplication of the matrix by a number.

Multiplication of the block matrix by the numeric matrix. Such a multiplication is performed in the same way as it is done in multiplying the numeric matrices according to the rule “a row by a column” taking into account the manner of two first operations.

Example. Loss of a unit (a single noise). Let the test image given in Fig. 10.3 be analyzed. The original image has definite boundaries and is characterized by the presence of both large objects and small details. Let a message (Fig. 10.3) be divided into $8 \times 8 = 64$ units, which are in series transmitted over a communication channel. By a single noise they imply a distortion or loss of one from 64 units. An example of such a noise is shown in Fig. 10.4.

Now let the two-sided transformation of the original image with the orthonormalized Hadamard matrix of the 8th order be done (this is done before the transmission). The image obtained without transformation is shown in Fig. 10.4, and the image obtained with the strip-transformation is shown in Fig. 10.5.

An analysis of the image shows that without strip-transformation this image loses its scenario (Fig. 10.4). When using the strip-transformation with the Hadamard matrices of the 8th order (Fig. 10.5) the image is obtained without any significant

Fig. 10.4 The image with the loss of one unit without strip-transformation



Fig. 10.5 The image obtained using the strip-transformation with the Hadamard matrix of the 8th order



distortions. Having estimated the images obtained by the subjective way, it is possible to note that the quality of the obtained image is quite acceptable.

Image scaling. In practice an image transmitted is represented in the form of a matrix consisted from separate pixels (brightness values). As a rule, the number of luminance range is taken as 256, that corresponds to eight binary digits.

After two-sided strip-transformation the image can transgress the bounds of a digit plane, and it is necessary to return it into acceptable bounds by means of dividing the image value by a definite scale parameter.

The most unfavorable case is typical for a purely white image. As a consequence of the strip-transformation with the Hadamard matrix of the n order all elements of such image become zero (black) ones apart from one element, the value of which will exceed the permissible range by n times.

It is obvious that the introduction of too great scale parameter can adversely affect the quality of the image restored. Therefore, a problem of choosing a minimal value of this scale parameter arises. It is possible to indicate the following versions of solving the problem. The simplest one of them is to use a fixed scale parameter n at the transmitting point and the coefficient $1/n$ at the receiving end. Shortcomings of this version are evident. A more flexible way may consist in introduction of an adaptive scale parameter that is specially calculated for each image and transmitted with it over a communication channel.

It is also possible to use a threshold filter (an amplitude detector) for limiting maximum values of the signal transmitted. With all this going on, it is possible to decrease the contrast of the image being received. For example, a white-black image of the “chess-board” type can turn into a black-grey one. The presence of prior statistical information concerning properties of the images transmitted can also help to solve the problem of scale operation.

The above is related to the case of white and black images. Technically such images are represented in the form of a matrix that consists from a number of pixels (brightness values). Just this matrix is subjected to fragmentation in the process of the strip-transformation. As to the color images, the situation is somewhat more complicated. One of the standard methods for presenting the color images is the application of three-layered matrix Red–Green–Blue. In this case, each of the three layers of the image matrix is exposed to the strip-transformation action.

10.3.2 Choice of Optimal Transformation Matrices

As a consequence of dividing the original image into fragments, shown in Fig. 10.1, a block-matrix containing $m \times n$ blocks is obtained. The entries of this matrix are rectangular and have dimensions $x \times y$. All fragments of this matrix are of the same dimensions. In those cases, when the number of pixels in a row or column of the original image matrix cannot be divided by m or n , giving an integer, it is necessary to add pixels from the right or from the bottom of the image. They should not distort the image or excessively contrast it.

The separation of the image into fragments permits to decrease significantly the calculation costs. The larger fragments mean the smaller dimension of the transformation matrix \mathbf{A} . The image fragment dimension should be chosen on the basis of an expected duration of the pulse noise, i.e. the linear dimensions of the distorted image segment. The best version will be the choice of fragment dimension equal to

maximal noise duration. This will allow the noise to be distributed over the image at the output of the system in the most uniform way. The chosen fragment dimension will determine the dimensions of the transformation matrix.

To attenuate the amplitude of pulse noise as much as possible, it is necessary to secure the uniform distribution of the noise over the image by applying the inverse transformation at the receiving end of the communication channel. This makes it possible to reconstruct information about distorted and “lost” fragments. Moreover, this arises a need to determine the type of a transformation matrix \mathbf{A} that will minimize the noise amplitude in the reconstructed image.

In case of the one-sided strip transformation, the level of noise Δ' in the reconstructed image is determined by Eq. 10.5. If the matrix \mathbf{A} is symmetrical and orthogonal, then Eq. 10.9 will take place.

$$\Delta' = \mathbf{A}\Delta \quad (10.9)$$

Similar equation can be derived from Eq. 10.8 for the two-sided transformation in a view of Eq. 10.10.

$$\Delta' = \mathbf{A}\Delta\mathbf{A} \quad (10.10)$$

Let us assume that the noise in the communication channel distorts only one fragment of the image (a single noise pulse). This means that only one of the block-vector components, Δ_i from Eq. 10.9 or that of the block-matrix, Δ_i from Eq. 10.10 can be non-zero.

In both cases the noise level Δ' in the signal reconstructed will be determined by a maximal entry module of the orthogonal matrix \mathbf{A} . Indeed, if in Eq. 10.9 we assume that $\Delta_1 = 1$, $\Delta_2 = \dots = \Delta_N = 0$, then $\Delta' = \mathbf{A}_1$, where \mathbf{A}_1 is the first column of the matrix \mathbf{A} . Thus, the noise amplitude Δ' will be equal to the maximal entry module of the first column of the matrix \mathbf{A} (and in the general case to the whole matrix \mathbf{A}).

In a similar manner, assuming, for example, that in Eq. 10.10 $\Delta_{11} = 1$, and the remaining components are $\Delta_{jj} = 0$, the following equations will be obtained:

$$\Delta' = \mathbf{A}_1 \cdot \mathbf{A}_1^T = [a_{1i} \cdot a_{1j}]_1^n.$$

Therefore, the maximal entry of the matrix Δ' will be equal to a_M^2 , where a_M is the maximal entry module of the first column of the matrix \mathbf{A} . At an arbitrary position of the non-zero entry in the matrix Δ , the maximal entry module a_M of the matrix \mathbf{A} will be obtained.

Since the aim set is to attenuate to the limit the noise amplitude, then in both cases it is required to search such class of orthogonal matrices, the one the maximal entry module of which is minimal. The well-known decision of this task relates to the cases n , which are divided by four. Such matrices are the normalized Hadamard matrices. The less known decision for even n , which are not divided by four, is

represented by so called Conference-matrices (*C*-matrices). They have a zero diagonal and their rest entries are equal to ± 1 .

The Hadamard matrices provide an ideally uniform distribution of a single noise pulse over the whole image area, decreasing its amplitude by n times (at $m \neq n$ by \sqrt{mn} times). The *C*-matrices providing the noise attenuation by $(n - 1)$ times are only a little inferior to them. For odd n , the general solution of the problem is unknown for the authors. As a result of long term searches, the orthogonal matrices for $n = 3, 5, 7, 9, 11$, optimal in this sense, have been found. More detailed information about these and other matrices, closed to the Hadamard matrices, are given in the Sect. 10.4.

10.4 Hadamard Matrices and Matrices Closed to Them

The strip-method bases itself on isometric transformations of signals and images with the help of orthogonal matrices. One of the main requirements for these matrices is the most complete “mixing” of fragments of an original signal or image in the case of the straight transformation, as well as a uniform distribution of a pulse noise along the time duration of a reconstructed signal, or over the area of a reconstructed image at the inverse transformation.

In respect to mathematics, this means that the orthogonal matrices with entries closed in absolute value should be used. The classical representatives of such matrices are the Hadamard matrices. Moreover, the subject to a technical problem to be solved some additional requirements such as the matrices symmetry, the cycle structure (Toeplitz or Hankel matrices) can be set up.

Below the description and specific form of matrices, completely or partially meeting, are presented. First, there are the Hadamard and *C*-matrices, which provide the utmost degree “mixing” of signal and image fragments. Unfortunately, these matrices are far from being present in all cases. Therefore, the problem to find the orthogonal matrices similar to them with respect to their characteristics arises. Among the versions worthy of notice, there are matrices based on orthogonal systems of functions (trigonometric functions and polynomials), two-level *D*-matrices (matrices containing elements of only two types, e.g. $\pm a, \pm b$) and minimax matrices (*M*-matrices). Second, a generalization of Hadamard matrices for odd n , since their maximum modulo entries are minimal as compared to all other orthogonal matrices of the odd order, is considered.

The Hadamard matrices, the shortened Hadamard matrices, and the Conference matrices are represented in Sects. 10.4.1–10.4.3, respectively. Section 10.4.4 provides the optimal orthogonal matrices of the odd order (*M*-matrices). Two-, three-, and many-levels *M*-matrices are discussed in Sect. 10.4.5.

10.4.1 Hadamard Matrices

The Hadamard matrices are widely used in the theory of coding (codes correcting errors), theory of planning multifactor experiments (orthogonal block-diagrams), and other fields of mathematics. Below the definition of these matrices and description of their main properties are given [3, 21–25].

Definition 1 A Hadamard matrix of order n is such $n \times n$ matrix \mathbf{A} with entries $+1$ or -1 , for which $\mathbf{A}\mathbf{A}^T = n\mathbf{I}$, where \mathbf{I} is the identity matrix.

It is evident that the Hadamard matrix is a non-singular matrix and its rows in pairs are orthogonal. The transposition of rows or columns and multiplication them by -1 again yield a Hadamard matrix. These operations allow any matrix to be transformed into one of a “normalized” form, when in the first column and row all elements are equal to $+1$. Dividing the Hadamard matrix by \sqrt{n} , an orthogonal matrix $\mathbf{A}_0 = \mathbf{A}/\sqrt{n}$, is obtained that meets the condition $\mathbf{A}_0\mathbf{A}_0^T = \mathbf{I}$. The simplest Hadamard matrix has the form

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

It is orthogonal: $\mathbf{A}^T\mathbf{A} = 2\mathbf{I}$ and symmetrical.

After dividing this matrix by $\sqrt{2}$ it becomes orthonormal

$$\mathbf{A}_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

It is easy to make sure that if \mathbf{M} and \mathbf{N} are the Hadamard matrices of orders m and n , respectively, then their Kronecker product, i.e. the matrix $\mathbf{M} \otimes \mathbf{N}$, is the Hadamard matrix of order $m \cdot n$. For example, if \mathbf{A} is the Hadamard matrix of the second order, then as a result of the Kronecker product $\mathbf{A} \otimes \mathbf{A}$ the Hadamard matrix of the 4th order is obtained

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

It is known that there are no Hadamard matrices of the odd order [26]. To provide the existence of Hadamard matrices of the even order $n > 2$, it is necessary to have n divisible by 4. It should be noted that thereby nothing but the required condition has been proved. From this condition it does not follow that at n divisible by 4 the Hadamard matrix has to exist. The hypothesis, according to which this condition is sufficient, also has not yet proved. In the geometry language the

question concerning the existence of the Hadamard matrix of order $n = 4k$ is equivalent to the question concerning the possibility to inscribe a regular hyper-simplex into a $(4k - 1)$ -dimensional cube.

To obtain the Hadamard matrices in practice, it is possible to use the command *hadamard* of the MATLAB packet. It allows the Hadamard matrices to be built for the cases, when n , $n/12$ or $n/20$ are powers of 2. Unfortunately, such n as 28, 36, 44, 52, 56, 60, and others, which are divisible by 4, do not refer to these cases, though for them the Hadamard matrices have been found long ago. A list of all known Hadamard matrices, which has been composed by Sloan, can be found at the site <http://neilsloane.com/hadamard/>. In the Sloan's library there are given all Hadamard matrices for $n = 28$ and at least by one matrix for all n values divisible by 4, right up to $n = 256$. They have names of the type: had.1.txt, had.2.txt, had.4.txt, had.8.txt, ..., 256.syl.txt, and are arranged in the form of text files containing arrays of signs + and -, corresponding to positive and negative entries of the Hadamard matrices. Contents of several files of such a type are given in Table 10.1.

The system of notation is clear from the first column, where both versions of a system for recording the Hadamard matrix of order 4, are shown.

Let us notice that the Hadamard matrices of order 2, 4, 8, and 12 are single (accurate to the isomorphism). At $n = 16$, there are some various Hadamard matrices. In the Sloan's library they are denoted as: had.16.0, had.16.hed, had.16.syl, had.16.twin, had.16.1, had.16.2, had.16.3, had.16.4. Three non-equivalent Hadamard matrices for $n = 20$ are denoted as had.20.pal, had.20.will, had.20.toncheviv. Further, in the library there are given 60 matrices of order 24 and 487 matrices of order 28, as well as the examples of Hadamard matrices for number to 256 inclusive for each n divisible by 4.

In the process of designing them, there were used methods proposed by Paley, Plackett-Burman, Sylvester, Tourin, and Williamson. Certain information about these methods can be found in the digest [27], the authors of which constructed the

Table 10.1 Examples of Hadamard matrices txt-files

++ ++	++++++	+ - - - - - - - - -	+++++
+ - + -	+ - + - + - + -	++ + - - - + + - +	+ - + - + - + - + - + - + -
+ + - -	++ - - + + - -	+++ + - - + + - -	++ - - + + - - + + - -
+ - - +	+ - - + + - - +	+ - - + - - + + + +	+ - + + - - + + - - + + - +
1 1 1 1	++++ - - - -	++++ + - - - + +	++++ - - + + + - - - - -
1 -1 1 -1	+ - + - - + + +	+++ + + + - - - - +	+ - + - + + + - + - - - + -
1 1 -1 -1	++ - - - + + +	++++ + + + - - - - -	++ - - + + + - - - - - + +
1 -1 -1 1	+ - - + - + + -	++++ + + + +	+ - + + + - + - - - + - + -
		+ - - - + + + -	++++ + + + - - - - - - -
		++ - - + + + -	+ - + - + - + - - - + - + -
		+ - + - - + + -	++ - - + + - - - + + - + +
		+	+ - - + + - - - + - - + + -
		+	++++ - - - - - - + + + +
		+	+ - + - - + - + - + - + - -
		+	++ - - - - + + - + + + - -
		+	+ - - + - + + - - + + - - +

Hadamard matrix of order 428. The greatest order, for which Hadamard matrix is presently known, is 668.

Not all Hadamard matrices represented in Table 10.1 are symmetrical. In Table 10.2 there are given versions of the matrices, which are symmetrical relative to the main or side diagonals and in a number of cases are more convenient for being used in the strip-method. The Hadamard matrix is named regular, if every row and every column contain the same number of “1”. Such matrices have the maximum number of “1” entries (among all possible Hadamard matrices of a given order). For example, the 1st row of Table 10.1 contains a regular Hadamard matrix of order 4.

10.4.2 Shortened Hadamard Matrices

With a permutation of rows, columns and a multiplication of them by -1 , it is possible to provide their symmetrical form with positive entries in the first row and the first column. Discarding this row and column, a shortened (reduced) matrix of order $n-1$ will be obtained. This matrix will no longer be orthogonal but becomes the circular one. All its rows are obtained with a cyclic shift of the first. This property is

Table 10.2 Symmetrical Hadamard matrices

$n = 4$	$\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$
$n = 8$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$
$n = 12$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \end{bmatrix}$

useful in processing signals with the strip-method since it provides “smoothness” of the signal transmitted [3].

