2000

Image analysis using multiscale boundary extraction algorithm

Nawapak Eua-Anant
Iowa State University

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Image analysis using multiscale boundary extraction algorithm

by

Nawapak Eua-Anant

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Electrical Engineering (Communications and Signal Processing)
Major Professor: Lalita Udpa

Iowa State University
Ames, Iowa
2000

Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of
Nawapak Eua-Anant
has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.
Committee Member
Signature was redacted for privacy.
Committee Member
Signature was redacted for privacy.
Committee Member
Signature was redacted for privacy.
Committee Member
Signature was redacted for privacy.
Major Professor
Signature was redacted for privacy.
For the Major Program
Signature was redacted for privacy.
For the Graduate College

DEDICATION

I would like to dedicate this dissertation to my friends and family, to all the great teachers and scientists who have enlightened the world, and to peace, liberty, and harmony in the world.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The complete analysis and interpretation of the information in image data is a complex process. This dissertation presents 3 major contributions to image analysis, namely, global multiscale detection, local scale analysis, and boundary extraction. Global scale analysis is related to identification of the various scales presented in the image. A new approach for global scale analysis is developed based on the differential power spectrum normalized variance ratio (DPSNVR). The DPSNVR is the ratio of the second order normalized central moment of the power spectrum of the image to that of the multiscale differential mask. Local maxima in DPSNVR graph directly indicate the global scales in the image. Local scale analysis performs a more detailed analysis of the edges to eliminate effects of blurring. A method based on mutilscale feature matching has been proposed. Details obtained at all scales are treated using a scale invariant normalization scheme. Besides local scale analysis, a multiscale data fusion algorithm has been implemented which leads to the new concept of multiple scale differential masks. The multiple scale differential mask generated using a range of scale values possesses the remarkable shape preservation property which makes it superior to traditional multiscale masks. Finally the complete sequential boundary extraction algorithm based on particle motion in a velocity field is presented. The boundary extraction algorithm incorporates edge localization, boundary representation, and automated selection of boundary extraction parameters. The global scale analysis techniques in conjunction with the boundary extraction algorithm provide a multiscale image segmentation algorithm.

CHAPTER 1. INTRODUCTION

Since the invention of digital computers, the world has changed unimaginably. Digital computers linking mathematical theories to real world applications have opened up a new era of science and technology. The powerful computational capability of today's computers enables many highly complex tasks that were once performed by human operators. One of the rapidly growing areas, digital image processing, is a result of a marriage between multi-dimensional signal processing theory and digital computers. Digital imagery, in its general form, refers to a discretely sampled version of a continuous multidimensional scene and is typically represented by an array of discrete numbers. Digital image processing ranges from low level processes such as image acquisition and image enhancement to highly sophisticated tasks such as object recognition and scene interpretation similar to those performed by a human brain. Today, digital image processing has been employed in a variety of applications ranging from movie production to deep space exploration.

An important area of digital image processing involves analysis of pictorial information contained in image data. Image analysis basically consists of three steps: low-level image acquisition and enhancement, intermediate-level image representation, and high-level image interpretation. The low level processing functions as a front-end of a visual system including sensing and preprocessing that supplies images in suitable formats. The intermediate level processing, involving feature extraction and representation, retrieves and converts important features from an image into explicitly organized data structures. This step drastically reduces amount of data to be analyzed while retaining useful structural information for further analysis. Finally, the extracted information is used in conjunction with a knowledge-based system for less explicit but more abstract high-level scene interpretation. The final process involves the labeling, recognition, storage, interpretation, and classification of extracted objects. In terms of sophistication, this task is analogous to the recognition and interpretation functions of a human brain.

In general, image analysis uses differential structures such as corners, lines, and edges since these structures are natural representation of pictorial information. For instance,

edges separating objects from background determine object location and extension while corners determine the shape of an object. Differential structures are constructed using differential operators and methods of differential geometry. Among these differential structures, edges are the most important since objects can be concisely described by a sequence of edges. In fact, biological visual systems are known to exploit edge information. Processes such as edge detection and boundary extraction that provide structured edge information are essential for image analysis. Nevertheless, since an image usually comprises multiscale structures, feature extraction has to handle the problem of scale. Objects in an image may have different meanings depending upon scale of observation and thus analysis of an image inevitably requires scale of interest to be specified. Without prior knowledge of scale of observation, the problem of scale selection increases the complexity of image analysis. Therefore image analysis essentially requires not only processing methods for enhancing image data but also a strategy for identifying scales of objects in the image. This dissertation focuses primarily on the problem of scale, multiscale differential operators, edges, and boundary extraction algorithm, all of which are crucial in image analysis.

