

Computational Investigation of Feature Extraction and Image
Organization

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Xiuwen Liu, B.Eng., M.S., M.S.

* * * * *

The Ohio State University

1999

Dissertation Committee:

Prof. DeLiang L. Wang, Adviser

Prof. Song-Chun Zhu

Prof. Anton F. Schenk

Prof. Alan J. Saalfeld

Approved by

Adviser

Department of Computer
and Information Science

© Copyright by

Xiuwen Liu

1999

ABSTRACT

This dissertation investigates computational issues of feature extraction and image organization at different levels. Boundary detection and segmentation are studied extensively for range, intensity, and texture images. We developed a range image segmentation system using a LEGION network based on a similarity measure consisting of estimated surface properties. We propose a nonlinear smoothing algorithm through local coupling structures, which exhibits distinctive temporal properties such as quick convergence.

We propose spectral histograms, consisting of marginal distributions of a chosen bank of filters, as a generic feature vector based on that early steps of human visual processing can be modeled using local spatial/frequency representations. Spectral histograms are studied extensively in texture modeling, classification, and segmentation. Experiments in texture synthesis and classification demonstrate that spectral histograms provide a sufficient and unified feature in capturing perceptual appearance of textures. Spectral histograms improve significantly the classification performance for challenging texture images. We also propose a model for texture discrimination based on spectral histograms which matches existing psychophysical data. A new energy functional for image segmentation is proposed. With given regional features, an iterative and deterministic algorithm for segmentation is derived. Satisfactory results

are obtained for natural texture images using spectral histograms. We also developed a novel algorithm which automatically identifies homogeneous texture features from input images. By incorporating texture structures, we achieve accurate texture boundary localization through a new distance measure. With extensive experiments, we demonstrate that spectral histograms provide a generic feature which can be used effectively to solve fundamental vision problems.

Based on a novel and biologically plausible boundary-pair representation, perceptual organization is studied. A network is developed which can simulate many perceptual phenomena through temporal dynamics. Boundary-pair representation provides a unified explanation of edge- and surface-based representations.

A prototype system for automated feature extraction from remote sensing images is developed. By combining the advantages of the learning-by-example method and a locally coupled network, a generic feature extraction system is feasible. The system is tested by extracting hydrographic features from large images of natural scenes.

In memory of my parents, Fu-Lu Liu and She-Zi Liu, who taught me values and knowledge silently.

ACKNOWLEDGMENTS

I express my gratitude for my advisor, Prof. DeLiang Wang, who not only generously gives his time and energy, but also teaches me fundamental principles that are essential for my scientific career. He not only gives me many scientific insights and ideas, but also takes every chance to improve my skills in presentation and communication. I would also like to thank Prof. Song-Chun Zhu for sharing his time and ideas with me. I benefit much from his computational thinking of vision problems.

I would like to thank my colleagues at Department of Computer and Information Science, Department of Civil and Environmental Engineering and Geodetic Science, and Center for Mapping for providing me an excellent environment for doing research. I am especially grateful for Dr. John D. Bossler providing me opportunities to work on challenging and yet fruitful problems. I would also like to thank Dr. Anton F. Schenk, Dr. Alan J. Saalfeld, Dr. J. Raul Ramirez, Dr. Joseph C. Loon, Dr. Ke Chen, Dr. Shannon Campbell, and many other faculty members and colleagues for their strong support. I would also express my thanks to my colleagues in the Vision Club at The Ohio State University, Dr. James Todd, Dr. Delwin Lindsey, and Dr. Tjeerd Dijkstra, for stimulating discussions. Many thanks go to my teammates, Dr. Erdogan Cesmeli, Mingying Wu, and Qiming Luo for their help and insightful discussions. A Presidential Fellowship from The Ohio State University helped me

focus on my dissertation work in the last year of my Ph.D. study and is greatly acknowledged.

I would like to thank my Lord Jesus Christ for His wonderful guidance, arrangements, and opportunities He gives especially to me. I would like to express my sincere gratitude for the strong support from my family. My mother-in-law takes a good care of our family so that both my wife and I can focus on our studies. My wife Xujing provides a comfort and reliable home for me. Without her support and encouragement, it would be impossible for me to finish my study. I thank my daughter Teng-Teng for the enjoy we have together and for her support. I thank my families in China, my sisters and brothers for their encouragement, understanding and support.

VITA

August 14, 1966 Born - Hebei Province, China

July, 1989 B.Eng. Computer Science,
Tsinghua University, Beijing, China

August, 1989 - February, 1993 Assistant Lecturer,
Tsinghua University, Beijing, China

March, 1995 M.S. Geodetic Science and Surveying,
The Ohio State University

June, 1996 M.S. Computer & Information Science,
The Ohio State University

PUBLICATIONS

Journal Articles

X. Liu and J. R. Ramirez, “Automated vectorization and labeling of very large hypsographic map images using a contour graph.” *Surveying and Land Information Systems*, vol. 57(1), pp. 5-10, 1997.

X. Liu and D. L. Wang, “Range image segmentation using an oscillatory network.” *IEEE Transactions on Neural Networks*, vol. 10(3), pp. 564-573, 1999.

X. Liu, D. L. Wang, and J. R. Ramirez, “Boundary detection by contextual nonlinear smoothing.” *Pattern Recognition*, 1999.

Conference Papers

Y. Li, B. Zhang, and **X. Liu**, “A robust motion planner for assembly robots.” In *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, p. 1016, 1993.

X. Liu and D. L. Wang, "Range image segmentation using an oscillatory network." In *Proceedings of the 1997 IEEE International Conference on Neural Networks*, vol. 3, pp. 1656-1660, 1997.