Let some properties of the shortened Hadamard matrices be analyzed. At $n = 4$, taking as a basis the matrix from the first column of Table 10.1, the following Hadamard matrix of the third order will be obtained

$$\bar{A}_3 = \begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & 1 \end{bmatrix} \quad \bar{A}_3^{-1} = -\frac{1}{2} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

In the case given the inverse matrix, also one can find itself circulante.

Let us consider the eigenvalues λ_i and eigenvectors \mathbf{H}_i of the matrix \bar{A}_3

$$\lambda_1 = -1 \quad \lambda_2 = -2 \quad \lambda_3 = 2$$

$$\mathbf{H}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{H}_2 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \quad \mathbf{H}_3 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}.$$

The first vector corresponds to two-multiple noise that in filtration with the strip-method remains unchanged; however other two-multiple noises can increase.

For $n = 8$ we obtain the following shortened Hadamard matrix of the seventh order and the one inverse to it.

The eigenvalues of the matrix \bar{A}_7 have the form $-1, -2, 2$. In general case, the eigenvalues of the shortened Hadamard matrix obtained from the Hadamard matrix of order n are divided into three groups: one of them is always equal to -1 , a half of the rest ones is equal to \sqrt{n} , and another half is equal to $-\sqrt{n}$.

\bar{A}_7	\bar{A}_7^{-1}
$\begin{bmatrix} -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$	$\frac{1}{4} \begin{bmatrix} -1 & -1 & 0 & 0 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & -1 & 0 & 0 & -1 \\ -1 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}$

10.4.3 Conference Matrices

Definition 2 The name Conference-matrix (C -matrix) is given to any matrix C of order n with zero on the main diagonal and $+1$ and -1 on the rest places satisfying the condition $C^T C = (n - 1)I$.

Thus, rows (and columns) of C -matrices are orthogonal in pairs. The simplest C -matrices have the form as Eq. 10.11.

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & -1 & 1 \\ -1 & 1 & 0 & -1 \\ -1 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & -1 & -1 & 1 \\ 1 & 1 & 0 & 1 & -1 & -1 \\ 1 & -1 & 1 & 0 & 1 & -1 \\ 1 & -1 & -1 & 1 & 0 & 1 \\ 1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix} \tag{10.11}$$

The first and third of them are symmetrical, the second and fourth are skew-symmetrical. The skew-symmetric C -matrices as well as the Hadamard matrices exist only at $n = 2$ and n , divisible by 4. From the point of view of the strip-method, they in all respects are inferior to the Hadamard matrices, and therefore will not be considered below.

The symmetrical C -matrices of order n can exist only in the case, when $n - 2$ is divisible by 4, and $n - 1$ can be presented in the form of a sum of squares of two integer numbers. For example, at $n = 2, 6, 10, 14, 18$ they exist and for $n = 22$ do not, since number 21 is not presented by a sum of two squares. For $n = 26, 30$ the C -matrices exist since equalities $25 = 3^2 + 4^2, 29 = 2^2 + 5^2$ have a place. For $n = 34$, as well as for $n = 22$, a negative answer is obtained. For $n = 38, 42, 46$ the answer will also be negative.

Let us consider two problems, where we meet the C -matrices.

Conference arrangement problem. Let us suppose that n directors of some company have decided to arrange a conference by telephone in such a way as to provide any director with the possibility to speak to every one of his colleagues and the rest ones could listen to their discussion. The construction of such a conference-communications is equivalent to construction of a C -matrix.

Problem of weighing. What is the best scheme of weighing, if it needs to weigh n objects at n procedures of weighing?

The strategy of weighing is described by the C -matrix given by its entries c_{ij} :

- $c_{ij} = 1$, if in weighing i the object j is located on the left pan;
- $c_{ij} = -1$, if in weighing i the object j is on the right pan;
- $c_{ij} = 0$, if in weighing i the object j does not take part.

For n divisible by 4, the best scheme of weighing is given with the Hadamard matrix and for even n , which are not divided by 4, is provided by the symmetrical C -matrix.

The normalized matrices, the order of which differs from the Hadamard ones on 2, are of the extreme quality similar to that the Hadamard matrices possess: their entry maximal in absolute value is minimal (for the class of orthogonal matrices). Further we will denote the entry maximal in absolute value as α . The value of this entry for the C -matrices equals $\alpha = 1/\sqrt{n-1}$, i.e. it is only a little inferior to the

It is quite close to the optimal one. The value of its maximum element after normalization is equal to $\alpha = 1/\sqrt{n-2} = 0.25$ (for the matrix C_{18} , $\alpha = 1/\sqrt{n-1} = 0.2425$).

The Hadamard matrices and C -matrices are closely connected. In particular, it is possible to construct Hadamard matrices from C -matrices [28].

Suppose C is a symmetric C -matrix of order m . Then the matrix

$$A = \begin{bmatrix} C + I_m & C - I_m \\ C - I_m & -C - I_m \end{bmatrix}$$

is a Hadamard matrix of order $2m$.

This matrix is rather close to the optimal one; after normalization the value of its maximal entry is equal to $\alpha = 1/\sqrt{n-2} = 0.25$ (for the matrix C_{18} , $\alpha = 1/\sqrt{n-1} = 0.2425$). Moreover, if C is antisymmetric C -matrix, then $I + C$ is a Hadamard matrix of order m .

In the aggregate the Hadamard matrices and C -matrices give the solution of the orthogonal Procrustean problem (the problem to find orthogonal matrices with an entry minimal in absolute value) almost for all even n , with the exception of several values such as $n = 22$ and $n = 34$.

The situation for odd n is too much worse. Here only a few optimal matrices for small values of n are known. Information about them is given below.

10.4.4 Optimal Orthogonal Matrices of the Odd Order (M -Matrices)

Let us name the matrices providing a solution of the orthogonal Procrustean problem for odd n minimax, or simply M -matrices. Their main property is the minimality of the value α , i.e. the values of the entry maximal in absolute value on the class of all orthogonal matrices of a given dimension. Here it is possible to indicate three problems [29]:

Problem 1 Search of particular M -matrices for various numbers n .

Problem 2 Determination of an accurate bottom boundary α^* for the value of maximal entries of M -matrices α depending on n : $\alpha \geq \alpha^* = f(n)$.

Problem 3 Determination of the number k of entry levels in the M -matrix for different n .

Therefore, the Hadamard matrices can be called one-level since all their entries are equal in absolute value. The C -matrices are two-level, modulus of their entries is equal to 0 or 1. For an odd n , the M -matrices appear to be the k -level ones; k depending on n [30, 31].

It should be expected that the solution of all three problems set will depend on what remainder is, when the odd number n is divided by 4 (1 or 3). Correspondingly, a set of M -matrices breaks up into two subsets that differ in bottom boundaries, number of levels k and type of matrices.

Let us move to description of particular M -matrices for $n = 3, 5, 7, 9, 11$. Searching for these matrices is performed by numerical and symbolic modelling in the MATLAB and MAPLE packets with the help of specially developed software. As a result we have managed to determine an analytic type of entries of the optimal matrices M_3, M_5, M_7, M_9 , as well as to find the matrix M_{11} in the numerical form, having preliminary obtained a system of non-linear algebraic equations for determining its entries. A more detailed procedure of searching is explained below by an example of the matrix M_{11} .

For the case $n = 3$, the optimal matrix providing the solution of the orthogonal Procrustean problem, is provided by Eq. 10.12.

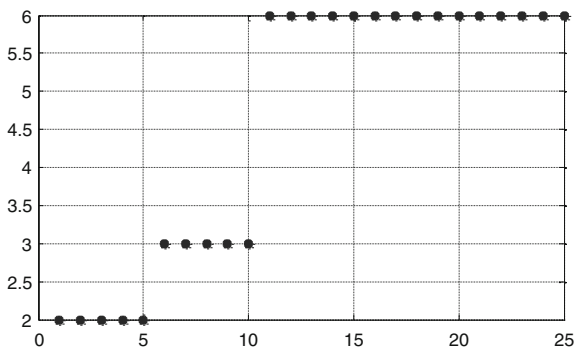
$$M_3 = \frac{1}{3} \begin{bmatrix} -1 & 2 & 2 \\ 2 & -1 & 2 \\ 2 & 2 & -1 \end{bmatrix} \tag{10.12}$$

This matrix is orthogonal and symmetrical, the value of its maximal entry is equal to $\alpha = 2/3$. The matrix contains entries of two types, i.e. it has two levels. For $n = 5$ the optimal matrix occurs to be of three levels (Eq. 10.13)

$$M_3 = \frac{1}{11} \begin{bmatrix} -2 & 3 & 6 & 6 & 6 \\ 3 & 6 & -6 & 6 & -2 \\ 6 & -6 & -3 & 2 & 6 \\ 6 & 6 & 2 & -6 & 3 \\ 6 & -2 & 6 & 3 & -6 \end{bmatrix}. \tag{10.13}$$

It is also orthogonal and symmetrical, the value of its maximal entry $\alpha = 6/11$. Distribution of the absolute value of its entries by levels is shown in Fig. 10.6.

Fig. 10.6 Distribution of the matrix M_5 entries by levels



From its 25 entries, 15 ones are on the upper level, the rest ones by 5 are on the remaining two levels. Thus, the entries of the upper levels amounts to 60 % of the total number (67 % for the matrix M_3 and 100 % for the Hadamard matrices).

In investigating the case $n = 7$, there were found two matrices: the five-level matrix M_7 of the value $\alpha = \frac{5+7\sqrt{7}}{53} \approx 0.444$ and two-level matrix N_7 of the value $\alpha = \frac{2+3\sqrt{2}}{14} \approx 0.446$. The structures of these matrices are the following:

$$\begin{aligned}
 \mathbf{M}_7 &= \begin{bmatrix} a, & -d, & c, & a, & -a, & -a, & -a \\ -d, & c, & a, & a, & a, & a, & -a \\ c, & a, & -d, & a, & -a, & a, & a \\ a, & a, & a, & -c, & b, & -b, & b \\ -a, & a, & -a, & b, & c, & -a, & -d \\ -a, & a, & a, & -b, & -a, & -d, & -e \\ -a, & -a, & a, & b, & -d, & -e, & a \end{bmatrix} \\
 \mathbf{N}_7 &= \begin{bmatrix} a, & a, & a, & a, & b, & b, & -b \\ a, & -b, & -b, & a, & -a, & b, & a \\ a, & -b, & a, & -b, & b, & -a, & a \\ a, & a, & -b, & -b, & -a, & -a, & -b \\ b, & -a, & b, & -a, & -b, & a, & -a \\ b, & b, & -a, & -a, & -b, & a, & b \\ -b, & a, & a, & -b, & -a, & b, & a \end{bmatrix}
 \end{aligned}$$

Unlike the preceding cases, the entries of these matrices are irrational.

For the matrix \mathbf{M}_7 they contain $\sqrt{7}$: $a = 3 + 3\sqrt{7}$, $b = 9$, $c = 5 - \sqrt{7}$, $d = -6 + 3\sqrt{7}$, $e = 4 + \sqrt{7}$.

In normalizing all of them should be divided by $22 + \sqrt{7}$. Entries of the matrix N_7 contain $\sqrt{2}$: $a = 2 + \sqrt{2}$, $b = 2$. In normalizing all of them should be divided by $2 + 4\sqrt{2}$. Let us show both of these matrices in detailed writing (without any normalization) (Eq. 10.14).

$$\mathbf{M}_7 = \begin{bmatrix} 3 + 3\sqrt{7}, & 6 - 3\sqrt{7}, & 5 - \sqrt{7}, & 3 + 3\sqrt{7}, & -3 - 3\sqrt{7}, & -3 - 3\sqrt{7}, & -3 - 3\sqrt{7} \\ 6 - 3\sqrt{7}, & 5 - \sqrt{7}, & 3 + 3\sqrt{7}, & 3 + 3\sqrt{7}, & 3 + 3\sqrt{7}, & 3 + 3\sqrt{7}, & -3 - 3\sqrt{7} \\ 5 - \sqrt{7}, & 3 + 3\sqrt{7}, & 6 - 3\sqrt{7}, & 3 + 3\sqrt{7}, & -3 - 3\sqrt{7}, & 3 + 3\sqrt{7}, & 3 + 3\sqrt{7} \\ 3 + 3\sqrt{7}, & 3 + 3\sqrt{7}, & 3 + 3\sqrt{7}, & -5 + \sqrt{7}, & 9, & -9, & 9 \\ -3 - 3\sqrt{7}, & 3 + 3\sqrt{7}, & -3 - 3\sqrt{7}, & -9, & 4 + \sqrt{7}, & -3 - 3\sqrt{7}, & 6 - 3\sqrt{7} \\ -3 - 3\sqrt{7}, & 3 + 3\sqrt{7}, & 3 + 3\sqrt{7}, & -9, & -3 - 3\sqrt{7}, & 6 - 3\sqrt{7}, & -4 - \sqrt{7} \\ -3 - 3\sqrt{7}, & -3 - 3\sqrt{7}, & 3 + 3\sqrt{7}, & 9, & 6 - 3\sqrt{7}, & -4 - \sqrt{7}, & 3 + 3\sqrt{7} \end{bmatrix} \tag{10.14}$$

$$N_7 = \begin{bmatrix} 2 + \sqrt{2} & 2 + \sqrt{2} & 2 + \sqrt{2} & 2 + \sqrt{2} & 2 & 2 & -2 \\ 2 + \sqrt{2} & -2 & -2 & 2 + \sqrt{2} & -2 - \sqrt{2} & 2 & 2 + \sqrt{2} \\ 2 + \sqrt{2} & -2 & 2 + \sqrt{2} & -2 & 2 & -2 - \sqrt{2} & 2 + \sqrt{2} \\ 2 + \sqrt{2} & 2 + \sqrt{2} & -2 & -2 & -2 - \sqrt{2} & -2 - \sqrt{2} & -2 \\ 2 & -2 - \sqrt{2} & 2 & -2 - \sqrt{2} & -2 & 2 + \sqrt{2} & -2 - \sqrt{2} \\ 2 & 2 & -2 - \sqrt{2} & -2 - \sqrt{2} & 2 + \sqrt{2} & 2 + \sqrt{2} & 2 \\ -2 & 2 + \sqrt{2} & 2 + \sqrt{2} & -2 & -2 - \sqrt{2} & 2 & 2 + \sqrt{2} \end{bmatrix}$$

Distribution of the entry modulus for the normalized matrix M_7 level by level, which has been obtained in MATLAB with the help of the command “plot(sort(abs(M7(:))), ‘*’)”, is shown in Fig. 10.7.

From this figure, it is seen that the bottom level contains 6 entries. The next ones contain 4, 3, and 6 entries, respectively. The most numerous upper level contains 30 entries, which amounts to about 61 % (approximately as much as in the case with the matrix M_5).