In the previous work [22], multiscale differential operators based on image moment vectors and boundary extraction algorithm based on particle motion in a force field were developed. The boundary extraction algorithm in conjunction with the edge operators that possess size adjustable features is capable of providing boundary information with sub-pixel resolution at a corresponding scale if the scale parameter is selected previously. The global scale analysis method based on statistical characteristics of edges in the scale domain was proposed for scale detection and selection. However, this strategy lacks the development of a theory for the analysis of global and local scales. In addition, the previous boundary extraction algorithm was performed at a single scale in each pass and required several user input parameters.

In this dissertation, the earlier work has been extended to obtain complete multiscale image analysis. General theories for global and local scale analysis and a general scheme for boundary extraction are also developed. Major contributions in this dissertation are 1) a mathematical scheme for the analysis of global scales, 2) a mathematical scheme for the analysis of local scales and the generalized multiple scale differential operator, and 3) the generalized boundary extraction algorithm based on particle motion in a velocity field and implementation of 2-dimensional sequential boundary extraction algorithm. The concept of global scale analysis based on the differential power spectrum normalized variance ratio is introduced. The behavior of edges in scale-frequency domain reveals

information of existing global scales embedded in an image. In contrast to global scale analysis, local scale analysis is based on feature matching of regional properties of an image and is performed in scale-space domain. Local scale analysis leads to the novel concept of multiple scale differential masks that extract features of multiple scales simultaneously. By preserving structures that survive over a range of scales, the new operator eliminates the problem of shape distortion introduced by large size differential operators. At the same time, spurious details that do not survive over a range of scales are suppressed. The use of multiple scale differential masks significantly simplifies the selection of scale for image analysis and hence reduces computational effort. For boundary extraction, the generalized algorithm developed from the previous model based on particle motion in a velocity field is improved by incorporating a new edge localization feature. The method can be used in conjunction with a novel multiple scale differential operators to increase the scale selection tolerance. Practical considerations are taken into account in developing a stable boundary extraction algorithm and suitable data structures for boundary representation. Several termination criteria that enable the algorithm to run smoothly are implemented. In addition, compacted boundary representation based on image features such as corners is also included. In summary, a complete framework for multiscale image analysis is established.

The rest of this dissertation is organized as follows. In Chapter 2, a literature review of previous work related to differential structures, scale-space analysis, edge detection, image segmentation, and multiscale image analysis are presented. Chapter 3 describes several techniques for global scale analysis. The concept of feature matching in local scale analysis and the new multiple scale differential operators are introduced in Chapter 4. The generalized boundary extraction algorithm based on particle motion in a velocity field and its practical implementation are presented in Chapter 5. Concluding remarks are given in the final chapter.

CHAPTER 2. REVIEW OF LITERATURE

2.1 Differential Structures and Multiscale Differential Operators

Differential structures of an image play an important role in image analysis. For example, the zero order differential structure of an image is the image itself, the subject of image analysis. The gradient and curvature of an image are the first and second order differential structures of an image respectively. Since areas of high gradient magnitude correspond to object edges while local extrema of image curvature correspond to object corners, the significance of gradient and curvature as natural descriptors of edges and corners is obvious. Derivatives of an image with respect to spatial position provide crucial information of differential structures associated with spatial positions. Such information is useful in image analysis to indicate where to look for features embedded in an image. Interestingly, biological visual systems also exploit differential structures, especially edges, since typical receptive fields found in animal eyes act as differential operators [46]. In fact, differential structures are natural descriptors of image structures.

Differential operators are fundamental tools in the analysis and synthesis of differential structures. Digital image analysis algorithms rely on primitive difference operators of small window sizes [10, 27, 31, 39, 40, 42, 69, 72] that mimic mathematical differential operators by utilizing the smallest neighborhoods available in a discrete image. For instance, the Sobel, Robert, and Prewitt opeators are implemented on masks of size 3×3 pixels [31, 39, 40]. Although, implementation of these operators is intuitive and simple, it overlooks the fundamental problem of physical observables, the problem of scale. In an image, all objects are associated with scales. Objects in an image may have different meanings depending upon scale of observation. Objects of large scales may comprise objects of smaller scales or individual objects of small scales may be considered meaningless at larger scales. Hence, in image analysis, all operations require scales of interest to be known. Differential operators using fixed size difference masks do not posses the flexibility to handle multiscale feature extraction problems. Without taking

into account the scale of objects. operators working at fixed small scales are vulnerable to noise and ineffective in capturing image structures at arbitrary scales.