J. J. Loomis, **X. Liu**, Z. Ding, K. Fujimura, M. L. Evans, and H. Ishikawa, "Visualization of plant growth." In *Proceedings of the 1997 IEEE Conference on Visualization*, pp. 475-478, 1997.

X. Liu and J. R. Ramirez, "Automatic extraction of hydrographic features in digital orthophoto images." In *Proceedings of GIS/LIS'1997*, pp. 365-373, 1997.

X. Liu, D. L. Wang, and J. R. Ramirez, "Extracting hydrographic objects from satellite images using a two-layer neural network." In *Proceedings of the 1998 International Joint Conference on Neural Networks*, vol. 2, pp. 897-902, 1998.

X. Liu, D. L. Wang, and J. R. Ramirez, "A two-layer neural network for robust image segmentation and its application in revising hydrographic features." *International Archives of Photogrammetry and Remote Sensing*, vol. 32, part 3/1, 464-472, 1998.

X. Liu, D. L. Wang, and J. R. Ramirez, "Oriented Statistical Nonlinear Smoothing Filter." In *Proceedings of the 1998 International Conference on Image Processing*, vol. 2, pp. 848-852, 1998.

X. Liu, "A prototype system for extracting hydrographic regions from Digital Orthophoto Quadrangle images." In *Proceedings of GIS/LIS'1998*, pp. 382-393, 1998.

X. Liu and D. L. Wang, "A boundary-pair representation for perception modeling." In *Proceedings of the 1999 International Joint Conference on Neural Networks*, 1999.

X. Liu and D. L. Wang, "Modeling perceptual organization using temporal dynamics." In *Proceedings of the 1999 International Joint Conference on Neural Networks*, 1999.

Technical Report

J. J. Loomis, Z. Ding, **X. Liu**, K. Fujimura, and H. Ishikawa, "Flexible Object Reconstruction from Temporal Image Series." Technical Report OSU-CISRC-5/96-TR30, Department of Computer and Information Science, The Ohio State University, 1996.

X. Liu and D. L. Wang, “Range Image Segmentation Using a LEGION Network.” Technical Report OSU-CISRC-10/96-TR49, Department of Computer and Information Science, The Ohio State University, 1996.

X. Liu, D. L. Wang, and J. R. Ramirez, “Boundary Detection by Contextual Non-linear Smoothing.” Technical Report OSU-CISRC-7/98-TR21, Department of Computer and Information Science, The Ohio State University, 1998.

K. Chen, D. L. Wang, and **X. Liu**, “Weight adaptation and oscillatory correlation for image segmentation.” Technical Report OSU-CISRC-8/98-TR37, Department of Computer and Information Science, The Ohio State University, 1998.

X. Liu, K. Chen, and D. L. Wang, “Extraction of hydrographic regions from remote sensing images using an oscillator network with weight adaptation.” Technical Report OSU-CISRC-4/99-TR12, Department of Computer and Information Science, The Ohio State University, 1999.

FIELDS OF STUDY

Major Field: Computer and Information Science

Studies in:

Perception and Neurodynamics	Prof. DeLiang L. Wang
Machine Vision	Prof. Song-Chun Zhu
Digital Photogrammetry	Prof. Anton F. Schenk
Geographic Information Systems	Prof. Alan J. Saalfeld

TABLE OF CONTENTS

	Page
Abstract	ii
Dedication	iv
Acknowledgments	v
Vita	vii
List of Tables	xiii
List of Figures	xiv
Chapters:	
1. Introduction	1
1.1 Motivations	1
1.2 Thesis Overview	6
2. Range Image Segmentation Using a Relaxation Oscillator Network	9
2.1 Introduction	10
2.2 Overview of the LEGION Dynamics	13
2.2.1 Single Oscillator Model	13
2.2.2 Emergent Behavior of LEGION Networks	15
2.3 Similarity Measure for Range Images	20
2.4 Experimental Results	25
2.4.1 Parameter Selection	25
2.4.2 Results	26
2.4.3 Comparison with Existing Approaches	33
2.5 Discussions	35

2.5.1	Biological Plausibility of the Network	35
2.5.2	Comparison with Pulse-Coupled Neural Networks	36
2.5.3	Further Research Topics	38
3.	Boundary Detection by Contextual Nonlinear Smoothing	40
3.1	Introduction	41
3.2	Contextual Nonlinear Smoothing Algorithm	45
3.2.1	Design of the Algorithm	45
3.2.2	A Generic Nonlinear Smoothing Framework	49
3.3	Analysis	51
3.3.1	Theoretical Results	51
3.3.2	Numerical Simulations	54
3.4	Experimental Results	58
3.4.1	Results of the Proposed Algorithm	58
3.4.2	Comparison with Nonlinear Smoothing Algorithms	65
3.5	Conclusions	73
4.	Spectral Histogram: A Generic Feature for Images	75
4.1	Introduction	76
4.2	Spectral Histograms	82
4.2.1	Properties of Spectral Histograms	85
4.2.2	Choice of Filters	85
4.3	Texture Synthesis	87
4.3.1	Comparison with Heeger and Bergen's Algorithm	96
4.4	Texture Classification	101
4.4.1	Classification at Fixed Scales	104
4.4.2	Classification at Different Scales	105
4.4.3	Image Classification	108
4.4.4	Training Samples and Generalization	111
4.4.5	Comparison with Existing Approaches	113
4.5	Content-based Image Retrieval	115
4.6	Comparison of Statistic Features	121
4.7	A Model for Texture Discrimination	124
4.8	Conclusions	129
5.	Image Segmentation Using Spectral Histograms	131
5.1	Introduction	131
5.2	Formulation of Energy Functional for Segmentation	134
5.3	Algorithms for Segmentation	135