For $n = 9$ the best from found matrices has four levels and the value $\alpha = \frac{3+\sqrt{3}}{12} = 0.3943$.

$$M_9 = \begin{bmatrix} [d, & b, & b, & b, & b, & b, & b, & b, & b] \\ [b, & a, & a, & a, & -a, & -a, & -c, & -c, & -c] \\ [b, & a, & -c, & -a, & -c, & a, & a, & -c, & -a] \\ [b, & a, & -a, & -c, & a, & -c, & -a, & -c, & a] \\ [b, & -a, & -c, & a, & a, & -c, & a, & -a, & -c] \\ [b, & -a, & a, & -c, & -c, & a, & -c, & -a, & a] \\ [b, & -c, & a, & -a, & a, & -c, & -c, & a, & -a] \\ [b, & -c, & -c, & -c, & -a, & -a, & a, & a, & a] \\ [b, & -c, & -a, & a, & -c, & a, & -a, & a, & -c] \end{bmatrix}$$

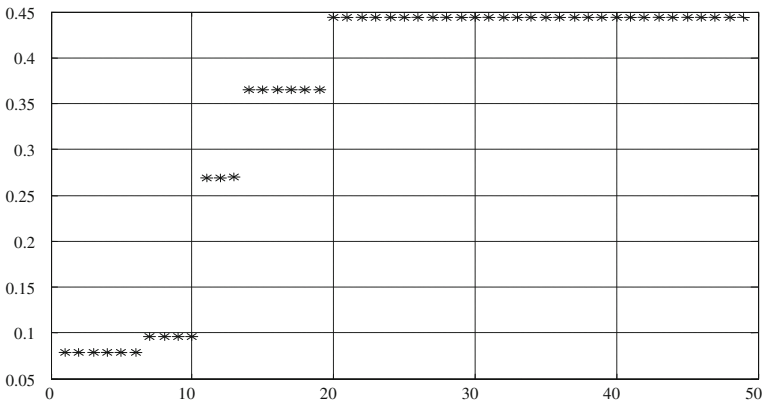
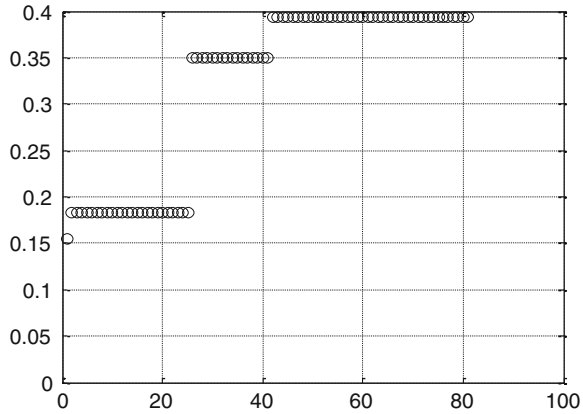


Fig. 10.7 Distribution of the matrix M_7 entries by levels

Fig. 10.8 Distribution of the matrix M_9 entries by levels



$$\begin{aligned}
 12a &= 3 + \sqrt{3}, & a &= 0.3943, \\
 6b &= \sqrt{6\sqrt{3} - 6}, & b &= 0.3493, \\
 4c &= \sqrt{3} - 1, & c &= 0.1830, \\
 3d &= 2\sqrt{3} - 3, & d &= 0.1547,
 \end{aligned}$$

$$\text{Maximal entry } \frac{3 + \sqrt{3}}{12} = 0.394337.$$

Its structure and entries are the following. Here we deal with an irrationality of the type “a root from a root” arising from the solution of a biquadratic equation. Distribution of modulus of the matrix M_9 entries on levels is shown in Fig. 10.8.

From Fig. 10.8, it is seen that on the bottom level there is one entry, on the next two levels there are 34 and 16 entries, respectively. On the upper level, there are 40 entries, which amounts to 49 % of their total number. Unfortunately, $n = 9$ is the final case, when it has been managed to get explicit expressions for entries of the M -matrix.

For $n = 11$ the best orthogonal matrix founded in MATLAB, has a six-level structure

$$M_{11} = \begin{bmatrix}
 -b & a & f & a & a & d & c & e & a & -a & -a \\
 -d & f & a & -a & e & -a & b & c & -a & -a & a \\
 -a & -e & -c & a & d & -a & a & -a & f & a & b \\
 a & -d & a & b & a & a & -f & -a & -e & -c & a \\
 a & a & e & a & -b & -a & a & -d & -a & -f & -c \\
 a & -a & a & -d & a & -e & a & f & c & b & -a \\
 -f & b & d & -c & -a & a & a & -a & a & e & a \\
 e & a & a & a & f & -c & -a & a & b & d & a \\
 a & a & -a & -f & c & a & d & b & -a & a & e \\
 a & -c & -b & e & -a & -f & a & a & a & -a & d \\
 -c & -a & a & a & -a & b & e & a & -d & a & -f
 \end{bmatrix}.$$

The numerical values of its entries are as follows: $a = 0.34295283$, $b = 0.33572291$, $c = 0.30893818$, $d = 0.2439851$, $e = 0.15671878$, $f = 0.045364966$. Their distribution over the levels is shown in Fig. 10.9.

The index $\alpha = 0.3429$ is equal to the value of the entry a . Let us notice that a percentage of entries maximal in absolute value amounts to $6/11 \approx 54.5\%$, which accurately coincides with the value of the index α for the matrix M_5 .

Equally with the search of optimal matrices of the odd order, there is a similar task with regard to those matrices of the even order, for which there are no C -matrices. First of all this refers to the orders $n = 22$, $n = 34$, and $n = 66$. Let us give the best result obtained for $n = 22$. The two level matrix M_{22} has the following form:

$$M_{22} = \begin{pmatrix} 0 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 0 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 0 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 & 1 & -1 \\ -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 0 \\ 0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 \\ -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & 0 \end{pmatrix}$$

The distribution of modules of its elements over the levels is shown in Fig. 10.10.

The index of this matrix $\alpha = 0.2236$, which is worse than estimate 0.2182 for the non-existent C -matrix only by 0.0054. This index is a little worse, i.e. only by 0.0033, than the index $\alpha = 0.2269$ for six level matrix M_{22} , obtained in [32, 33]. Similar two level matrices exist and for cases $n = 34$ and $n = 66$.

10.4.5 Two-, Three-, and Many-Levels M -Matrices

The Hadamard matrices have many remarkable properties marking them out on a set of orthogonal matrices. Unfortunately, at $n > 2$ there exist no Hadamard matrices, if n is odd or becomes odd after dividing by 2. In such cases there is a

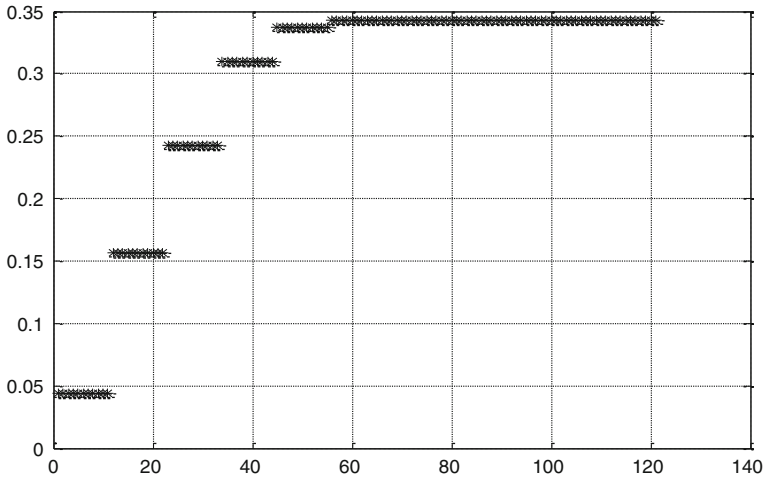
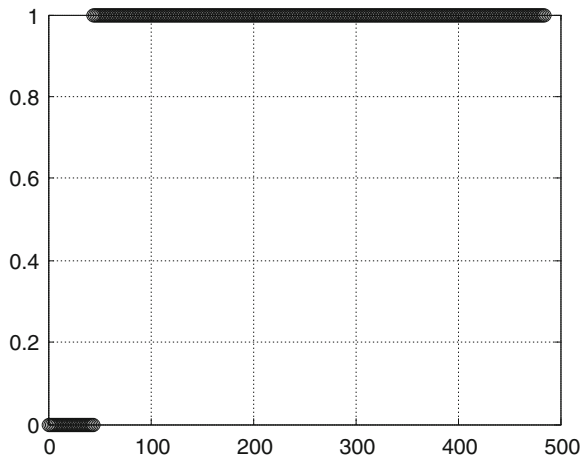


Fig. 10.9 Distribution of the matrix M_{11} entries by levels

Fig. 10.10 Distribution of the matrix M_{22} entries by levels



problem of searching some orthogonal matrices that due to their properties are close to Hadamard matrices. On mathematical statement of this problem it is necessary to indicate, what properties of Hadamard matrices should be saved in particular. According to the author's opinion three versions of setting the problem are the most natural ones:

1. An orthogonal matrix of a given order n , for which the highest possible (maximal) absolute value element is minimal (the minimax problem) has to be found.
2. An orthogonal matrix of a given order n , for which the minimum absolute value element is maximal (the maximin problem) has to be found.

3. An orthogonal matrix of a given order n , for which the difference between the maximal module and minimal module elements is minimal (the problem concerning the matrices with a minimal swing of elements).

In all versions, the cases connected with logical design of arraying indicated matrices, finding specific matrices with properties given, and analyzing the asymptote at a large n originate. It should be noted that from the theoretical point of view all three problems are equally substantial. At the same time the first problem seems to be of the greatest practical interest since it well agrees with the standard criterion signal-to-noise ratio traditionally used in the communication theory. It is interesting that in all three cases the optimal orthogonal matrices have a property, due to which their elements are able to group by a value, i.e., their elements are divided into a small number of groups (levels) with equal absolute values. At the same time, for optimal minimax matrices (problem 1) it is typical that the group of elements, which are maximal by their absolute values, is the most numerous one. On the contrary, for optimal maximin matrices (problem 2) the group of elements that are minimal by their absolute values is the most numerous one. In problem 3, it is possible to expect a more symmetrical pattern of the level-wise element distribution.

Let the orthogonal matrices be called the r -level ones, if the absolute values of their elements possess precisely r values. For example, the Hadamard matrices are single-level ones, the unitary matrix and permutation matrices are two-level ones.

From the point of view of tasks of processing images and signals, encoding, masking, constructing noise combating codes, the integer-valued two-level orthogonal matrices and ones obtained from them by the way of multiplication by a constant, are of a particular interest.

It is possible to outline a number of classes of such matrices.

The C-matrices. Such matrices are orthogonal with elements ± 1 and zero main diagonal.

The D-matrices. Matrices of such a type are orthogonal of the following form:

$$\mathbf{D}_n = \frac{1}{n} \begin{bmatrix} 2-n & 2 & 2 & \cdots & 2 \\ 2 & 2-n & 2 & \cdots & 2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 2 & 2 & 2 & \cdots & 2-n \end{bmatrix}.$$

In particular, at $n = 3, 4, 5$ they look like as mentioned below:

$$\mathbf{D}_3 = \frac{1}{3} \begin{bmatrix} -1 & 2 & 2 \\ 2 & -1 & 2 \\ 2 & 2 & -1 \end{bmatrix} = \mathbf{M}_3, \quad \mathbf{D}_4 = \frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix},$$

$$\mathbf{D}_5 = \frac{1}{5} \begin{bmatrix} -3 & 2 & 2 & 2 & 2 \\ 2 & -3 & 2 & 2 & 2 \\ 2 & 2 & -3 & 2 & 2 \\ 2 & 2 & 2 & -3 & 2 \\ 2 & 2 & 2 & 2 & -3 \end{bmatrix}.$$

Let us note that the matrix \mathbf{D}_3 coincides with the optimal matrix \mathbf{M}_3 , and the matrix \mathbf{D}_4 with the Hadamard matrix \mathbf{A}_4 .

At $n = 6, 8$ the D -matrices have a view:

$$\mathbf{D}_6 = \frac{1}{3} \begin{bmatrix} -2 & 1 & 1 & 1 & 1 & 1 \\ 1 & -2 & 1 & 1 & 1 & 1 \\ 1 & 1 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & 1 & 1 \\ 1 & 1 & 1 & 1 & -2 & 1 \\ 1 & 1 & 1 & 1 & 1 & -2 \end{bmatrix}$$

$$\mathbf{D}_8 = \frac{1}{4} \begin{bmatrix} -3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 & 1 & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & 1 & 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & -3 \end{bmatrix}.$$

The using of the D -matrices of higher orders is not efficient, since by a value of the maximum element $a = \frac{2}{n} - 1$ they significantly worse than the optimal M -matrices. At the same time the above indicated D -matrices can be used as “building blocks” for constructing other two-level and three-level matrices.

Let us note that the matrix \mathbf{D}_4 is a particular case of the family of two-level matrices having the form

$$\begin{bmatrix} a & -b & -b & b \\ b & a & b & b \\ b & -b & a & -b \\ -b & -b & b & a \end{bmatrix},$$

which are orthogonal at any a, b . Particularly the versions $[a, b] = [1 \ 2], [1 \ 3], [2 \ 3]$ are possible.

The composed orthogonal matrices. One of the methods applied for constructing the two-level orthogonal matrices is based on application of the Kronecker product of single-level and two-level matrices. For example, when analyzing the Kronecker product of the matrix $\begin{bmatrix} a & b \\ b & -a \end{bmatrix}$ and Hadamard matrix $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, one get a family of two-level matrices of the 4th order.

At $n = 6$, multiplying the matrix \mathbf{D}_3 on the Hadamard matrix $\mathbf{A}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and on unitary matrix $\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, one get two two-level matrices of the 6th order:

$$\mathbf{D}_3 \otimes \mathbf{A}_2 = \begin{bmatrix} \mathbf{D}_3 & \mathbf{D}_3 \\ \mathbf{D}_3 & -\mathbf{D}_3 \end{bmatrix} = \begin{bmatrix} -1 & 2 & 2 & -1 & 2 & 2 \\ 2 & -1 & 2 & 2 & -1 & 2 \\ 2 & 2 & -1 & 2 & 2 & -1 \\ -1 & 2 & 2 & 1 & -2 & -2 \\ 2 & -1 & 2 & -2 & 1 & -2 \\ 2 & 2 & -1 & -2 & -2 & 1 \end{bmatrix},$$

$$\alpha = \sqrt{3}/2 = 0.4714,$$

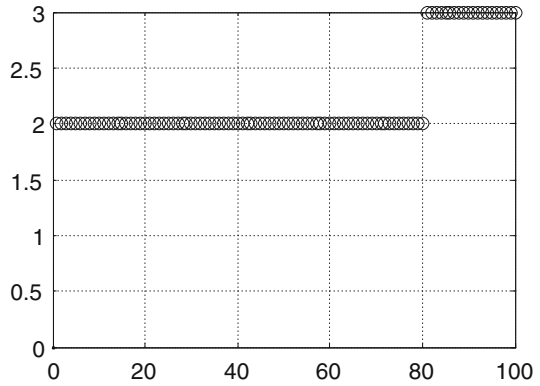
$$\mathbf{D}_3 \otimes \mathbf{I} = \begin{bmatrix} \mathbf{D}_3 & 0 \\ 0 & \mathbf{D}_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 & 0 & 0 & 0 \\ 2 & 1 & -2 & 0 & 0 & 0 \\ -2 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & -2 \\ 0 & 0 & 0 & 2 & 1 & 2 \\ 0 & 0 & 0 & -2 & 2 & 1 \end{bmatrix},$$

$$\alpha = 2/3 = 0.6667.$$

Let us note that at the Kronecker multiplication of orthogonal matrices $\mathbf{A} = \mathbf{A}_1 \otimes \mathbf{A}_2$ their indices are multiplied: $\alpha = \alpha_1 \alpha_2$. The unitary matrix \mathbf{I} has the index $\alpha = 1$, therefore at $\mathbf{A}_2 = \mathbf{I}$ we get $\alpha = \alpha_1$. Provided \mathbf{A}_2 is the Hadamard matrix then the result index is equal to $\alpha = \alpha_1/\sqrt{n}$.