Marr and Hildreth [62] introduced two fundamental concepts of multiscale image analysis. First. in natural images. intensity changes occur over a wide range of scales. To deal separately with each scale. it is necessary to filter the image in order to reduce the range of scales over which intensity changes take place. Second. it is essential to utilize pictorial information at several scales. Several attempts were made to develop multiscale differential operators using larger. smoother. and size adjustable difference masks capable of rendering image derivatives at different scales with better noise suppression ability. For example. Rosenfeld and Thurston [71] introduced the solution to multiscale edge detection problems based on variable size difference operators called the difference of boxes (DOB) which is the difference between the average of pixels in a pair of non overlapping variable size square neighborhoods. Macleod [60] and Argyle [2] suggested the use of smoother differential operators for edge detection in noisy images. Macleoud's operator approximates the first derivative of the two-dimensional Gaussian function using the difference of two displaced two-dimensional Gaussian functions while Argyle suggested the use of a one-dimensional split Gaussian function. Other multiscale differential operators include moment based edge operators [23. 57. 58. 59. 68. 78] and multiscale Laplacian operators [61. 62]. Each technique differs from the others only in the order and configuration of the masks. Canny [9] proposed 3 criteria for evaluating the performance of multiscale differential operators: *i*) detection by maximizing output signal to noise ratio, *ii*) localization by minimizing the root-mean-square distance from detected edges to true edges. and *iii*) the multiple response constraint for reducing the number of spurious extrema. An efficient approximation of Canny's operator is the sampled version of the first derivative of a two-dimensional Gaussian function or the Gaussian differential filter [26]. Because derivatives of a Gaussian function possess many good characteristics including causality with respect to resolution and can be computed with less effort. they have been widely used in many applications.

In general. multiscale differential operators have the capability to render differential structures at any desired resolution. These techniques have improved noise suppression performance with a trade off between the accuracy of edge localization and level of noise suppression. This presents the problem of scale selection where one must compromise between maximizing signal strength at a particular scale while minimizing signal strength at other scales. The success or failure of image analysis relies strongly on the choice of scale parameters used in multiscale masks.

2.2 Scale-Space Theory and Scale Analysis

The mathematical framework for the analysis and selection of scales appears in scale-space literature [4, 26, 43, 48, 49, 75, 79, 82]. The scale-space theory is based on the concept that measurement and interpretation of the physical world cannot be completed without specifying the scale of an object. The significant difference between observation made in the real world and a pure mathematical concept is that the mathematical definition of a point having an infinitesimal width and the smoothness constraint of the underlying function cannot be achieved in real world observations. In other words, mathematical rules are dimensionless and scale independent. Witkin [82] introduced the term "scale-space" representation as a result of embedding an image in a scale-parameter family of smoothed images. A scale-space image is obtained by convolving the image with Gaussian kernels over a continuum of scales. Koenderink [43] pointed that convolving the Gaussian kernel with the image is equivalent to the solution of the isotropic diffusion process with the original image as an initial condition. This process prevents spurious structures to be created after resolution is diminished. Based on the Gaussian scale-space concept, Florack *et at* [26] and ter Harr Romeny *et al* [79] introduced the concept of scale invariance and formulated the framework for analysis and synthesis of scale-space differential structures in images. In fact, the Gaussian kernel is a natural representation of a point in scale-space representation of the physical world.

One fundamental assumption of scale-space analysis is causality with respect to resolution: no spurious details at the coarser level of resolution should be created as the scale increases [80]. Babaud *et al* [4] proved that the Gaussian function is the only kernel that possesses this property. In fact, the properties of the Gaussian convolution can be summarized as follows:

1. **Causality:** the Gaussian convolution is causal in the sense that the numbers of local extrema as well as zero crossing paths of diffused images are less than or equal to those of the original image [4, 83].

2. **Continuous semi-group structure:** the convolution of two Gaussian kernels results in the Gaussian kernel of the same scale-parameter family.

3. **Isotropy, linearity, and shift invariance:** the Gaussian smoothing operator is isotropic, linear, and shift invariant.