5.4	Segmentation with Given Region Features	139
5.4.1	Segmentation at a Fixed Integration Scale	140
5.4.2	Segmentation with Multiple Scales	150
5.4.3	Region-of-interest Extraction	153
5.5	Automated Seed Selection	156
5.6	Localization of Texture Boundaries	160
5.7	Discussions	164
5.8	Conclusions	168
6.	Perceptual Organization Based on Temporal Dynamics	169
6.1	Introduction	170
6.2	Figure-Ground Segregation Network	172
6.2.1	Boundary-Pair Representation	172
6.2.2	Incorporation of Gestalt Rules	175
6.2.3	Temporal Properties of the Network	177
6.3	Surface Completion	178
6.4	Experimental Results	179
6.5	Conclusions	184
7.	Extraction of Hydrographic Regions from Remote Sensing Images Using an Oscillator Network with Weight Adaptation	188
7.1	Introduction	189
7.2	Weight Adaptation	193
7.3	Automated Seed Selection	200
7.4	Experimental Results	202
7.4.1	Parameter Selection	203
7.4.2	Synthetic Image	203
7.4.3	Hydrographic Region Extraction from DOQQ Images	204
7.5	Discussions	218
8.	Conclusions and Future Work	221
8.1	Contributions of Dissertation	221
8.2	Future Work	222
8.2.1	Correspondence Through Spectral Histograms	222
8.2.2	Integration of Bottom-up and Top-down Approaches	223
8.2.3	Psychophysical Experiments	227
8.3	Concluding Remarks	228
	Bibliography	229

LIST OF TABLES

Table	Page
3.1 Quantitative comparison of boundary detection results shown in Figure 3.15.	71
3.2 Quantitative comparison of boundary detection results shown in Figure 3.16.	71
4.1 L_1 -norm distance of the spectral histograms and RMS distance between images.	99
4.2 Classification errors of methods shown in [108] and our method . . .	115
4.3 Comparison of texture discrimination measures	128
7.1 Comparison of error rates using neural network classification and the proposed method.	205

LIST OF FIGURES

Figure	Page
1.1 A texture image and the corresponding numerical arrays. (a) A texture image with size 128×64 . (b) A small portion with size 40×30 of (a) centered at pixel $(64, 37)$, which is on the boundary between the two texture regions. (c) Numerical values of (b). To save space, the values are displayed in hexadecimal format.	2
1.2 Demonstration of nonlinearity for texture images. (a) A regular texture image. (b) The image in (a) was circularly shifted left and downward for 2 pixels at each direction. (c) The pixel-by-pixel average of (a) and (b). The relative variance defined in (3.20) between (a) and (b) is 137, and between (a) and (c) is 69. The distance between the spectral histograms defined in Chapter 4 between (a) and (b) is 1.288 and between (a) and (c) is 38.5762.	5
2.1 A stable limit cycle for a single relaxation oscillator. The thick solid line represents the limit cycle and thin solid lines stand for nullclines. Arrows are used to indicate the different traveling speed, resulting from fast and slow time scales. The following parameter values are used: $\epsilon = 0.02$, $\beta = 0.1$, $\gamma = 3.0$, and a constant stimulus $I = 1.0$. . .	15
2.2 The temporal activities of the excitatory unit of a single oscillator for different γ values. Other parameters are same as for Figure 2.1. (a) $\gamma = 3.0$. (b) $\gamma = 40.0$	16
2.3 Architecture of a two-dimensional LEGION network with eight-nearest neighbor coupling. An oscillator is indicated by an empty ellipse, and the global inhibitor is indicated by a filled circle.	19

2.4	Illustration of LEGION dynamics. (a) An input image consisting of seven geometric objects, with 40×40 pixels. (b) The corrupted image of (a) by adding 10 which is presented to a 40×40 LEGION network. (c) A snapshot of the network activity at the beginning. (d)-(j) Subsequent snapshots of the network activity. In (c)-(j), the grayness of a pixel is proportional to the corresponding oscillator's activity and black pixels represent oscillators in the active phase. The parameter values for this simulation are following: $\epsilon = 0.02$, $\beta = 0.1$, $\gamma = 20.0$, $\theta_x = -0.5$, $\theta_p = 7.0$, $\theta_z = 0.1$, $\theta = 0.8$, and $W_z = 2.0$	21
2.5	Temporal evolution of the LEGION network. The upper seven plots show the combined temporal activities of the seven oscillator blocks representing the corresponding geometric objects. The eighth plot shows the temporal activities of all the stimulated oscillators which correspond to the background. The bottom one shows the temporal activity of the global inhibitor. The simulation took 20,000 integration steps using a fourth-order Runge-Kutta method to solve differential equations.	22
2.6	Segmentation result of the LEGION network for the range image of a column. (a) The input range image. (b) The background region. (c)-(f) The four segmented regions. (g) The overall segmentation result represented by a gray map. (h) The corresponding intensity image. (i) The 3-D construction model. As in Figure 2.4, black pixels in (b)-(f) represent oscillators that are in the active phase.	28
2.7	Segmentation results of the LEGION network for range images. In each row, the left frame shows the input range image, the middle one shows the segmentation result represented by a gray map, and the right one shows the 3-D construction model for comparison purposes. . . .	30
2.8	Segmentation results of the LEGION network for several more range images. See the caption of Figure 2.7 for arrangement.	31
2.9	Two examples with thin regions. The global inhibition and potential threshold are tuned to get the results shown here. See the caption of Figure 2.7 for arrangement.	32