The Kronecker product of the matrix D_5 and Hadamard matrix $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ gives the two-level matrix of the 10th order with the index $\alpha = 0.3\sqrt{2} = 0.4242$ (the matrix C_{10} has the similar index $\alpha = 1/3$).

Fig. 10.11 Distribution of the matrix C_{10} entries by levels



$C_{10} =$

2	-3	2	-2	-2	2	2	2	3	-2
2	2	-2	3	2	2	2	-2	2	-3
2	2	-2	-2	2	2	-3	3	2	2
-2	-2	-2	2	2	-2	3	3	2	2
3	-2	3	2	2	-2	-2	-2	2	2
-2	-2	2	2	3	3	-2	2	-2	-2
3	-2	-3	2	-2	-2	-2	2	-2	-2
-2	-2	-2	2	-3	3	-2	-2	2	2
2	2	2	3	-2	2	2	2	-2	3
-2	3	2	2	-2	-2	-2	2	3	-2.

The distribution of the matrix C_{10} entries over the levels is shown in Fig. 10.11 ($n = 10, 2$ levels). Each of its line has 8 twins and 2 triplets.

Let us consider some other examples of two-level matrices. At $n = 13$, there is a matrix consisting of zeros and units, which has the index $\alpha = 1/3$ and contains 4 zeros in each of its lines

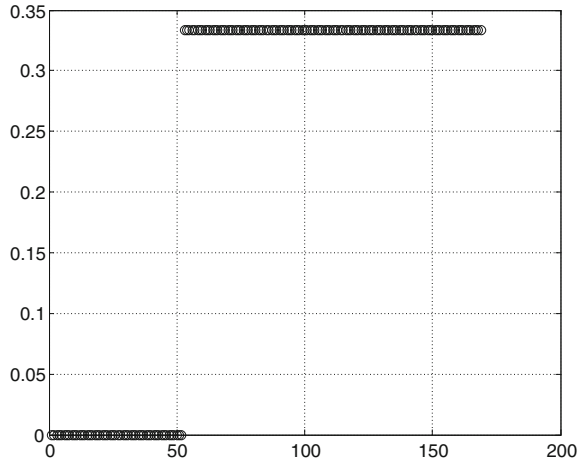
$$\begin{aligned}
 \mathbf{C}_{13} = & \\
 & \begin{matrix}
 0 & -1 & 1 & 1 & -1 & 0 & -1 & -1 & -1 & -1 & 0 & -1 & 0 \\
 -1 & -1 & 0 & 0 & 1 & -1 & 0 & -1 & -1 & 1 & 1 & 1 & 0 \\
 -1 & 0 & 1 & -1 & -1 & 1 & -1 & 0 & 1 & 0 & 1 & 1 & 0 \\
 0 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & 1 & -1 & 1 & -1 & 1 \\
 0 & -1 & -1 & 0 & -1 & 1 & 1 & -1 & 0 & 0 & -1 & 1 & 1 \\
 1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & 1 \\
 -1 & 0 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 0 & 1 \\
 -1 & 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & -1 & -1 \\
 -1 & -1 & 1 & -1 & 0 & -1 & 1 & 1 & 0 & -1 & -1 & 0 & 0 \\
 0 & 1 & 0 & -1 & -1 & -1 & -1 & 0 & -1 & 1 & -1 & 0 & 1 \\
 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 1 & 1 & 1 \\
 -1 & 1 & 1 & 1 & 1 & 0 & 0 & -1 & 1 & 0 & -1 & 0 & 1 \\
 1 & 0 & 0 & -1 & 1 & 0 & -1 & -1 & 0 & -1 & -1 & 1 & -1.
 \end{matrix}
 \end{aligned}$$

The distribution of the matrix \mathbf{C}_{13} entries over the levels is shown in Fig. 10.12 ($n = 13, 2$ levels, $\alpha = 1/3$).

At $n = 15$ there is a matrix, each line of which contains 7 twins and 8 triplets.

$$\begin{aligned}
 \mathbf{C}_{15} = & \\
 & \begin{matrix}
 3 & 2 & 2 & -3 & 3 & -3 & -2 & 2 & 2 & -2 & 3 & -3 & -3 & -3 & 2 \\
 -2 & -3 & -3 & -3 & 3 & 2 & 3 & 2 & 2 & 3 & -2 & -3 & -3 & 2 & 2 \\
 2 & -2 & 3 & -2 & -3 & -2 & -3 & 3 & -2 & 2 & -3 & -2 & 3 & 3 & 3 \\
 2 & 3 & 3 & 3 & 2 & -2 & 2 & -2 & 3 & 2 & -3 & 3 & -2 & 3 & 3 \\
 2 & -2 & 3 & -2 & 2 & 3 & 2 & 3 & 3 & -3 & 2 & 3 & 3 & 3 & -2 \\
 3 & 2 & 2 & -3 & -2 & 2 & 3 & -3 & 2 & 3 & 2 & -3 & 2 & -3 & -3 \\
 -3 & 3 & -2 & -2 & 2 & -2 & -3 & -2 & 3 & -3 & -3 & -2 & 3 & 3 & -2 \\
 2 & 3 & -2 & 3 & 2 & 3 & 2 & 3 & -2 & -3 & -3 & -2 & 3 & -2 & 3 \\
 3 & -3 & -3 & 2 & 3 & -3 & -2 & 2 & 2 & 3 & -2 & 2 & 2 & -3 & -3 \\
 -3 & 3 & 3 & -2 & 2 & 3 & -3 & 3 & -2 & 2 & -3 & 3 & -2 & -2 & -2 \\
 3 & 2 & -3 & -3 & -2 & -3 & 3 & 2 & -3 & -2 & -2 & 2 & -3 & 2 & -3 \\
 -3 & 3 & -2 & -2 & -3 & -2 & 2 & 3 & 3 & 2 & 2 & 3 & 3 & -2 & 3 \\
 3 & 2 & -3 & -3 & 3 & 2 & -2 & -3 & -3 & 3 & 3 & 2 & 2 & 2 & 2 \\
 -2 & 2 & 2 & 2 & 3 & -3 & 3 & 2 & -3 & 3 & 3 & -3 & 2 & 2 & -3 \\
 2 & 3 & -2 & 3 & -3 & 3 & -3 & 3 & 3 & 2 & 2 & -2 & -2 & 3 & -2.
 \end{matrix}
 \end{aligned}$$

Fig. 10.12 Distribution of the matrix C_{13} entries by levels



The distribution of the matrix C_{15} entries over the levels is shown in Fig. 10.13 ($n = 15, 2$ levels, $\alpha = 0.3$).

Moreover, at $n = 15$ there is the matrix C_{15} (the variant) that contains 7 twins and 8 units in each of its lines:

$$C_{15} = \begin{matrix} 2 & 2 & 2 & -2 & 1 & 2 & -1 & 1 & -1 & -1 & -1 & -1 & 2 & -1 & 2 \\ 1 & 2 & -1 & -2 & -2 & 2 & 2 & 1 & -1 & 2 & -1 & 2 & -1 & -1 & -1 \\ 1 & -1 & 2 & 1 & -2 & 2 & -1 & -2 & 2 & -1 & -1 & 2 & 2 & -1 & -1 \\ -1 & -2 & 1 & -1 & -1 & 1 & 1 & -1 & -2 & -2 & -2 & -2 & -2 & 1 & -2 \\ 1 & -1 & 2 & -2 & 1 & -1 & 2 & -2 & 2 & 2 & -1 & -1 & -1 & -1 & 2 \\ -2 & 2 & 2 & 1 & -2 & -1 & 2 & 1 & 2 & -1 & 2 & -1 & -1 & -1 & -1 \\ -2 & -1 & -1 & 1 & 1 & 2 & 2 & -2 & -1 & 2 & 2 & -1 & 2 & -1 & -1 \\ 2 & 1 & 1 & 2 & 2 & 1 & 1 & 2 & 1 & 1 & -2 & -2 & 1 & 1 & -2 \\ -1 & -2 & 1 & -1 & 2 & 1 & -2 & 2 & 1 & 1 & 1 & 1 & -2 & -2 & -2 \\ -2 & -1 & -1 & -2 & 1 & -1 & 2 & 1 & 2 & -1 & -1 & 2 & 2 & 2 & -1 \\ 2 & -2 & -2 & -1 & -1 & 1 & 1 & 2 & 1 & -2 & 1 & -2 & 1 & -2 & 1 \\ -1 & 1 & -2 & -1 & -1 & -2 & -2 & -1 & 1 & 1 & -2 & -2 & 1 & -2 & -2 \\ -1 & -2 & 1 & 2 & -1 & -2 & 1 & 2 & -2 & 1 & -2 & 1 & 1 & -2 & 1 \\ -2 & -1 & -1 & 1 & -2 & 2 & -1 & 1 & 2 & 2 & -1 & -1 & -1 & 2 & 2 \\ 1 & -1 & 2 & -2 & -2 & -1 & -1 & 1 & -1 & 2 & 2 & -1 & 2 & 2 & -1 \end{matrix}$$

The distribution of the matrix C_{15} (the variant) entries over the levels as it is shown in Fig. 10.14 ($n = 15, 2$ levels, $\alpha = 1/3$).

Similar matrices with elements consisting of two adjacent integer numbers exist at $n = 2^{2k} - 1$, i.e. at $n = 3, 15, 63, 255, \dots$. At a given k there exist two matrices, one

Fig. 10.13 Distribution of the matrix C_{15} entries by levels

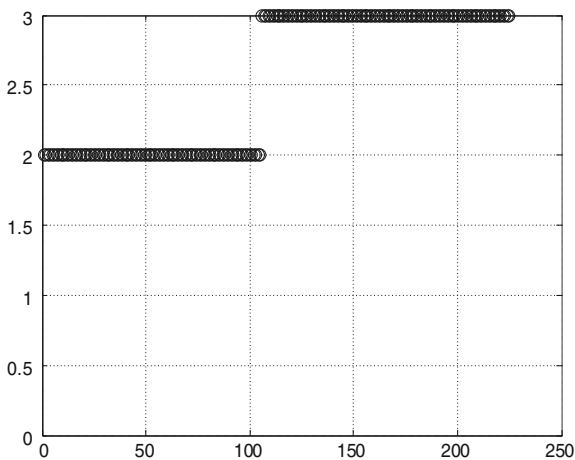
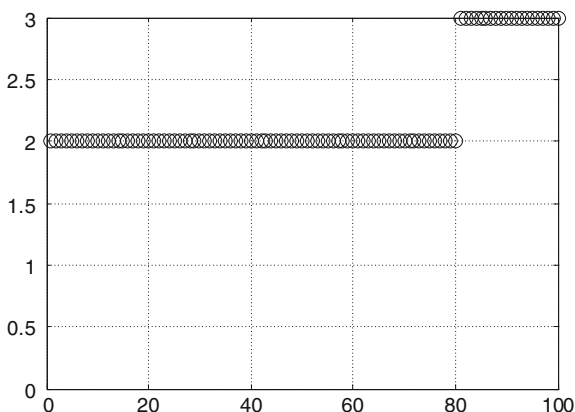


Fig. 10.14 Distribution of the matrix C_{15} (the variant) entries by levels

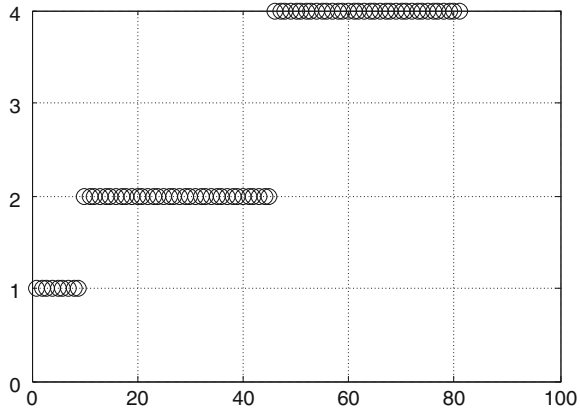


with elements $k - 1$ and k , the other with elements k and $k + 1$. In particular, at $n = 63$ ($k = 3$) such matrices will have elements which have modules equal to 3 and 4; 4 and 5.

With the help of the Kronecker product it is possible to construct three-level matrices too. When performing the multiplication, it is possible to use the following orthogonal matrices as the basic ones:

$$\mathbf{A}_2 = \begin{bmatrix} a & b \\ b & -a \end{bmatrix} \quad \mathbf{A}_3 = \begin{bmatrix} a & b & c \\ -c & a & -b \\ -b & c & a \end{bmatrix} \quad \mathbf{A}_4 = \begin{bmatrix} a & -b & -c & d \\ b & a & d & c \\ c & -d & a & -b \\ -d & -c & b & a \end{bmatrix} .$$

Fig. 10.15 Distribution of the Hadamard-Mersenne matrix M_9 entries by levels



At the same time a part of elements in these matrices can be made equal to each other. Below the three-level matrix of the 9th order is shown, which is based on the optimal matrix M_3 , the distribution of modules of its elements is illustrated too.

The distribution of the Hadamard-Mersenne matrix M_9 entries over the levels is shown in Fig. 10.15 ($n = 9, \alpha = 4/9$).

The integer-valued three-level matrix of the 25th order with a unit diagonal can be obtained with the help of cyclic shift of line:

$$1 \ -4 \ -4 \ 6 \ -4 \ 6 \ 6 \ 6 \ 6 \ 6 \ 6 \ 6 \ 6 \ -4 \ -4 \ 6 \ -4 \ -4 \ 6 \ -4 \ 6 \ -4 \ 6 \ -4 \ -4 \ -4.$$

The Hadamard-Mersenne matrices. The minimax matrices of orthogonal bases (i.e. M -matrices) with a minimal number of levels depending on a division remainder r by 4, can be divided into 4 cases:

- The matrices with $r = 0$: Hadamard matrices (H) [3, 21, 29, 34], containing matrices of the Sylvester chain.
- The matrices with $r = 1$: Hadamard-Fermat matrices (F) [35], including orders from the Fermat chain.
- The matrices with $r = 2$: Hadamard-Euler matrices (E) [36] (and C -matrices [28], with exceptions based on Euler criterion).
- The matrices with $r = 3$: Hadamard-Mersenne matrices (M) [37], including orders from a chain of Mersenne numbers.

The Hadamard-Mersenne matrices represent a class of two-level matrices of the odd order, which are close to the Hadamard matrices. The dimensionality of these matrices is equal to Mersenne numbers $2^k - 1$, and their elements tend to values $\{1, -1\}$, as the values of a integer-valued argument k increases, as it takes place with the Hadamard matrices [38].