4. **Simplicity:** the Gaussian function is differentiable and its Fourier transform is simple enabling the direct computation of higher order differential operators.

Lindenberg [48] derived the scale-space theory for discrete signals where the Gaussian kernels are replaced by the kernels derived from the modified Bessel functions. The modified kernels assure that the causality property is not violated as a result of signal discretization.

Once multiscale information is obtained, schemes for data reduction and extraction of significant structures from scale-space representation are necessary. The solutions of multiscale data utilization do not come directly from scale-space theory but from the behavior of image structures in scale-space such as image gradient [5], maximum gradient paths [34], zero crossings of Laplacian images [32, 56], intensity extrema [47], and gradient watersheds [28, 29, 67]. In these techniques, exhaustive searches, tracking schemes that hierarchically links structures of interest at different scales together, either from coarse scales to fine scales or from fine scales to coarse scales, and techniques for handling structures of different topologies at different scales are employed.

To extract image features of different size efficiently, scale parameters of kernels need to match desired structures. In the early use of multiscale differential masks, without prior knowledge of scale, selection of scales was done in an *ad hoc* fashion. Modeling structures of different sizes as diffused versions of an ideal structure with different degrees of blur, Lindenberg [50, 51, 52, 53] proposed a methodology for selecting scale parameters to match local structure sizes by maximizing the normalized measure of local structure strength. Automatic local scale detection for image structures including grey-level blobs, corners, and junctions [50, 53], edges and ridges [51, 52], distances between ridges [1], and lines [55] have been proposed. In addition to local scale analysis, other techniques including blur estimation [20, 84] and estimation of depths from focus [17] and defocus [76, 77] are closely related to local scale analysis. In these techniques, degrees of blur of local details are estimated in terms of Gaussian spread constants. In [17, 76, 77] this information is further converted into depth information. Besides local scale analysis, the scheme for global scale detection based on global properties of edges in scale-space has been proposed by Eua-Anant [22] and Eua-Anant and Udpa [25]. Local minima of edge-characteristic-scale curves are used as the guidance for global scale detection. However, the method is not versatile and is sensitive to noise.

2.3 Image Segmentation

Image segmentation plays a fundamental role in image analysis and computer vision. Typically, image analysis applications employ image segmentation in the early stage

to partition the image into constituent parts. Image segmentation can be broadly divided into 3 classes, namely pixel-oriented, region-oriented, and edge-oriented techniques [31, 37, 40, 74]. Pixel-oriented techniques mainly rely on global measurement profiles such as pixel intensity histogram while local connectivity information is discarded. Consequently these methods perform poorly in the presence of noise and intensity variations. Also, these methods usually require prior knowledge of visual attributes of regions to be segmented. On the other hand, region-oriented techniques such as region growing and region splitting and merging are dynamic algorithms that link local pixels into segmented regions by considering both similarity and connectivity between pixels in local areas. Based on statistical and structural approaches, these methods are more robust in the presence of noise and yield more promising results at the expense of higher computational effort to perform massive parallel searches in the regions of interest. Besides pixel-oriented and region-oriented techniques that focus on extracting regions of similar attributes, edge-oriented techniques emphasize the extraction of edges or boundaries between regions by exploiting information of discontinuity between the regions. The major advantage of boundary extraction over region extraction is the reduction in storage and computational cost, whereas the disadvantage of boundary extraction is the possibility of getting incomplete boundaries or missing fuzzy boundaries.

2.3.1 Edge-Oriented Image Segmentation

The first step in most edge-oriented image segmentation techniques is to enhance edges at desired scales using edge operators while reducing noise by means of filtering. Generally, in a monochrome image, edges are associated with abrupt changes of intensity distribution and edge operators are normally employed to detect the presence of these transitions. Such changes can be detected in the form of extrema and zero crossings of image derivatives. Due to the fact that many edges found in real images are approximation to step edges, the most commonly used edge operators are the first and second order differential operators.