2.10	A hierarchy obtained from multiscale segmentation. The top is the input range image and each segmented region is further segmented by increasing the level of global inhibition. As in Figure 2.6, black pixels represent active oscillators, corresponding to the popped up region. See Figure 2.6(i) for the corresponding 3-D model.	39
3.1	An example with non-uniform boundary gradients and substantial noise. (a) A noise-free synthetic image. Gray values in the image: 98 for the left ‘[’ region, 138 for the square, 128 for the central oval, and 158 for the right ‘]’ region. (b) A noisy version of (a) with Gaussian noise of $\sigma = 40$. (c) Local gradient map of (b) using the Sobel operators. (d)-(f) Smoothed images from an anisotropic diffusion algorithm [106] at 50, 100, and 1000 iterations. (g)-(i) Corresponding edge maps of (d)-(f) respectively using the Sobel edge detector.	44
3.2	Illustration of the coupling structure of the proposed algorithm. (a) Eight oriented windows and a fully connected window defined on a 3 x 3 neighborhood. (b) A small synthetic image patch of 6 x 8 in pixels. (c) The resulting coupling structure for (b). There is a directed edge from (i_1, j_1) to a neighbor (i_0, j_0) if and only if (i_1, j_1) contributes to the smoothing of (i_0, j_0) according to equations (3.12) and (3.9). Each circle represents a pixel, where the inside color is proportional to the gray value of the corresponding pixel. Ties in (3.9) are broken according to left-right and top-down preference of the oriented windows in (a).	53
3.3	Temporal behavior of the proposed algorithm with respect to the amount of noise. Six noisy images are obtained by adding zero-mean Gaussian noise with σ of 5, 10, 20, 30, 40, and 60, respectively, to the noise-free image shown in Figure 3.1(a). The plot shows the deviation from the ground truth image with respect to iterations of the noise-free image and six noisy images.	56
3.4	Relative variance of the proposed algorithm for the noise-free image shown in Figure 3.1(a) and four noisy images with Gaussian noise of zero-mean and σ of 5, 20, 40 and 60, respectively.	57
3.5	Relative variance of the proposed algorithm for real images shown in Figure 3.9-3.12.	58

3.6	The oriented bar-like windows used throughout this chapter for synthetic and real images. The size of each kernel is approximately 3 x 10 in pixels.	59
3.7	The smoothed images at the 11th iteration and detected boundaries for three synthetic images by adding specified Gaussian noise to the noise-free image shown in Figure 3.1(a). Top row shows the input images, middle the smoothed image at the 11th iteration, and bottom the detected boundaries using the Sobel edge detector. (a) Gaussian noise with $\sigma = 10$. (b) Gaussian noise with $\sigma = 40$. (c) Gaussian noise with $\sigma = 60$	61
3.8	The smoothed image at the 11th iteration and detected boundaries for a synthetic image with corners. (a) Input image. (b) Smoothed image. (c) Boundaries detected.	62
3.9	The smoothed image at the 11th iteration and detected boundaries for a grocery store advertisement. Details are smoothed out while major boundaries and junctions are preserved accurately. (a) Input image. (b) Smoothed image. (c) Boundaries detected.	62
3.10	The smoothed image at the 11th iteration and detected boundaries for a natural satellite image with several land use patterns. The boundaries between different regions are formed from noisy segments due to the coupling structure. (a) Input image. (b) Smoothed image. (c) Boundaries detected.	63
3.11	The smoothed image at the 11th iteration and detected boundaries for a woman image. While the boundaries between large features are preserved and detected, detail features such as facial features are smoothed out. (a) Input image. (b) Smoothed image. (c) Boundaries detected.	64
3.12	The smoothed image at the 11th iteration and detected boundaries for a texture image. The boundaries between different textured regions are formed while details due to textures are smoothed out. (a) Input image. (b) Smoothed image. (c) Boundaries detected.	65

3.13	Deviations from the ground truth image for the four nonlinear smoothing methods. Dashed line: The SUSAN filter [117]; Dotted line: The Perona-Malik model [105]; Dash-dotted line: The Weickert model of edge enhancing anisotropic diffusion [137]; Solid line: The proposed algorithm.	67
3.14	Relative variance of the four nonlinear smoothing methods. Dashed line: The SUSAN filter [117]; Dotted line: The Perona-Malik diffusion model [105]; Dash-dotted line: The Weickert model [137]; Solid line: The proposed algorithm.	68
3.15	Smoothing results and detected boundaries of the four nonlinear methods for a synthetic image shown in Figure 3.7(a). Here noise is not large and all of the methods perform well in preserving boundaries.	70
3.16	Smoothing results and detected boundaries of the four nonlinear methods for a synthetic image with substantial noise shown in Figure 3.7(b). The proposed algorithm generates sharper and better connected boundaries than the other three methods.	72
3.17	Smoothing results and detected boundaries of a natural scene satellite image shown in Figure 3.10. Smoothed image of the proposed algorithm is at the 11th iteration while smoothed images of the other three methods are chosen manually. While the other three methods generate similar fragmented boundaries, the proposed algorithm forms the boundaries between different regions due to its coupling structure.	72
4.1	Basis functions of Fourier transform in time and frequency domains with their Fourier transforms. (a) An impulse and its Fourier transform. (b) A sinusoid function and its Fourier transform.	79
4.2	A texture image with its Gabor filter response. (a) Input texture image. (b) A Gabor filter, which is truncated to save computation. (c) The filter response obtained through convolution.	81
4.3	A texture image and its spectral histograms. (a) Input image. (b) A Gabor filter. (c) The histogram of the filter. (d) Spectral histograms of the image. There are eight filters including intensity filter, gradient filters D_{xx} and D_{yy} , four LoG filters with $T = \sqrt{2}/2, 1, 2,$ and $4,$ and a Gabor filter $Gcos(12, 150^\circ)$. There are 8 bins in the histograms of intensity and gradient filters and 11 bins for the other filters.	84