The classical method of constructing Hadamard matrices of the order $n = 2$ is based on using the iteration formula, where the iteration process begins from the matrix $\mathbf{A}_1 = 1$

$$\mathbf{A}_{2n} = \begin{bmatrix} \mathbf{A}_n & \mathbf{A}_n \\ \mathbf{A}_n & -\mathbf{A}_n \end{bmatrix}.$$

On the analogy, at the start of constructing the Hadamard-Mersenne matrices we will use a modified Eq. 10.15, where the M_n -matrix contains elements of the form $\pm a$ and $\pm b$ (without any limitation of the commonness, let us consider that $a = 1$), and the matrix M_n^* was formed with the help of permutation of the levels a and $-b$.

$$\mathbf{M}_n^* = \mathbf{S}_{2n} = \begin{bmatrix} \mathbf{H}_n & \mathbf{H}_n \\ \mathbf{H}_n & \mathbf{H}_n^* \end{bmatrix} \tag{10.15}$$

The matrix S_{2n} obtained with this formula is symmetrical; its order is even and less than the order of the next Hadamard-Mersenne matrix M_{2n+1} by 1. At the second step the matrix S_{2n} is “bordered” by the way of adding a line and column, where λ and e are the proper number and eigenvector of the matrix S_{2n}

$$\mathbf{M}_{2n+1} = \mathbf{H}_{2n+1} = \begin{bmatrix} -\lambda & e' \\ e & \mathbf{S}_{2n} \end{bmatrix}.$$

If the iteration process is started from

$$\mathbf{H}_3 = \begin{bmatrix} a & -b & a \\ -b & a & a \\ a & a & -b \end{bmatrix},$$

then the matrix obtained by such a manner will be symmetrical and orthogonal.

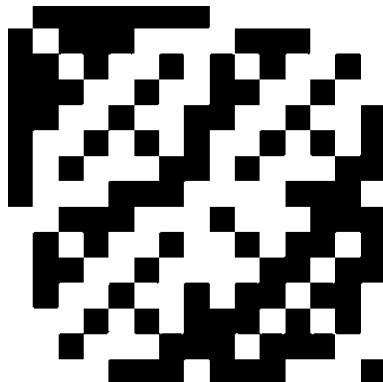
The eigenvalue of the matrix \mathbf{S}_{2n} will be equal to $\lambda = -a$. At the same time half of the components of the eigenvector consists of $-b$, and the remaining half consists of a . It takes place for next values of generating pair: $b = a/2$ at $n = 3$ and $b = \frac{p \pm \sqrt{4p}}{p-4} a$, $p = n + 1$ on the contrary.

The structure of the Hadamard-Mersenne matrix of the 15th order is shown in Fig. 10.16, where the white fields are the matrix elements with the value $a = 1$, and the black fields are the elements of the matrix with the value b .

Hadamard-Fermat matrices. These matrices represent a class of three-level matrices of the odd order, close by their properties to the properties of the Hadamard matrices. The size of these matrices is equal to numbers $2^{2k} + 1$ and as the values of integer-valued argument k increase, their elements tend to values $\{1, -1\}$, as it takes place with the Hadamard matrices.

Let the Hadamard-Fermat matrix \mathbf{F}_n be of the order n and designate by \mathbf{S}_{n-1} a symmetrical matrix obtained from the matrix \mathbf{F}_n by means of deletion of its first line

Fig. 10.16 The structure of the Hadamard-Mersenne matrix M_{15}



and column. After that, let Eq. 10.15 be modified by replacing it with quadruplicating its order according to rule [39] in Eq. 10.16, where the matrix S_{n-1}^* was formed by replacing the values of levels a with $-b$ and vice versa.

$$S_{4n-4} = \begin{pmatrix} S_{n-1}^* & S_{n-1} & S_{n-1} & S_{n-1} \\ S_{n-1} & S_{n-1}^* & S_{n-1} & S_{n-1} \\ S_{n-1} & S_{n-1} & S_{n-1}^* & S_{n-1} \\ S_{n-1} & S_{n-1} & S_{n-1} & S_{n-1}^* \end{pmatrix} \tag{10.16}$$

The matrix S_{4n-4} obtained according to Eq. 10.16 is symmetrical. Its order is even and less than the order of the next Hadamard-Fermat matrix F_{4n-3} by a unit. To complete the recursive process an additional bordering (addition of a line and column) is needed. The most important requirement is the orthogonality of the matrix obtained due to edging.

To find the location of the orthogonal edging, let the method based on the properties of proper numbers and eigenvectors of block matrices be applied. Matrix F_{4n-3} is formed by edging the matrix S_{4n-4} (Eq. 10.16) in such a way, where λ and e are the proper number and eigenvector of the matrix S_{4n-4} , respectively,

$$F_{4n-3} = \begin{pmatrix} -\lambda & e' \\ e & S_{4n-4} \end{pmatrix}. \tag{10.17}$$

The matrix obtained in such a way will be symmetrical and orthogonal, if the iteration process is started from the matrix S_{4n-4}

$$F_5 = \begin{pmatrix} a & s & s & s & s \\ s & a & -b & -b & -b \\ s & -b & a & -b & -b \\ s & -b & -b & a & -b \\ s & -b & -b & -b & a \end{pmatrix}.$$

Fig. 10.17 The structure of the Hadamard-Fermat matrix F_5

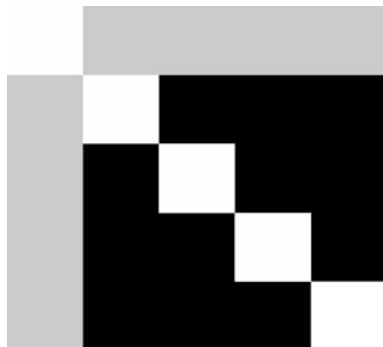


Fig. 10.18 The structure of the Hadamard-Fermat matrix F_{17}



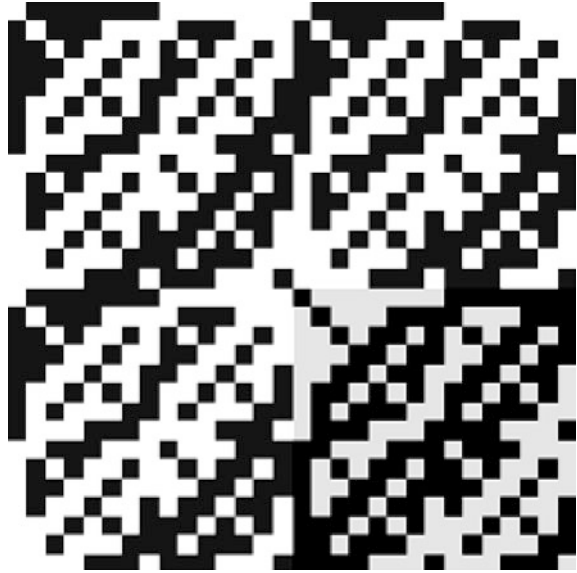
The matrix S_4 is obtained by deleting its first line and first column (“edging”). Here $a = -\lambda$ is the proper number of the matrix S_4 , taken with an inverse sign; s are the elements of the corresponding eigenvector, at that $b < s < a$.

At $n = 5$, in particular, we have $b = s = 2a/3$, where in the general case $b = \frac{n-q}{q}a$, $s = \frac{\sqrt{np-2\sqrt{p}}}{2q}a$, $q = \frac{p+\sqrt{p}}{2}$, $p = n - 1$.

The structure of the matrix F_5 and matrix F_{17} constructed according to the iteration are shown in Figs. 10.17 and 10.18. The intermediate level of the second matrix corresponds to the elements of the marked vector. Here, the white field is the matrix element of the form $a = 1$, the black field is the element of the form $-b$, the grey field is the element of “edging” элемент $b < s < a$.

The Hadamard-Euler matrices. In [40] a class of two-level matrices is named as the Hadamard-Euler matrices. These matrices are represented by the square matrices E_n of the order n , consisting of the numbers $\pm a$ and $\pm b$. Such matrices are constructed on the basis of the formula, where $H_{n/2}$ is the two-level Hadamard-Mersenne matrix of the half odd order, which consists of the numbers $\{a = 1, -b\}$ with a recalculation of their level in such a manner as to have $b = 1/2$ at $n = 6$, and in the remaining cases, $b = \frac{q-\sqrt{8q}}{q-8}$, $q = n + 2$

Fig. 10.19 The structure of the Hadamard-Euler matrix \mathbf{E}_{30}



$$\mathbf{E}_n = \begin{bmatrix} \mathbf{H}_{n/2} & \mathbf{H}_{n/2} \\ \mathbf{H}_{n/2} & -\mathbf{H}_{n/2} \end{bmatrix}.$$

The Hadamard-Mersenne matrix of the 3rd order has the form:

$$\mathbf{H}_3 = \begin{bmatrix} a & -b & a \\ -b & a & a \\ a & a & -b \end{bmatrix}.$$

The Hadamard-Euler matrix obtained on its basis will be of the 6th order, where $a = 1$, $b = 0.5$

$$\mathbf{E}_6 = \begin{bmatrix} a & -b & a & a & -b & a \\ -b & a & a & -b & a & a \\ a & a & -b & a & a & -b \\ a & -b & a & -a & b & -a \\ -b & a & a & b & -a & -a \\ a & a & -b & -a & -a & b \end{bmatrix}.$$

The Hadamard-Euler matrices can be used in a number of cases instead of C-matrices, e.g., when the last ones do not exist. An important property of the Hadamard-Mersenne matrices is kept for them, i.e. with an increase of n the modules of all elements tend to 1. As an example, in Fig. 10.19 the structure of the Hadamard-Euler of the 30th order is demonstrated.

Thus, in asymptotes (at large n) all three described classes of matrices (Hadamard-Mersenne, Hadamard-Fermat, and Hadamard-Euler) tend to a common limit of the form of orthogonal matrices with elements ± 1 .

10.5 Conclusion

This chapter contains the description and analysis of the matrix method of transforming images, which is based on a procedure of the strip-transformation that can be considered as a finite dimensional analogue of the holographic principle of image transformation. The main tasks on investigation and implementation of the strip-method are formulated for increase of immunity with regard to pulse noises present in communication channels, determination of requirements for strip-transformation operators, development of the strip-method for the case of transmitting, storage and processing two-dimensional images, and search of optimal matrices of two-dimensional strip-transformation. In solving these tasks the main requirements for an operator of transformation are considered. It is shown that the operator should be linear, isometric, or finite dimensional. This leads to use the matrices, which have the symmetry of a certain type and have entries equal in absolute value.

The possibility to apply two-dimensional strip-transformation for storage and noise immune transmission of images is considered. At the same time two-sided matrix transformations of an original image have been used, in the process of which image fragments are mixed and superimposed on each other. Great attention is paid to the implementation of the Hadamard matrices and matrices close to them. They include Hadamard-Mersenne, Hadamard-Fermat, and Hadamard-Euler matrices.

References

1. Mironovskii LA, Slaev VA (2006) The strip method of transforming signals containing redundancy. *Meas Tech* 49(7):631–638
2. Mironovskii LA, Slaev VA (2006) The strip method of noise-immune image transformation. *Meas Tech* 49(8):745–754
3. Mironovsky LA, Slaev VA (2011) Strip-Method for Image and Signal Transformation. De Gruyter, Berlin
4. Totty RE, Clark GC (1967) Reconstruction error in waveform transmission. *IEEE Trans Inf Theory* 13(2):333–338
5. Andrews HC (1970) Computer techniques in image processing. Academic Press, New York
6. Haar A (1955) Zur Theorie der Orthogonal Funktionen-System. Inaugural Dissertation. *Math Annalen* 5:17–31
7. Livak EN Algorithms Compression. [http://mf.grsu.by/UchProc/livak/po/comprsite/theory_ jpeg.html](http://mf.grsu.by/UchProc/livak/po/comprsite/theory_jpeg.html). Accessed 14 June 2014 (in Russian)
8. Mironovsky LA, Slayev VA (1975) Equalization of the variance of a nonstationary. *Signal Telecom Radio Eng* 29–30(5):65–72
9. Costas JP (1952) Coding with Linear Systems. *Proc IRE* 40(9):1101–1103

10. Lang GR (1963) Rotational transformation of signals. *IEEE Trans Inf Theory* 9(3):191–198
11. Leith EN, Upatnieks J (1962) Reconstructed wavefronts and communication theory. *J Opt Soc America* 52(10):1123–1133
12. Medianik AI (1997) Proper simplex inscribed in a cube and hadamard matrices of the half-circulante type. *Math Phys Analis Geom* 4(4):458–471
13. Mironovskii LA, Slaev VA (2002) Optimal Chebishev pre-emphasis and filtering. *Meas Tech* 45(2):126–136
14. Paley REAC (1933) On orthogonal matrices. *J Math Phys* 12:311–320
15. Pierce WH (1968) Linear-real codes and coders. *Bell Syst Techn J* 47(6):1067–1097
16. Rao KR, Narasimhan MA, Revuluri K (1975) Image data processing by hadamard-haar transforms. *IEEE Trans Comput C-23*(9):888–896
17. Votolin DS (1998) Algorithms of images compression. Moscow State University (in Russian)
18. Mironovsky LA, Slayev VA (2013) Double-sided noise-immune strip transformation and its root images. *Meas Tech* 55(10):1120–1127
19. Selyakov IS (2005) Analysis and computer imitation of images STRIP-transformation. Master's dissertation, Saint-Petersburg State University of Aerospace Instrumentation
20. Mironovsky LA, Slaev VA (2011) Root images of two-sided noise immune strip-transformation. International workshop on physics and mathematics IWPM 2011. Hangzhou, China
21. Hadamard J (1893) Resolution d'une Question Relative aux Determinants. *Bull Sci Math ser 2* 17(1):240–246
22. Ruben S (1990) Methods of cipher video compression. *Comput Press* 10:22–30 (in Russian)
23. Umnyashkin SV (2004) Mathematical Foundations of Signal Cipher Processing and Coding. National Research University of Electronic Technology (in Russian)
24. Williamson J (1944) Hadamard's determinant theorem and the sum of four squares. *Duke J Math* 11:65–81
25. Zalmanzon LA (1989) Transformations of Furier, Walsh, Haar and their implementation in control, communication and other areas. Science, Moscow (in Russian)
26. Shintyakov DV (2006) Algorithm for searching hadamard matrices of odd order. *Techn Sci* 2:207–211
27. Contemporary Design Theory: A Collection of Essays (1992) Orthogonal arrays. Wiley, New York
28. Belevitch V (1950) Theorem of $2n$ -networks with application to conference telephony. *Electr Commun* 26:231–244
29. Balonin NA, Mironovsky LA (2006) Hadamard matrices of the odd order. *Inf Control Syst* 22(3):46–50 (in Russian)
30. Golova EA (2013) Properties investigation of M-matrices applied for filtration. Master's dissertation, Saint-Petersburg State University of Aerospace Instrumentation
31. Golub GH, van Loan CF (1989) Matrix computations, 3rd edn. John Hopkins University Press, Baltimore
32. Balonin NA, Sergeyev MB (2011) M-matrices. *Inf Control Syst* 50(1):14–21 (in Russian)
33. Balonin NA, Sergeyev MB (2011) M-matrix of the 22-th order. *Inf Control Syst* 54(5):87–90 (in Russian)
34. Hadamard matrices monitoring. <http://mathworld.wolfram.com/HadamardMatrix.html>. <http://mathscinet.ru>. Accessed 14 June 2014 (in Russian)
35. Balonin NA, Sergeyev MB, Mironovsky LA (2012) Calculation of Hadamard-Fermat matrices. *Inf Control Syst* 61(6):90–93 (in Russian)
36. Balonin NA, Sergeyev MB (2013) Two ways to construct Hadamard-Euler matrices. *Inf Control Syst* 62(1):7–10 (in Russian)
37. Balonin NA, Mironovsky LA, Sergeyev MB (2012) Computation of Hadamard-Mersenne matrices. *Inf Control Syst* 60(5):92–94 (in Russian)

38. Gantmacher F (1959) Matrix theory. Chelsea Publishing, New York
39. Gersho AB, Gray RM (1992) Vector quantization and signal compression. Kluwer, Boston
40. Golov AS (2011) Robustness Investigation of strip-method to channel noises. Bachelor's thesis, Saint-Petersburg State University of Aerospace Instrumentation

Chapter 11

A Generalized Criterion of Efficiency for Telecommunication Systems

Alexey A. Borisenko, Vyacheslav V. Kalashnikov,
Alexey E. Goryachev and Nataliya I. Kalashnykova

Abstract The chapter develops a generalized criterion to estimate the efficiency of telecommunication systems that can be applied to economics information systems, too. The criterion combines evaluation of such special properties as information quantity, noise immunity, data transmission speed, and transmission cost. In contrast to other criteria, the proposed one is non-dimensional and normalized, thus estimating a telecommunication system by means of real numbers between 0 and 1. The design of the developed criterion based upon the concept of conditional entropy is rather simple. It allows one to calculate the system's characteristic value with sufficient accuracy for practice, thus comparing various telecommunication systems to transmit the economic information. The generalized criterion is composed as a

A.A. Borisenko · V.V. Kalashnikov (✉) · A.E. Goryachev · N.I. Kalashnykova
Department of Computer Science, Sumy State University, Rimsky-Korsakov St. 2,
Sumy 40007, Ukraine
e-mail: kalash@itesm.mx; slavkamx@mail.ru; 5352008@ukr.net

A.A. Borisenko
e-mail: 5352008@ukr.net

A.E. Goryachev
e-mail: 5352008@ukr.net

N.I. Kalashnykova
e-mail: nkalash2009@gmail.com; 5352008@ukr.net

V.V. Kalashnikov
Tecnológico de Monterrey (ITESM), Campus Monterrey, Ave. Eugenio Garza Sada
2501 Sur, 64849 Monterrey, NL, Mexico

V.V. Kalashnikov
Central Economics and Mathematics Institute (CEMI), Nakhimovsky Pr. 47,
Moscow 117418, Russia

N.I. Kalashnykova
Department of Physics and Mathematics (FCFM), Universidad Autónoma
de Nuevo León (UANL), 66450 San Nicolás de los Garza, NL, Mexico

product of some partial criteria, which permits to estimate the telecommunication system not only as a whole body, but also with respect to their partial characteristics, such as their productivity, reliability, and transmission costs.