4.4	Gibbs sampler for texture synthesis.	88
4.5	Texture image synthesis by matching observed statistics. (a) Observed texture image. (b) Initial image. (c) Synthesized image after 14 sweeps. (d) The total matched error with respect to sweeps.	90
4.6	Temporal evolution of a selected filter for texture synthesis. (a) A Gabor filter. (b) The histograms of the Gabor filter. Dotted line - observed histogram, which is covered by the histogram after 14 sweeps; dashed line - initial histogram; dash-dotted line - histogram after 2 sweeps. solid line - histogram after 14 sweeps. (c) The error of the chosen filter with respect to the sweeps. (d) The error between the observed histogram and the synthesized one after 14 sweeps. Here the error is multiplied by 1000.	91
4.7	More texture synthesis examples. Left column shows the observed images and right column shows the synthesized image within 15 sweeps. In (b), due to local minima, there are local regions which are not perceptually similar to the observed image.	92
4.8	Real texture images of regular patterns with synthesized images after 20 sweeps. (a) An image of a leather surface. The total matched error after 20 sweeps is 0.082. (b) An image of a pressed calf leather surface. The total matched error after 20 sweeps is 0.064.	94
4.9	Texture synthesis for an image with different regions. (a) The observed texture image. This image is not a homogeneous texture image and consists mainly of two homogeneous regions. (b) The initial image. (c) Synthesized image after 100 sweeps. Even though the spectral histogram of each filter is matched well, compared to other images, the error is still large. Especially for the intensity filter, the error is still about 7.44 %. The synthesized image is perceptually similar to the observed image except the geometrical relationships among the homogeneous regions. (d) The matched error with respect to the sweeps. Due to that the observed image is not homogeneous, the synthesis algorithm converges slower compared with Figure 4.5(d).	95
4.10	A synthesis example for a synthetic texton image. (a) The original synthetic texton image with size 128×128 . (b) The synthesized image with size 256×256	96

4.11	A synthesis example for an image consisting of two regions. (a) The original synthetic image with size 128×128 , consisting of two intensity regions. (b) The synthesized image with size 256×256	97
4.12	A synthesis example for a face image. (a) Lena image with size 347×334 . (b) The synthesized image with size 256×256	97
4.13	The synthesized images of the 40 texture images shown in Figure 4.16. Here same filters and cooling schedule are used for all the images. . .	98
4.14	Synthesized images from different initial images for the texture image shown in Figure 4.3(a). (a)-(c) Left column is the initial image and right column is the synthesized image after 20 sweeps. (d) The matched error with respect to the number of sweeps.	100
4.15	Synthesized images from Heeger and Bergen's algorithm and the matched spectral histogram error for the image shown in Figure 4.3(a). (a) Synthesized image at 3 iterations. (b) Synthesized image at 10 iterations. (c) Synthesized image at 100 iterations. (d) The L_1 -norm error of the observed spectral histogram and the synthesized one. . .	102
4.16	Forty texture images used in the classification experiments. The input image size is 256×256	103
4.17	The divergence between the feature vector of each image in the texture image database shown in Figure 4.16. (a) The cross-divergence matrix shown in numerical values. (b) The numerical values are displayed as an image.	106
4.18	(a) The classification error for each image in the texture database along with the ratio between the maximum and minimum divergence shown in (b) and (c) respectively. (b) The maximum divergence of spectral histogram from the feature vector of each image. (c) The minimum divergence between each image and the other ones.	107
4.19	The classification error of the texture database with respect to the scale for feature extraction.	108

4.20	(a) Image “Hexholes-2” from the texture database. (b) The classification error rate for the image. (c) The ratio between maximum divergence and minimum cross divergence with respect to scales. . . .	109
4.21	(a) Image “Woolencloth-2” from the texture database. (b) The classification error rate for the image. (c) The ratio between maximum divergence and minimum cross divergence with respect to scales. . . .	110
4.22	(a) A texture image consisting of five texture regions from the texture database. (b) Classification result using spectral histograms. (c) Divergence between spectral histograms and the feature vector of the assigned texture image. (d) The ground truth segmentation of the image. (e) Misclassified pixels, shown in black.	112
4.23	(a) The classification error for each image in the database at integration scale 35×35 . (b) The classification error at different integration scales. In both cases, solid line – training is done using half of the samples; dashed line – training is done using all the samples.	113
4.24	The classification error with respect to the ratio of testing samples to training samples. Solid line – integration scale 35×35 ; dashed line – integration scale 23×23	114
4.25	A group of 10 texture images used in [108]. Each image is 256×256 .	116
4.26	A group of 10 texture images used in [108]. Each image is 256×256 .	117
4.27	Image retrieval result from a 100-image database using a given image patch based on spectral histograms. (a) Input image patch with size 35×35 . (b) The sorted matched error for the 100 images in the database. (c) The first nine image with smallest errors.	119
4.28	Image retrieval result from a 100-image database using a given image patch based on spectral histograms. (a) Input image patch with size 53×53 . (b) The sorted matched error for the 100 images in the database. (c) The first nine image with smallest errors.	120