Keywords Telecommunication systems · Conditional entropy · Noise immunity · Generalized efficiency criterion

11.1 Introduction

A system of any kind designed for collecting, keeping, transforming, processing, and transmitting information is usually called a telecommunication system. Such an entity is an indispensable part of each control system that has to control either technological processes or administration bodies of enterprises. In the former case (technological processes) one gets information from various gadgets (controllers) and processes it with an aid of computers, the results being then transmitted to the objects of control in order to switch their states (often without intervention on part of personnel). As examples of the control objects mentioned above, one can list chemical reaction processes, steel manufacturing procedures, as well as machine tools, compressors, reactors, etc.

In the second case (administration of enterprises), various documents serve as the input and output pieces of information. These are accumulated in the system's memory and used to help solve many problems related to the enterprise administration, such as, for example, financial support or accounting tasks. Nowadays, information and help systems are widespread, hence databases of enormous (sometimes even global) scale are needed and created.

One of the principal features of the information systems employed in the technological (manufacturing) control is that they must react to the production's needs momentarily. The latter makes out special requirements to the speed of information's transmission and transformation: a minute delay can lead to gross material losses, and sometimes even to disastrous outcomes. Although the delays are not so critical for the administration systems, both types of information systems (technological and administrative ones) may suffer severely from possible errors in the transmitted or processed messages, which can entail substantial damages.

How can a telecommunication system be characterized? Its principal parameters are: the overhead cost, the time of recovery of the outlay, the annual gross income, the lifetime revenue, the present value of return percentage, as well as the system's speed, the reliability of data processing, the completeness of the output, the time needed to adjust the system, its structure's flexibility, and the list of tasks that the system accomplishes [1–3]. A number of these characteristics can be reduced with only the main of those kept intact: the speed, the reliability, and the cost of information processing per a unit of time. In more detail, the cost of information processing should include not only overhead capital costs but also the expenditures

generated during the running time: e.g., reparation costs in case of the system's failure, expenses for the development of additional service codes, training and publicity costs, etc. In addition, the material damage caused by decreasing the amount of information processed and transmitted via the system due to revealed errors should also be summed up to the total cost.

After a criterion evaluating a telecommunication system has been selected, the following step is to develop its general structure and to choose its optimal specific version. The chosen structure will mandatory include information sources, the number of which can reach dozens and even hundreds, as well as the tools transforming and processing the information. Next, transmission channels such as cable, telephone, optical fiber cable, or radio channels are to be selected. Each type of the transmission channels has its own specific merits and flaws related both to the transmission costs and to the transmission speed and reliability indexes. The problem of choice of one or another channel is solved first by transmission experts, whose decision is then tested numerically by making use of an accepted generalized criterion evaluating the telecommunication system as a whole. The generalized criterion should take into account the speed, reliability and cost of information transmission, as well as the possibility of use of transmission codes. Estimation of the information value and the frequency of inquires is also a must. The transmission channels and means decided, a question arises of a proper selection of the tools processing and transforming information in order to transmit it at a higher quality level. As examples of such tools, the methods and instruments of noise-proof coding can be mentioned. The latter might be not only usual computing systems but also more powerful ones boasting distributed processing tools based, e.g., on Internet.

The chapter's structure is as follows. Section 11.2 provides the basis of economic evaluation of telecommunication systems. Section 11.3 deals with the general problem statement. Later on, Sect. 11.4 shows how it is possible to manipulate with an ideal telecommunication system as a landmark for real-life devices, while Sect. 11.5 represents the main results of the chapter. Finally, in order to illustrate the developed algorithm, Sect. 11.6 considers an imaginary "real-life" telecommunication system, obtains its generalized efficiency parameter, and discusses the possible ways to increase the value of the latter by modifying the system in question. The short Sect. 11.7 with the conclusion, acknowledgments, and reference list finishes the chapter.

11.2 Economic Evaluation of Telecommunication Systems

A telecommunication system's operation quality is evaluated by the degree of achievement of the system's goals subject to restrictions on its parameters. The principal aims of a newly developed telecommunication system are: to elevate the productivity of a technological procedure, to improve the quality of a commodity, to decrease its production cost, etc. A telecommunication system may achieve such targets in multiple ways, e.g., by developing one or another version of its structure,

each being marked by an outstandingly low cost, a high reliability, a superb noise immunity, speed, low operation costs, and/or other parameters, such as, for example, the system's openness, the existence of special service functions. Because of that, the problem of an optimal structure of a telecommunication system (compared to many others) is quite a complicated task that needs a proper efficiency criterion.

Papers [1, 2] discuss some economic criteria, such as the term of return, overall and reduced costs, an equivalent income rate, etc. However, quite often, these criteria produce contradictory recommendations about the selection of the structure of a telecommunication system, hence they cannot be accepted as universal tools. Besides, the economic canons do not take into account such important technical features of telecommunication systems as their operation speed, noise immunity, and reliability, thus showing the deficiency of purely economic measures. The latter means that the quality of a telecommunication system can hardly be estimated with the aid of only economic yardsticks. Therefore, we need a kind of generalized touchstone that could combine both economic and technical criteria.

However, any accepted measure must somehow take into account the following economic characteristics of the evaluated telecommunication system: (a) buying and overhead capital expenditures W_{bo} paid, when the system is acquired and installed; (b) service costs per a time unit W_{serv} , as well as (c) the positive effect of the system's exploitation per a time unit W_+ . Then the system's payback period can be computed by Eq. 11.1.

$$T = \frac{W_{bo}}{W_+ - W_{serv}} \quad (11.1)$$

Next, the regulatory costs W_{red} during the prescriptive payback time period T_{pay} (the reduced expenditures) are determined by Eq. 11.2.

$$W_{red} = W_{bo} + T_{pay}W_{serv} \quad (11.2)$$

Therefore, the total expenditures of the telecommunication system (for the complete service time period T_{tot}) will be determined by Eq. 11.3.

$$W_{tot} = W_{bo} + T_{tot}W_{serv} \quad (11.3)$$

It is worthwhile to notice that each telecommunication system should provide for the positive effect W_+ higher than the service costs W_{serv} : only then the system's payback period will be reasonable. However, even if the difference ($W_+ - W_{serv}$) is positive, the system in question need not be optimal, since this difference may be of various magnitudes for different systems. That leads to different payback periods, only after which the use of such a system starts giving rise to net profit. In some cases, notwithstanding, it makes sense to increase the payback period, because the latter may well cut down the buying and overhead expenditures. Then the drop in the positive effect of running the system is completely reimbursed with the said

savings in the initial capital expenditures. All the above examples show that even the problem of selection of an economic efficiency measure for a telecommunication system is far from being a simple choice.

This section outlines the main items of the economic evaluation of telecommunication systems, while its Sects. 11.2.1 and 11.2.2 explain their technical-economic and purely technical assessments, respectively.

11.2.1 Technical-Economic Evaluation of Telecommunication Systems

Apart from the purely economic measures of the telecommunication systems, evaluations of their technical characteristics are also required. For instance, their running costs depend a lot upon the systems' reliability, which can be enhanced by investing solidly into higher quality equipment, diagnostic software, control systems, etc. Especially important for the telecommunication systems is the degree of reliability of the information processing and transmission, which can be elevated with the aid of special noise-proof codes. Nevertheless, the latter implies extra expenditures for coding and decoding devices with the appropriate software, which as a rule abates the transmission speed and increases the transmission cost. As a result of the lower degree of the reliability of the information processing and the delays in its transmission, the positive effect of running the telecommunication system may go down and end becoming negative, thus doing away with the system's efficiency.

Since recently, the privacy in information processing and transmission has become a crucial feature. On the one hand, this furtiveness diminishes the possible damage caused by an unauthorized intercept of the transmitted information, but on the other hand, the expenditures needed to guarantee such privacy may reach hefty sums. Moreover, the secrecy defending devices use to cut down the information transmission speed, which is accompanied with the above-mentioned minor points.

All the listed above, together with some other contradictory requirements toward telecommunication systems (e.g., taking into account both the information value and the results of its aging), impulse the search for certain generalized criteria, which would evaluate both technical and economic parameters of the systems in a combined mode.

11.2.2 Technical Evaluation of Telecommunication Systems

The technical evaluation of telecommunication systems is conducted by analyzing its levels. Nowadays, a telecommunication system usually comprises the physical, channel, network, transportation, session, representation, and application levels [4, 5].

The physical (hardware) level is represented by the material transmission channels, such as, for example, twisted pairs or coaxial cables, or wi-fi networks, through which elementary signals are dispatched [5]. Such channels are characterized with their broadband features, noise immunity, high transmission speed, appropriate voltage levels, and a series of other physical parameters. The said channels boast modems at both emitting and perceiving extremes, thus they can emit and perceive information at both ends, which allows one to ask to repeat a message should any errors of transmission occur. In order to detect transmission errors, one can use the noise-proof coding of the transmitted signals, which permits one to correct the errors either by requesting a repetition of the message, or by making use of special error-correcting codes at the perceiving extreme. This is the fastest information transmission level, hence it has the highest channel capacity, marked with its transmission speed and reliability.

The channel level is based upon a physical channel of transmission, but it provides information in the form of files, which later, when passing to the lower (physical) level, are transformed into signals. This level is formed as a functionally complete set of operations aimed at the transmission of information between the nodes of a telecommunication system. The first thing done by a channel level is to check the availability of a physical channel for transmission of information, and whether the latter is connected to both the emitting and perceiving devices in order to arrange a link between two computers. The channel level mandatory controls the transmitted information with the aid of a special noise-proof coding of the transmitted blocks of digital data. The most common way of such control is adding checksums (or, hash sums) to the transmitted digital blocks. A wrong checksum is revealed when summing up the bites of a block detects some errors that might have been generated during its transmission. Whenever that happens, the channel level suggests that the transmission of this block be repeated. Here again, just as in the case of a physical level, the channel is evaluated according to the information transmission speed and reliability it provides.

A network level is usually represented with a set of computers linked by one of the standard connection schemes that makes use of a protocol of a channel level. There may be various networks, and a kind of network connection is established among them, which is exactly what is called the network link. The latter is arranged with the help of special devices collecting information about inter-network links: these devices are called routers. A set of routers organize routes among several networks, and the selection of routes is called a routing process. The problem of routing is number one for the network level. The information transmitted among networks is then recollected and packed as information packages, the beginning of each package being endowed with the network's number and the number of its node to receive the package. All nodes of the same network are supplied with the network's number at the initial block and with the node's number at the end. Two different kinds of protocols are employed at the network level: the routing protocols conduct the communication among the networks, while the network protocols work immediately within the networks. It is clear that the information transmission time along a route is one of the decisive criteria to evaluate the route: the less this time,

the better the route. Another important criterion to evaluate a routing scheme is the reliability of the information transmission, which depends upon not only the selected network route's quality but also the level of noise affecting the latter.

The principal task of the transportation level of a telecommunication system is to prevent possible losses of information when transmitting the blocks of data to addressees. This level comprises various services that, for example, help resume communication after a break, as well as restore data blocks, and what is most important, detect and correct errors by either requesting a reiteration of the transmission of a data block, or by correcting the errors at the perceiving unit. Here, one can see the relationships between the transportation level and the lower levels of the telecommunication system. If the lower levels of a telecommunication system guarantee a high accuracy and reliability of the information transmission, there is no need in developing complicated gadgets to control errors at the transportation level. Otherwise, i.e., if the lower levels cannot boast an advanced degree of defense against transmission or storage errors, then the transportation level must provide for a higher degree of such a defense, including numerous checks of data blocks, their duplication, handshaking, analysis of checksums, etc. To resume, the main problem solved at the transportation level is the quality of bringing information to the addressee, which means both the speed and precision of the transmission procedure.

The session level is responsible for revealing (at a specified time moment), who (what) is the information source and who (what) is the information receiver, and finally, who (what) is synchronizing the telecommunication. The latter is extremely important in order to fix the connection supporting points, to which it is possible to return in case of a connection break. For practical purposes, the session level is usually attached to the transportation level, although the former boasts its own functioning protocols. Here again, the same as at the other levels of a telecommunication system, the obvious requirement is the high quality of connection articulated via the timely and exact transmission of information to the receiver. Here, the principal efforts are applied to establish a reliable synchronization of operations by both the emitting and perceiving units (ends) of the telecommunication system.

The application level of a telecommunication system is characterized by managing a plenty of protocols allowing an access to a sundry of distinct sources of the information system, which is part of the telecommunication system in question. These are: websites, reference subsystems, e-mail, printers, etc. This level permits a user to insert his/her own tools to defend the transmitted information from errors and an unauthorized access, thus enhancing the quality and reliability of the perceived messages. At the application level, the operational efficiency of the communication is a big deal as much as at the lower levels, too.

Finally, the representation level solves the problem of transforming one form of information to another without changing the contents; e.g., by transforming certain symbols into other characters. The coding-decoding items are also treated at this level.

The session, application and representation levels are special in that they are independent of the network. The latter means that any kind of information

processed within these levels doesn't depend on the network's structure and its technical features. A different network technology or new protocols used at the lower levels in no way affect the performance of the three upper levels and/or their protocols. On the contrary, the physical, channel, and network levels (and partially, the transportation level, too) are tightly related to each other, hence their protocols are changed after a variation of the network's structure or the physical carriers of information.

Based upon the above-mentioned features, it is possible to conclude that the main parameters of any telecommunication system are (independently of the level of operation) the information transmission and processing speed, as well as the reliability of the perceived data blocks. Therefore, these parameters must be included and serve as the principal technical criteria to evaluate the efficiency of the telecommunication system, no matter whether it operates within a simplest local network or in a global international one, including Internet.

11.3 Problem Specification

The information transmission and transformation are indispensable when solving a variety of applied problems. It is especially true, if one talks about telecommunication systems designed for treating economic issues with the aid of modern information technologies [3]. The telecommunication systems are responsible for the speed of information transmission, its reliability, and in the end, the efficiency of the decisions made by the administration. In order to evaluate a tested telecommunication system one needs to have an appropriate measure at hand, which may be quite a difficult problem. Indeed, the modern telecommunication systems differ from each other in many aspects, such as their structures, operation algorithms, modes of information transmission and transformation, to mention only few of the principal features.

In practice, people mostly use the well-developed particular technical criteria to evaluate the efficiency of telecommunication systems, such as their speed, noise immunity, expenditures (costs), etc. (cf., [4–7]). However, the latter touchstones are often incompatible, like for example, the system's speed and expenditure. Therefore, the particular measures used separately are unable to estimate the telecommunication system's efficiency in all its integrity. Because of that, apart from the partial measures, there is a need of generalized criteria that would make one able to compare telecommunication systems and select the most efficient among them. In order to boast such strength, on the one hand, the generalized criteria must take into account the partial characteristics, assign certain weights to the latter and moreover, provide the highest possible impartiality. On the other hand, they should be relatively simple, easily calculated and finally, giving the opportunity to compare the telecommunication systems within a numerical range fixed beforehand; in other words, they have to be normed, too. The authors of paper [8] examined one of the generalized criteria to estimate the efficiency of telecommunication systems, which

is based upon a generalized measure developed earlier for assessing Automated Control System (ACS) [9, 10]. Nevertheless, the criterion presented in [8] was lacking the detailed algorithm to compute its values, which hindered its practical use. Motivated by the desire to overcome this handicap, the authors have improved the criterion in question and developed a more reliable method to estimate its numerical values. Exactly these new results are presented in the chapter.

Section 11.3.1 contains a short survey of the relevant literature, Sect. 11.3.2 details the unsolved part of the problem, and Sect. 11.3.3 specifies the novelty of the chapter.

11.3.1 A Survey of Recent Results of Research

In the majority of cases, telecommunication systems are assessed with the aid of especially developed partial criteria. For instance, transmission systems are usually compared to each other with respect to the total expenditure, the number of subscribers available, the noise immunity, the facilities for simultaneous conferences and video-conferences with several communication partners, etc. (see [4–7]). Nonetheless, as already mentioned above, those and a plenty of other characteristics can be reduced to the following three particular features: the system's speed, noise immunity, and expenditures for information transmission. This is why the majority of practitioners select these properties as the measures to assess telecommunication systems. The rest of the attributes are either reduced to the three main qualities or brushed aside as secondary ones, at least at the very initial stage of planning the development of a new telecommunication system.

On the contrary, unfortunately, generalized criteria are not very popular, although it is exactly they that allow one to conduct the thorough evaluation of a telecommunication system thus prompting the selection of the optimal system. It is worth noting that the first generalized criterion was actually proposed by Shannon in his works, where he assessed information transmission systems. Indeed, Shannon proposed to take into account two technical parameters at the same time, namely, the information transmission speed and reliability [10, 11]. However, the latter generalized criterion did not comprise the information transmission costs, i.e., it was purely technical and not technical-economical; in addition, it wasn't normed.

11.3.2 The Unsolved Part of the Problem

Works [8–10] introduced a new criterion involving the costs, and moreover, it was generalized and normed. Nonetheless, there was no algorithm to calculate the criterion's value. Exactly such an algorithm is the crucial target to hit when developing a new generalized measure for assessing the efficiency of a telecommunication system.

11.3.3 The Results of This Chapter

The main goal of this chapter is to develop a numerical method of computing the value of a generalized criterion of efficiency of a telecommunication system, which allows one to evaluate its performance based upon the quantity of the transmitted information, the reliability of the latter, and last but not least, the transmission speed and cost.

11.4 Replacement of a Telecommunication System with Its Ideal Model

The generalized criterion of the efficiency of telecommunication systems is founded upon the idea of a generalized measure for automated control systems (cf., [9, 10]). The measure makes use of an ideal system, and the selected criterion is based upon subjective perceptions of the user (which may be quite different for various classes of such systems). Nevertheless, the criterion in question keeps in mind real (objective) factors as well, which are necessary for creating real-life systems, too. Exactly the same approach is proposed in this work.

As an ideal object, each user would be happy to have an impeccable transmission system with null service expenditures that would transmit and process all the input information without losses, and moreover, the transmission time should be comparable with that provided by the best systems of the given rank, and all that at the minimum possible cost. Of course, in reality, it is almost impossible to fulfill all these requirements at the same time, but this list may explain what an ideal telecommunication system could be like, and also serve as a landmark for those who develop and design real-life systems.

In order to solve the considered problem of assessing the efficiency of telecommunication systems, the ideas by Winograd and Cowan [12] are developed in this research, and a given telecommunication system is replaced with two different systems. Namely, the first of them will be the original telecommunication system imagined as ideal, i.e., having no errors and/or failures when working, and boasting the minimum possible time needed to transform, process, and transmit information. Meanwhile, the second one is a reduced telecommunication system that only transmits the information, i.e., it is nothing more than an information transmission system. In addition, it is supposed that the transition time achieved by the second system coincides with that of a real telecommunication system. Besides, other parameters of the second system, such as the reliability and cost of transmission are exactly the same as those of the evaluated real-life telecommunication system. This approach permits us to use the prowess of all already known methods of evaluation of transmission systems for assessing the existing telecommunication systems.

11.5 Main Results

One of the principal parameters of an information transmission system is its speed, that is, the quantity of information units transmitted per a time period: the higher this speed, the better the system. The calculation of such transmission speed is based on the assumption that the information source (input unit) A generates information in form of any of the messages $a_1, a_2, \dots, a_i, \dots, a_N$, with the corresponding probabilities $p(a_1), p(a_2), \dots, p(a_i), \dots, p(a_N)$. After that, the transmission system dispatches them to the perceiving (output) unit B , which, in turn, comprehends them as the messages $b_1, b_2, \dots, b_j, \dots, b_N$, with the respective probabilities $p(b_j) j = 1, 2, \dots, N$ (see, [11]). As is well-known (cf., [11] again), the unconditional entropy of the input unit A (representing the degree of uncertainty, when generating any of the N messages listed above) is determined by Eq. 11.4.

$$H(A) = - \sum_{i=1}^N p(a_i) \log_2 p(a_i) \quad (11.4)$$

This value characterizes an average quantity of information units $I = H(A)$ that could be theoretically transmitted toward the perceiver B under the given probability distribution of the messages and in the absence of transmission noise.

However, real-life telecommunication/transmission systems are always subject to various types of noise implying that the message $b_j, j = 1, 2, \dots, N$, might be identical to the input message a_i with certain conditional probability $p(a_i|b_j)$. As a consequence, there arises a kind of uncertainty about the fact that the message b_j perceived by output unit B is indeed the message a_i actually emitted by source A . This uncertainty is measured with the conditional entropy $H(A|B)$ provided by Eq. 11.5.

$$H(A|B) = - \sum_{j=1}^N \sum_{i=1}^N p(b_j) p(a_i|b_j) \log_2 p(a_i|b_j) \quad (11.5)$$

The conditional entropy can be interpreted as an average amount of loss of information when transmitting all possible messages from the list.

The probability $p(b_j)$ of having perceived the message b_j involved in Eq. 11.5 can be calculated by Eq. 11.6 of complete probability shown below, whenever the unconditional probability $p(a_i)$ of generating message a_i by emitting unit A , as well as the conditional probability $p(b_j|a_i)$ of perceiving message b_j instead of a_i by the output unit B , are known.

$$p(b_j) = \sum_{i=1}^N p(a_i) p(b_j|a_i) \quad (11.6)$$

Probabilities $p(b_j)$ should also satisfy the evident condition (Eq. 11.7).

$$\sum_{j=1}^N p(b_j) = 1 \quad (11.7)$$

The value of conditional entropy (Eq. 11.5) is used to update the quantity of information transmitted from input unit A , which is represented by the reduced (due to the existing noise) amount I calculated by Eq. 11.8.

$$I = H(A) - H(A|B) \quad (11.8)$$

In the case when $p(a_i|b_j) = p(a_i)$, one evidently has Eq. 11.9.

$$\begin{aligned} H(A|B) &= - \sum_{j=1}^N \sum_{i=1}^N p(b_j) p(a_i|b_j) \log_2 p(a_i|b_j) \\ &= - \sum_{j=1}^N \sum_{i=1}^N p(b_j) p(a_i) \log_2 p(a_i) \\ &= \sum_{j=1}^N p(b_j) \left[- \sum_{i=1}^N p(a_i) \log_2 p(a_i) \right] \\ &= H(A) \sum_{j=1}^N p(b_j) = H(A) \end{aligned} \quad (11.9)$$

The latter, together with Eq. 11.8, clearly entails that the quantity of information I transmitted from input unit A is zero, and no statistical relationship among the emitted and perceived messages exists. On the contrary, if there is no noise, one contemplates a rigid statistical relationship, under which one of the conditional probability values is 1 (for example, $p(a_i|b_j) = 1$), whereas all the others are 0 (that is, $p(a_i|b_k) = 0$, $k \neq j$). Indeed, under this assumption (Eq. 11.5) leads to $H(A|B) = 0$, which means that the complete information emitted by input unit A and contained in message a_i will be transmitted to output unit B without distortions in the quantity $I = H(A)$.

On the other hand, perceiving unit B is also characterized by its own entropy with respect to entering messages b_j perceived with the corresponding probabilities $p(b_j)$, $j = 1, 2, \dots, N$, when transmitted from emitting unit A . Hence, the receiver unit's entropy is described by Eq. 11.10.

$$H(B) = - \sum_{j=1}^N p(b_j) \log_2 p(b_j) \quad (11.10)$$

Analogous to Eq. 11.5, the conditional entropy of the receiving device B given the emitting unit A is determined via unconditional and conditional probability values $p(a_i)$ and $p(b_j|a_i)$, as shown below in Eq. 11.11.

$$H(B|A) = - \sum_{i=1}^N \sum_{j=1}^N p(a_i)p(b_j|a_i) \log_2 p(b_j|a_i) \quad (11.11)$$

The latter conditional entropy measures the loss of information occurring at receiving unit B owing to some uncertainty about the message a_i that appears at emitting unit A . This uncertainty, caused by the transmission noise, arises because input unit A may not be sure about which of the messages $a_i, j = 1, 2, \dots, N$, will be perceived as message b_j by receiver B .

If conditional entropy $H(B|A)$ is nonzero, the information quantity transmitted to the receiver may be reduced to the value provided by Eq. 11.12.

$$I = H(B) - H(B|A) \quad (11.12)$$

Whenever the quantities of emitted and perceived information are the same, one has Eq. 11.13.

$$H(A) - H(A|B) = H(B) - H(B|A) \quad (11.13)$$

It is easy to see that if the transmission system is flawless and no information losses occur, then $H(B) = H(A)$, hence the corresponding unconditional probabilities coincide: $p(a_i) = p(b_j)$.

In order to simplify the numerical operations with the data, we will represent the conditional probabilities through a channel matrix, the i th row of which comprises the conditional probabilities of transition of the message a_i emitted by input unit A to all possibly perceived messages $b_j, j = 1, \dots, N$, by receiver B . For example, row 1 corresponding to emitted message a_1 contains the following (conditional) probabilities of the former message being perceived as message $b_j, j = 1, \dots, N$ with the sum of these entries equal to 1 mentioned below:

$$[p(b_1|a_1) \quad p(b_2|a_1) \quad \dots \quad p(b_N|a_1)]$$

All the remaining rows of the matrix have a similar structure (see, transition matrices used in the Markov chains).

Example 1 If an information source generates two signals 0 and 1 and they are transmitted to a receiver through a channel with noise, the receiving unit may perceive 1 instead of the transmitted signal of 0, and vice versa, perceive 0 rather than the transmitted 1. These possible transitions can be described with the following channel matrix, whose entries are the corresponding conditional probability values described by Eq. 11.14.

$$P = \begin{bmatrix} p(0|0) & p(1|0) \\ p(0|1) & p(1|1) \end{bmatrix} \quad (11.14)$$

In case of a noiseless channel, matrix (Eq. 11.14) reduces to the unit matrix (Eq. 11.15).

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (11.15)$$

Here, row 1 means that if the source generates the information unit 0 and the latter is transmitted via the channel to the receiver, then the latter perceives it as 0 with probability 1 and erroneously comprehends as 1 with probability 0. Similarly, row 2 confirms that if the generated and transmitted information unit is 1, then the receiving unit perceives it as 0 with probability zero, and recognizes it as 1 with probability 1.

However, if the transmission channel is noisy, then receiver B having gotten a message b_j from source A has to decide (based upon the conditional probability values $p(a_i|b_j)$, $i, j = 1, \dots, N$), which of the N possible input messages a_i generated by source A has been really sent. In that case, conditional probabilities $p(a_i|b_j)$ could also be arranged in form of a channel matrix with its columns containing these conditional probabilities $p(a_i|b_j)$ of the event that receiver B has recognized message a_i having obtained message b_j . Applied to Example 1, the channel matrix for the binary signals will read as Eq. 11.16.

$$P^T = \begin{bmatrix} p(0|0) & p(0|1) \\ p(1|0) & p(1|1) \end{bmatrix} \quad (11.16)$$

The last expression is clearly the transposed channel matrix (Eq. 11.14). Therefore, in the latter case, summing the conditional probabilities as entries of each column produces 1.

Section 11.5.1 demonstrates the proposed formula for the generalized criterion, and Sect. 11.5.2 describes the new algorithm that helps calculate the telecommunication system's generalized efficiency parameter.