4.29	Classification error in percentage of texture database for different features. Solid line: spectral histogram of eight filters including intensity, gradients, LoG with two scales and Gabor with three different orientations. Dotted line: Mean value of the image patch. Dashed line: Weighted sum of mean and variance values of the image patch. The weights are determined to achieve the best result for window size 35×35 . Dash-dotted line: Intensity histogram of image patches. . . .	122
4.30	Classification error in percentage of the texture database for different filters. Solid line: spectral histogram of eight filters including intensity, gradients, LoG with two scales and Gabor with three different orientations. Dotted line: Gradient filters D_{xx} and D_{yy} ; Dashed line: Laplacian of Gaussian filters $LoG(\sqrt{2}/2)$, $LoG(1)$, and $LoG(2)$. Dash-dotted line: Six Cosine Gabor filters with $T = 4$ and six orientations $\theta = 0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ$, and 150°	123
4.31	Classification error in percentage of the texture database for different distance measures. Solid line: χ^2 -square statistic. Dotted line: L_1 -norm. Dashed line: L_2 -norm. Dash-dotted line: Kullback-Leibler divergence.	124
4.32	Ten synthetic texture pairs scanned from Malik and Perona [87]. The size is 136×136	125
4.33	The averaged texture gradient for selected texture pairs. (a) The texture pair (+ O) as shown in Figure 4.32. (b) The texture gradient averaged along each column for (a). The horizontal axis is the column number and the vertical axis is the gradient. (c) The texture pair (R-mirror-R). (d) The averaged texture gradient for (c).	126
4.34	Comparison of texture discrimination measures. Dashed line - Psychophysical data from Krose [69]; dotted line - Prediction of Malik and Perona's model [87]; solid line - prediction of the proposed model based on spectral histograms.	128
5.1	Gray-level image with two regions with similar means but different variances.	136
5.2	Examples of asymmetric windows. The solid cross is the central pixel. (a) Square windows. (b) Circular windows.	138

5.3	<p>Gray-level image segmentation using spectral histograms. The integration scale $W^{(s)}$ for spectral histograms is a 15×15 square window, $\lambda_R = 0.2$, and $\lambda_B = 3$. Two features are given at $(32, 64)$ and $(96, 64)$. (a) A synthetic image with size 128×128. The image is generated by adding zero-mean Gaussian noise with different σ's at left and right regions. (b) Initial classification result. (c) Final segmentation result. The segmentation error is 0.00 % and all the pixels are segmented correctly.</p>	141
5.4	<p>The histogram and derived probability model of χ^2-statistic for the given region features. Solid lines stand for left region and dashed lines stand for right region. (a) The histogram of the χ^2-statistic between the given feature and the computed ones at a coarser grid. (b) The derived probability model for the left and right regions.</p>	142
5.5	<p>A row from the image shown in Figure 5.3 and the result using derived probability model. In (b) and (c), solid lines stand for left region and dashed lines stand for right region. (a) The 64th row from the image. (b) The probability of the two given regional features using asymmetric windows when estimating spectral histogram. The edge point is correctly located between columns 64 and 65. (c) Similar to (a) but using windows centered at the pixel to compute spectral histogram. Labels between columns 58 and 65 cannot be decided. This is because that the computed spectral histograms within that interval do not belong to either region.</p>	143
5.6	<p>Classification result based on χ^2-statistic for the row shown in Figure 5.4(a). Solid lines stand for left region and dashed lines stand for right region. (a) χ^2-statistic from the two given regional features using asymmetric windows when estimating spectral histogram. If we use the minimum distance classifier, the edge point will be located between columns 65 and 66, where the true edge point should be between columns 64 and 65. (b) Similar to (b) but using windows centered at the pixel to compute spectral histogram. The edge point is localized between 61 and 62.</p>	144

5.7	Gray-level image segmentation using spectral histograms. $W^{(s)}$ is a 15×15 square window, $\lambda_\Gamma = 0.2$, and $\lambda_B = 5$. Two features are given at $(32, 64)$ and $(96, 45)$. (a) A synthetic image with size 128×128 . The image is generated by adding zero-mean Gaussian noise with different σ 's at the two different regions. Here the boundary is 'S' shaped to test the segmentation algorithm in preserving boundaries. (b) Initial classification result. (c) Final segmentation result.	145
5.8	Texture image segmentation using spectral histograms. $W^{(s)}$ is a 29×29 square window, $\lambda_\Gamma = 0.2$, and $\lambda_B = 2$. Features are given at pixels $(32, 32)$ and $(96, 32)$. (a) A texture image consisting of two texture regions with size 128×64 . (b) Initial classification result. (c) Final segmentation result.	145
5.9	Texture image segmentation using spectral histograms. $W^{(s)}$ is a 29×29 square window, $\lambda_\Gamma = 0.2$, and $\lambda_B = 3$. (a) A texture image consisting of two texture regions with size 128×64 . (b) Initial classification result. (c) Final segmentation result.	146
5.10	Texture image segmentation using spectral histograms. $W^{(s)}$ is a 35×35 square window, $\lambda_\Gamma = 0.4$, and $\lambda_B = 3$. Four features are given at $(32, 32)$, $(32, 96)$, $(96, 32)$, and $(96, 96)$. (a) A texture image consisting of four texture regions with size 128×128 . (b) Initial classification result. (c) Final segmentation result.	147
5.11	Texture image segmentation using spectral histograms. $W^{(s)}$ is a 35×35 square window, $\lambda_\Gamma = 0.4$, and $\lambda_B = 3$. Four features are given at $(32, 32)$, $(32, 96)$, $(96, 32)$, and $(96, 96)$. (a) A texture image consisting of four texture regions with size 128×128 . (b) Initial classification result. (c) Final segmentation result.	147
5.12	Texture image segmentation using spectral histograms. $W^{(s)}$ is a 29×29 square window, $\lambda_\Gamma = 0.2$, and $\lambda_B = 3$. Four features are given at $(32, 32)$, $(32, 96)$, $(96, 32)$, and $(96, 96)$. (a) A texture image consisting of four texture regions with size 128×128 . (b) Initial classification result. (c) Final segmentation result.	148

5.13	Texture image segmentation using spectral histograms. $W^{(s)}$ is a 35×35 square window, $\lambda_\Gamma = 0.4$, and $\lambda_B = 3$. Four features are given at $(32, 32)$, $(32, 96)$, $(96, 32)$, and $(96, 96)$. (a) A texture image consisting of four texture regions with size 128×128 . (b) Initial classification result. (c) Final segmentation result.	148
5.14	A challenging example for texture image segmentation. $W^{(s)}$ is a 35×35 square window, $\lambda_\Gamma = 0.4$, and $\lambda_B = 20$. Two features are given at $(160, 160)$ and $(252, 250)$. (a) Input image consisting of two texture images, where the boundary can not be localized clearly because of their similarity. The size of the image is 320×320 in pixels. (b) Initial classification result. (c) Final segmentation result.	149
5.15	Another challenging example for texture segmentation. $W^{(s)}$ is a 35×35 square window, $\lambda_\Gamma = 0.4$, and $\lambda_B = 20$. Two features are given at $(160, 160)$ and $(252, 250)$. (a) Input image consisting of two texture images, where the boundary can not be localized clearly because of their similarity. The size of the image is 320×320 in pixels. (b) Initial classification result. (c) Final segmentation result.	149
5.16	Segmentation for a texton image with oriented short lines. $W^{(s)}$ is a 35×35 square window, $\lambda_\Gamma = 0.4$, and $\lambda_B = 10$. Two features are given at $(185, 67)$ and $(180, 224)$. (a) The input image with size of 402×302 in pixels. (b) The initial classification result. (c) The segmentation result using spectral histograms. (d) The initial classification result using two Gabor filters $Gcos(10, 30^\circ)$ and $Gcos(10, 60^\circ)$. (e) The segmentation result using two Gabor filters. The result is improved significantly. . .	150
5.17	Segmentation results at different integration scales. Parameters $\lambda_\Gamma = 0.4$, and $\lambda_B = 4$ are fixed. (a) The input image. (b) The percentage of mis-classified pixels.	151
5.18	Segmentation results using different segmentation scales for the image shown in Figure 5.17(a). In each sub-figure, the left shows the initial classification result and the right shows the segmentation result. Parameters $\lambda_\Gamma = 0.4$, and $\lambda_B = 4$ are fixed. (a) $W^{(s)}$ is a 1×1 square window. (b) $W^{(s)}$ is a 3×3 square window. (c) $W^{(s)}$ is a 5×5 square window. (d) $W^{(s)}$ is a 7×7 square window.	152

5.19	A texture image with a cheetah. The feature vector is calculated at pixel (247, 129) at scale 19×19 , $\lambda_\Gamma = 0.2$, and $\lambda_B = 2.5$. To demonstrate the accuracy of the results, the classification and segmentation results are embedded into the original image by lowering the intensity values of the background region by a factor of 2. (a) The input image with size 324×486 . (b) The initial classification result using 8 filters. (c) The final segmentation result using 8 filters. (d) The initial classification result using 6 filters consisting of D_{xx} , D_{yy} , $LoG(\sqrt{2}/2)$, $LoG(1)$, $LoG(2)$ and $LoG(3)$. (e) The final segmentation result corresponding to (d).	154
5.20	An indoor image with a sofa. The feature vector is calculated at pixel (146, 169) at scale 35×35 , $\lambda_\Gamma = 0.2$, and $\lambda_B = 3$. (a) Input image with size 512×512 . (b) Initial classification result. (c) Final segmentation result. (d) Segmentation result if we assume there is another region feature given at (223, 38).	155
5.21	Texture image segmentation with representative pixels identified automatically. $W^{(s)}$ is a 29×29 square window, $W^{(a)}$ is a 35×35 square window, $\lambda_C = 0.1$, $\lambda_A = 0.2$, $\lambda_B = 2.0$, $\lambda_\Gamma = 0.2$, and $T_A = 0.08$. (a) Input texture image, which is shown in Figure 5.8. (b) Initial classification result. Here the representative pixels are detected automatically. (c) Final segmentation result.	158
5.22	Texture image segmentation with representative pixels identified automatically. $W^{(s)}$ is a 29×29 square window, $W^{(a)}$ is a 43×43 square window, $\lambda_C = 0.4$, $\lambda_A = 0.4$, $\lambda_B = 5.0$, $\lambda_\Gamma = 0.4$, and $T_A = 0.30$. (a) Input texture image, which is shown in Figure 5.10. (b) Initial classification result. Here the representative pixels are detected automatically. (c) Final segmentation result.	158
5.23	Texture image segmentation with representative pixels identified automatically. $W^{(s)}$ is a 29×29 square window, $W^{(a)}$ is a 43×43 square window, $\lambda_C = 0.1$, $\lambda_A = 0.2$, $\lambda_B = 5.0$, $\lambda_\Gamma = 0.4$, and $T_A = 0.20$. (a) Input texture image, which is shown in Figure 5.11. (b) Initial classification result. Here the representative pixels are detected automatically. (c) Final segmentation result.	159

5.24	Texture image segmentation with representative pixels identified automatically. (a) Input texture image, which is shown in Figure 5.12. (b) Initial classification result. Here the representative pixels are detected automatically. (c) Final segmentation result.	159
5.25	Texture image segmentation with representative pixels identified automatically. $W^{(s)}$ is a 29×29 square window, $W^{(a)}$ is a 43×43 square window, $\lambda_C = 0.1$, $\lambda_A = 0.2$, $\lambda_B = 5.0$, $\lambda_\Gamma = 0.4$, and $T_A = 0.20$. (a) Input texture image, which is shown in Figure 5.13. Here the representative pixels are detected automatically. (c) Final segmentation result.	160
5.26	(a) A texture image with size 256×256 . (b) The segmentation result using spectral histograms. (c) Wrongly segmented pixels of (b), represented in black with respect to the ground truth. The segmentation error is 6.55%. (d) Refined segmentation result. (e) Wrongly segmentation pixels of (d), represented in black as in (c). The segmentation error is 0.95%.	161
5.27	(a) A synthetic image with size 128×128 , as shown in Figure 5.7(a). (b) The segmentation result using spectral histograms as shown in Figure 5.7(c). (c) Refined segmentation result.	163
5.28	(a) A texture image with size 256×256 . (b) The segmentation result using spectral histograms. (c) Refined segmentation result.	163
5.29	(a) A texture image with size 256×256 . (b) The segmentation result using spectral histograms. (c) Refined segmentation result.	164
5.30	Distance between scales for different regions. (a) Input image. (b) The distance between different integration scales for the left region at pixel (32, 64). (c) The distance between different integration scales for the right region at pixel (96, 64).	165
5.31	A natural image with a zebra. $\lambda_\Gamma = 0.2$, and $\lambda_B = 5.5$. (a) The input image. (b) The segmentation result with one feature computed at (205, 279). (c) The segmentation result with one feature computed at (308, 298). (d) The combined result from (b) and (c).	167