11.5.1 Evaluation of Efficiency of Telecommunication Systems

The knowledge of the information quantities contained in transmitted messages helps estimate such an important parameter of a real-life telecommunication system as its speed or productivity V , which is determined as the ratio of the transmitted information quantity and the average time taken for its transmission (Eq. 11.17).

$$V = \frac{H(B) - H(B|A)}{T} \quad (11.17)$$

If the transmission system is free from delays caused by correction of errors, failures of devices, etc., then the average transmission time equals the minimum time $T = t_{\min}$ necessary for the proper information transmission. Besides, whenever the telecommunication system is immune to information losses then the system's ideal productivity or transmission capacity is obtained by Eq. 11.18.

$$C_{ideal} = \frac{H(B)}{t_{\min}} = \frac{H(A)}{t_{\min}} \quad (11.18)$$

If a telecommunication system can afford only the minimal necessary expenditures W_{\min} to acquire the indispensable technical hardware and software, then the generalized efficiency parameter for such a system is proposed in form of Eq. 11.19.

$$E_{ideal} = \frac{H(A)}{t_{\min} W_{\min}} = \frac{C_{ideal}}{W_{\min}} \quad (11.19)$$

Now let us introduce a generalized efficiency criterion for a real-life telecommunication system. In order to do that, one have to take into account the reduction of the transmitted information insinuated by the transmission noise. The latter is achieved by replacing the information quantity transmitted by an ideal system $I_{ideal} = H(A)$ with that transmitted by a real-life system, which equals $HI_{ideal} = H(A) - H(A|B)$. It means that the amount of lost information is equal to $H(A|B)$ bits. In addition, the minimal transmission time t_{\min} must be increased by an average delay time t_{delay} generated by various causes, including the channel noise and failures of the hardware. As a consequence of these delays, the extra expenditures W_{exp} needed to remove the delays should be added to the minimal costs W_{\min} . As a result, the following generalized efficiency parameter of a real-life telecommunication system is represented by Eq. 11.20, where C_{real} is computed by Eq. 11.21.

$$E_{real} = \frac{H(A) - H(A|B)}{(t_{\min} + t_{delay})(W_{\min} + W_{exp})} = \frac{C_{real}}{W_{\min} + W_{exp}} \quad (11.20)$$

$$C_{real} = \frac{H(A) - H(A|B)}{t_{\min} + t_{delay}} \leq C_{ideal} \quad (11.21)$$

Parameter C_{real} is the real productivity of the telecommunication system counted in bits-per-second (or bps) units. Nevertheless, the latter criterion is not normed. In order to amend this flaw, the real efficiency parameter of the telecommunication system can be divided by its ideal efficiency value, and the generalized normed

measure is calculated by Eq. 11.22, where the value of E is dimensionless and belongs to the interval $[0, 1]$.

$$\begin{aligned}
 E &= \frac{E_{real}}{E_{ideal}} = \frac{V \times W_{\min}}{C_{ideal}(W_{\min} + W_{\exp})} \\
 &= \frac{[H(A) - H(A|B)] \times t_{\min} \times W_{\min}}{H(A) \times (t_{\min} + t_{delay}) \times (W_{\min} + W_{\exp})}
 \end{aligned} \tag{11.22}$$

Indeed, on the one hand, the value of E is equal to zero only in the case of an extremely noisy channel, when the conditional entropy $H(A|B)$ achieves the absolute entropy value $H(A)$; on the other hand, it reaches 1 if there is no noise, i.e., $H(A|B) = 0$. Meanwhile, both the expenditure costs W_{\exp} and the information transmission delay time t_{delay} are null, which is almost impossible to guarantee for a real-life telecommunication system. It is not difficult to see that the higher (i.e., closer to 1) the value of E , the more efficient is the analyzed telecommunication system. The latter means that the proposed generalized efficiency criterion (Eq. 11.22) provides for the possibility of finding the most efficient item among the existing telecommunication systems.

The considered generalized efficiency criterion (Eq. 11.22) can be further modified by including in its structure not only the telecommunication system's proper costs W_{\min} and its employment expenditures W_{\exp} but also the lost profit values caused by the decreases in the quality of communication owing to possible noises and hardware failures. The latter profit losses are evaluated by introducing special information utility functions for various telecommunication systems (see, e.g., [13]).

11.5.2 An Algorithm to Assess the Efficiency of a Telecommunication System

Based upon the above-mentioned techniques, the following method to calculate the value E of the generalized efficiency measure for a telecommunication system can be proposed. The latter is described with the algorithmic steps below:

1. Determine the number N of all possible messages to be transmitted, as well as their both unconditional and conditional probabilities to be transmitted and recognized. The said probability values are usually found by experiments, for example, following the way proposed in [11]. Nonetheless, they can be well-known beforehand, too: since the statistical properties of various alphabets have been thoroughly examined, the probabilities of generating distinct letters in an average message could be given even without experiments. Finally, in the complete absence of any hints about the necessary probabilities, one can accept them all equal to $1/N$, which, however, may decrease the accuracy of conclusions.

2. Compose the appropriate channel matrix with its entries being the (conditional) probabilities of transition of original messages to correct or incorrect ones, caused by either the source or the receiver.
3. Making use of the above-defined conditional and unconditional probability values determine the unconditional entropies of both the information emitting and receiving units. Then apply the channel matrices and calculate the source's and receiver's conditional entropies, as well as the quantities of transmitted information according to Eqs. 11.4–11.13.
4. Select the parameters of an ideal telecommunication system in line with the above-mentioned recommendations and Eqs. 11.17–11.18.
5. Calculate the possible time delays and expenditure costs of the real-life telecommunication system by taking into account the parameters of the subsystems that defend the transmitted information from possible errors and failures of hardware. Compute then the real-life system's speed/capacity parameter of the transmission system by Eq. 11.21.
6. Calculate the ideal and real efficiency parameters E_{ideal} and E_{real} of the evaluated telecommunication system according to Eqs. 11.19–11.20, respectively.
7. Find the value E of the generalized criterion of efficiency of the given telecommunication system by Eq. 11.22.
8. Compare the thus obtained generalized efficiency parameter with those for other versions of the telecommunication system and select the best (i.e., with the largest value of E) among them.
9. Finally, decide about the selection of the telecommunication system with the best (the highest) value of the generalized efficiency criterion E .

11.6 An Example of Evaluation of the Efficiency of a Telecommunication System

Assume that when designing a real-life telecommunication system, a model has been selected in such form that an information source A generating each time one of three possible messages (signals, symbols, etc.) is connected to a receiver B (perceiving those signals) with a noisy transmission channel described by the following channel matrix $P = P(B|A) = [p(b_j|a_i)]_{i=1, j=1}^N$ from Eq. 11.23.

$$P = P(B|A) = \begin{bmatrix} 0.98 & 0.01 & 0.01 \\ 0.10 & 0.75 & 0.15 \\ 0.20 & 0.30 & 0.50 \end{bmatrix} \quad (11.23)$$

Besides, the probabilities of generation of each of the three messages a_i , $i = 1, 2, 3$, by source A are as follows: $p(a_1) = 0.70$, $p(a_2) = 0.20$, $p(a_3) = 0.10$.

An ideal telecommunication system desired by the consumers who ordered the device should transmit one message from source unit A to receiver B in 1 micro-second (mcs), cost \$1,000,000 of conditional currency units (ccu), and be free of information losses and/or running expenditures.

In reality, after having completed the system's development, the latter turns out to cost \$1,300,000 ccu, and to transmit one message per 2 mcs. Here, the transmission time includes the delays occurring because of the extra time needed to correct the errors generated due to the channel noise. One need determine the value of the generalized efficiency criterion to evaluate the given real-life telecommunication system with the channel matrix (Eq. 11.23) and propose some steps to improve the system as much as possible.

The given channel matrix $P = P(B|A)$ entails that the message a_1 emitted by source A is expected to arrive to receiver B as is with probability of 0.98, whereas with probabilities of 0.01 each, it would be perceived erroneously as the other messages a_2 or a_3 , respectively. Similarly, the channel matrix (Eq. 11.23) predicts that message a_2 would be perceived correctly with probability of 0.75, while being recognized erroneously as a_1 and/or a_3 with probabilities of 0.10 and 0.15, respectively. At last, message a_3 generated and transmitted from source A would arrive as is to receiver B with probability of 0.5 or might be mistaken for the messages a_1 and a_2 with probabilities of 0.20 and 0.30, respectively.

In the absence of information losses within an ideal telecommunication system, all the information generated by source A will be transmitted safely to receiver B , in the exact quantity determined according to Eq. 11.4 as mentioned in Eq. 11.24.

$$\begin{aligned} H(A) &= -(0.7\log_2 0.7 + 0.2\log_2 0.2 + 0.1\log_2 0.1) \\ &= 1.1568 \text{ bits} \end{aligned} \quad (11.24)$$

Now determine by Eq. 11.25 the ideal telecommunication system's efficiency parameter value E_{ideal} (according to Eq. 11.19):

$$E_{ideal} = \frac{H(A)}{t_{\min} W_{\min}} = \frac{1.1568}{10^{-6} \times 10^6} = 1.1568 \text{ bps/ccu} \quad (11.25)$$

In order to evaluate the information quantity transmitted from the source to the receiver within the real-life telecommunication system, first the receiver's (unconditional) entropy $H(B)$ according to Eq. 11.10 is determined, then the conditional entropy $H(B|A)$ making use of Eq. 11.11 is calculated, and finally, the latter is subtracted from the former.

Before one gets $H(B)$ by Eq. 11.10, the unconditional probabilities $p(b_1), p(b_2)$, and $p(b_3)$ (via operations listed in the complete probability (Eq. 11.6)) ought to be computed by Eq. 11.26.

$$\begin{aligned} p(b_1) &= p(a_1)p(b_1|a_1) + p(a_2)p(b_1|a_2) + p(a_3)p(b_1|a_3) \\ &= 0.7 \cdot 0.98 + 0.2 \cdot 0.1 + 0.1 \cdot 0.2 = 0.726 \end{aligned} \quad (11.26)$$

Doing the same for the two remaining messages one gets $p(b_2) = 0.187$ and $p(b_3) = 0.087$. The checksum operation confirms that the sum of the above-computed probabilities is 1. Now substituting them into (Eq. 11.10) yields the unconditional entropy by Eq. 11.27.

$$\begin{aligned} H(B) &= -(0.726 \log_2 0.726 + 0.187 \log_2 0.187 + 0.087 \log_2 0.087) \\ &= 1.095 \text{ bits} \end{aligned} \quad (11.27)$$

The latter entropy is lower than $H(A)$ due to the existing transmission noise. In the absence of noise, these two values would be equal: $H(A) = H(B)$.

The average amount of information lost during the transmission of one message due to the noise is determined by Eq. 11.11 as follows from Eq. 11.28.

$$\begin{aligned} H(B|A) &= -0.7(0.98 \log_2 0.98 + 2 \cdot 0.01 \log_2 0.01) \\ &\quad - 0.2(0.75 \log_2 0.75 + 0.1 \log_2 0.1 + 0.15 \log_2 0.15) \\ &\quad - 0.1(0.2 \log_2 0.2 + 0.3 \log_2 0.3 + 0.5 \log_2 0.5) = 0.456 \text{ bits} \end{aligned} \quad (11.28)$$

The difference between the values of $H(B)$ and $H(B|A)$ yielded with the aid of Eq. 11.12 provides for the quantity of correct information transmitted to the receiver with one message. According to the previous calculations conducted in Eqs. 11.27–11.28, the latter amount of information equals $1.095 - 0.456 = 0.639$ bits. For instance, if the total number of such messages transmitted from the source is 400, then the receiver will obtain only 255.6 bits of the total 438 bits sent.

Now we are in a position to use Eq. 11.20 and determine the generalized efficiency of the real-life telecommunication system as given below in Eq. 11.29.

$$E_{real} = \frac{H(A) - H(A|B)}{(t_{\min} + t_{delay})(W_{\min} + W_{exp})} = \frac{0.639}{2 \cdot 10^{-6} \times 1.3 \cdot 10^6} = 0.246 \text{ bps/ccu} \quad (11.29)$$

Finally, the ratio of the efficiencies of the real-life and ideal telecommunication systems according to Eq. 11.22 yields the normed and dimensionless generalized efficiency of the evaluated system by Eq. 11.30.

$$E = \frac{E_{real}}{E_{ideal}} = \frac{0.246}{1.1568} = 0.212 \quad (11.30)$$

The latter result shows that the analyzed telecommunication system has a rather low efficiency: the existing channel noise drops it to only 21.2 %. One of the possible ways to increase the efficiency of the telecommunication system consists in

enhancing the degree of information protection by the use of certain noise-proof coding procedures, which, however, leads to the elevated transmission time and total system cost. Nevertheless, such a decision often proves to be reasonable, because the financial losses caused by a noisy channel could be much higher than the expenditures entailed by the information protection devices and software added to the telecommunication systems.

To continue the illustration by means of the considered example, assume that the selected noise-proof coding procedure increases the bulk of correct information reaching the receiver via one message from the previous 0.639 to 1 bit. It is clear that the transmission time will inevitably jump up, too. In order to take that into account, suppose that the transmission time needed to send one bit of information within the designed telecommunication system, previously being 2 mcs, now, after applying the noise-proof coding procedure, has grown up to 3 mcs. Besides, presume that the higher degree of information protection adds to the reliability of the received messages, thus cutting down the telecommunication system's total cost, e.g., from the previous \$1,300,000 to \$1,200,000 ccu. In this case, it is easy to find that the efficiency of the improved real-life telecommunication system enhances up to the value provided by Eq. 11.31.

$$E_{real} = \frac{H(A) - H(A|B)}{(t_{min} + t_{delay})(W_{min} + W_{exp})} = \frac{1.000}{3 \cdot 10^{-6} \times 1.2 \cdot 10^6} = 0.278 \text{ bps/ccu} \quad (11.31)$$

Then the normed generalized efficiency E grows too (Eq. 11.32).

$$E = \frac{E_{real}}{E_{ideal}} = \frac{0.278}{1.1568} = 0.240 \quad (11.32)$$

Therefore, for this imaginary "real-life" telecommunication system, the noise-proof coding elevates its generalized information transmission efficiency, which allows one to conclude that such a modification of the system's design should be undertaken.

11.7 Conclusion

The criterion of efficiency for telecommunication systems proposed in the chapter is generalized, normed, and dimensionless. It evaluates telecommunication systems on the base of their productivity (speed), the reliability of the perceived messages, as well as the system's total cost and running expenditures. Such a structure of the generalized criterion provides for the possibilities to compare the telecommunication systems with respect to the above-mentioned partial criteria, too. The algorithm that helps calculate the generalized efficiency parameter allows one to evaluate a

real-life telecommunication system with a precision enough for practical purposes. In addition, the generalized criterion permits to compare various telecommunication systems with the goal to select the optimal one.

Acknowledgments The research activity of the second author was financially supported by the R & D Department (Cátedra de Investigación) CAT-174 of the Tecnológico de Monterrey (ITESM), Campus Monterrey, and by the SEP-CONACYT project CB-2013-01-221676, Mexico. Also, the work of the fourth author was supported the National Council of Science and Technology (CONACyT) of Mexico as part of the project CB-2009-01-127691 and PAICYT project No. CE250-09.

References

1. Chen CT, Cheng HL (2009) A comprehensive model for selecting information system project under fuzzy environment. *Int J Project Manage* 27(4):389–399
2. Govindan M, Tang CM (2010) Information system evaluation: an ongoing measure. *Int J Busi Inform Sys* 6(3):336–353
3. Apte U, Karmarkar U (eds) (2007) *Managing in the information economy*, vol 1. Springer Science + Business Media, New York
4. Olifer V, Olifer N (2005) *Computer networks: principles, technologies protocols*. Wiley, Hoboken, NJ
5. Tanenbaum AS (2003) *Computer networks*, 4th edn. Prentice Hall, Upper Saddle River, NJ
6. Jacquet P (2009) Shannon capacity in poisson wireless network model. *Prob Info Transm* 45 (3):193–203
7. Chernega V, Plattner B (2013) *Computer networks*. SevNTU-Verlag, Sevastopol
8. Borisenko AA, Onanchenko EL, Teletov AS, Chernysh VI (1997) A generalized criterion for evaluation of efficiency of a data transmission system. *Vestnik SumGU* 2(8):101–103 (in Russian)
9. Yüksel S, Başar T (2013) *Stochastic networked control systems*. Springer Science + Business Media, New York
10. Reza FM (1994) *An introduction to information theory*. Dover Publications Inc, New York
11. Jones GA, Jones J (2000) *Information and coding theory*. Springer, London
12. Winograd S, Cowan JD (1963) *Reliable computation in the presence of noise*. The MIT Press, Cambridge, MA
13. Checkland P, Scholes J (1997) *Soft systems methodology in action*. Wiley, Salisbury, UK