6.1	On- and off-center cell responses. (a) Input image. (b) On-center cell responses. (c) Off-center cell responses (d) Binarized on- and off-center cell responses. White regions represent on-center response regions and black off-center regions.	173
6.2	The figure-ground segregation network architecture for Figure 6.1(a). Nodes 1, 2, 3 and 4 belong to the white region; Nodes 5, 6, 7, and 8 belong to the black region; Nodes 9 and 10, 11 and 12 belong to the left and right gray regions respectively. Solid lines represent excitatory coupling while dashed lines represent inhibitory connections.	174
6.3	Temporal behavior of each node in the network shown in Figure 6.2. Each plot shows the status of the node with respect to the time. The dashed line is 0.5.	178
6.4	Surface completion results for Figure 6.1(a). (a) White region. (b) Gray region. (c) Black region.	180
6.5	Layered representation of surface completion for results shown in Figure 6.4.	180
6.6	Images with virtual contours. (a) Kanizsa triangle. (b) Woven square. (c) Double kanizsa.	181
6.7	Surface completion results for the corresponding image in Figure 6.6.	182
6.8	Images with virtual contours. (a) Kanizsa triangle. (b) Four crosses. (c) Overlapping rectangular bars.	183
6.9	Surface completion results for the corresponding image in Figure 6.8.	183
6.10	Images with virtual contours. (a) Original pacman image. (b) Mixed pacman image. (c) Alternate pacman image.	183
6.11	Layered representation of surface completion for the corresponding images shown in Figure 6.10.	184
6.12	Bregman and real images. (a) and (b) Examples by Bregman [9]. (c) A grocery store image.	185
6.13	Surface completion results for images shown in Figure 6.12.	185

6.14	Bistable perception. (a) Face-vase input image. (b) Faces as figures. (c) Vase as figure.	186
6.15	Temporal behavior of the system for Figure 6.14(a). Dotted lines are 0.5.	187
7.1	Classification result of a noisy synthetic image using a three-layer perceptron. (a) The input image with size of 230×240 . (b) The ground truth image. (c) Positive and negative training samples. Positive examples are shown as white and negative ones as black. (d) Classification result from a three-layer perceptron.	190
7.2	Lateral connection evolution through weight adaptation illustrated using the 170th row from the image shown in Figure 7.1(a). (a) The original signal. (b) Initial connection weights. (c) Connection weights after 40 iterations. (d) Corresponding smoothed signal.	194
7.3	Architecture and local features for the seed selection neural network.	202
7.4	Segmentation result using the proposed method for a synthetic image. (a) A synthetic image as shown in Figure 7.1(a). (b) The segmentation result from the proposed method. Here $W_z = 0.25$ and $\theta_p = 100$	204
7.5	A DOQQ image with size of 6204×7676 pixels of the Washington East, D.C.-Maryland area.	207
7.6	Seed pixels obtained by applying a trained three-layer perceptron to the DOQQ image shown in Figure 7.5. Seed pixels are marked as white and superimposed on the original image. The network is trained using 19 positive and 28 negative samples, where each sample is a 31×31 window.	208
7.7	Extracted hydrographic regions from the DOQQ image shown in Figure 7.5. Hydrographic regions are marked as white and superimposed on the original image to show the accuracy of the extracted result. Here $W_z = 0.15$ and $\theta_p = 4000$	209
7.8	A ground truth generated by manually placing seeds based on the corresponding 1:24,000 USGS topographic map and DOQQ image. The result was manually edited.	210

7.9	Hydrographic region extraction result for an aquatic garden area with manually placed seed pixels. Due that no reliable seed region is detected, this aquatic region, which is very similar to soil regions, is not extracted from the DOQQ image as shown in Figure 7.7. Extracted regions are marked as white and superimposed on the original image.	211
7.10	Extraction result for an image patch from Figure 7.5. (a) The input image. (b) The seed points from the neural network. (c) A topographic map of the area. Here the map is scanned from the chapter version and not wrapped with respect to the image. (d) Extracted result from the proposed method. Extracted regions are represented by white and superimposed on the original image.	213
7.11	A DOQQ image with size of 5802×7560 pixels of Damascus, Pennsylvania-New York area.	215
7.12	Extracted hydrographic regions from the DOQQ image shown in Figure 7.11. The extracted regions are represented by white pixels and superimposed on the original image.	216
7.13	A ground truth generated based on a 1:24,000 USGS topographic map and DOQQ image.	217
8.1	A stereo image pair and correspondence using the spectral histogram. (a) The left image. (b) The right image. (c)-(e) The matching results of marked pixels in the left image. In each row, the left shows the marked pixel, the middle shows the probability of being a match in the paired image, and the right shows the high probability area in the paired image.	224
8.2	Comparison between an edge detector and the spectral histogram using a natural image of a giraffe. (a) The input image with size 300×240 . (b) The edge map from a Canny edge detector [13]. (c) The initial classification result using the method presented in Chapter 5. A spectral histogram is extracted at pixel (209, 291) and the segmentation scale is 29×29 . (d) The initial classification is embedded in the input image to show the boundaries.	226