2.3.1.1 Edge detection

Based behavior of derivatives of step edges, edge detection by means of thresholding image gradient magnitude or detecting zero crossings of image Laplacian is simple and straight forward [31, 40, 46, 62, 74]. In the gradient thresholding methods, pixels having gradient magnitude above the threshold level are considered as edge pixels while in the Laplacian-based methods, edge elements represent borders between pairs of pixels where

changes in signs of image Laplacian occur. Nevertheless, since differential operators functioning as a high pass filter enhance not only edges but also noise, these methods are sensitive to noise. In addition, since gradient thresholding methods mainly rely on gradient magnitude, slight variations in gradient magnitude especially at corners and junctions where the gradient field is weak often yield discontinuities in the results. In the case of blurred images, direct gradient thresholding without utilizing other information often produces unacceptable thick edges. For the Laplacian based methods, although the location of a step edge coincides with the zero crossing of the Laplacian, the converse is not always true and thus zero crossing detection often yields false boundary indications. Junctions also present a problem for the Laplacian-based edge detection methods. In areas where 3 or more regions meet, the assumption that different signs of Laplacian occupy different regions is invalid and thus detection of zero crossings gives rise to boundary localization errors.

The next step in edge-oriented image segmentation is to estimate boundaries from edge information. Among boundary estimation techniques, two approaches have gained significant popularity, namely template matching and edge linking. In template matching, the estimated boundary is modeled as a deformable curve interacting with the image dynamically in order to maximize the likelihood between the model and image data. In contrast to template matching, in edge linking, the boundary is sequentially extracted along the object edges by mean of a driving force such as connectivity of edge pixels. Compared to edge-linking techniques, boundary extraction techniques using deformable contour models yield superior performance: more robust, better boundary localization, and higher quality of extracted results while edge linking techniques are simpler, faster, and less sensitive to topologies of image structures.

2.3.1.2 Edge linking techniques

Following edge detection, edge linking techniques are used to assemble the edge elements into ordered sequences and filter out spurious edges. Early techniques for edge linking and sequential boundary extraction include heuristic graph searches [3, 63, 64] and dynamic programming [21, 65]. The process begins with an initial edge element, sequentially examines neighboring edge elements and gathers next adjacent edge elements satisfying edge linking conditions to form the boundary. The extracted boundaries are gradually extended in the direction that minimizes the cost functions of the boundaries [3, 21, 63, 64, 65]. Since in most techniques, 4 or 8 possible linking directions are considered for each step, the finest details of extracted boundaries are

limited by pixels, pixel grids, and pixel diagonals. These techniques do not make use of multiscale differential masks but employ local operators of small windows.

Direct use of image derivatives for boundary extraction can be found in the gradient based methods [8, 14, 33, 34, 66] and Laplacian based methods [32, 36, 38, 62]. In gradient based methods, gradient magnitude is used as indication of edge strength. Edge pixels having the same gradient direction and large enough gradient magnitude will be linked together with the current boundary. The boundary is grown approximately in the direction perpendicular to the gradient direction. In the case of broad edges, a thinning process is employed to erode edge curves. Nevertheless, the thinned edge pixels may be displaced from actual edge locations due to loss of information of edge strength during thresholding. In contrast to the gradient based methods, there is no need for edge thinning in the methods based on zero crossing detection. However, since not all zero crossings of image Laplacian correspond to edges, the methods can result in some false boundaries.

The algorithm proposed by Murray and O'Malley [66] performs edge enhancement, thresholding of the gradient image, and edge thinning. Instead of thresholding the gradient image to obtain an edge map directly, alternate techniques for extracting boundaries utilizing maximum gradient paths have been proposed [9, 14, 34]. In [9], adaptive thresholding was used to extract low magnitude edge pixels that are connected to high gradient edge pixels. Goshtasby and Shyu [33] used a minimum spanning tree to group high gradient pixels into boundary support regions. Each maximal path of the tree, equivalent to the maximum gradient path, was then fitted by a rational Gaussian curve. One important feature that these techniques lack is the direct exploitation of gradient direction information. The method utilizing both the gradient magnitude and direction in the initial step was proposed by Burns *et al* [8] where line-support regions were established from groups of adjacent pixels having large gradient magnitudes and similar gradient directions. Boundaries were extracted from the least square intensity surface fits of line-support regions. Improvement in noise immunity and boundary localization accuracy can be seen in the algorithm utilizing multiscale edge information [5, 32, 34, 56, 81]. Other edge linking techniques include fuzzy reasoning [45] and particle model [18]. In [18], the algorithm automatically tracks edge contours using Newtonian attractive forces at edge pixels. However, several parallel contours around an object as a result of particle motion yield poor edge localization and a fusion process is used to fuse mutliple contours into a single contour.

Since most edge linking techniques are directly designed to deal with edge elements,

the finest details of extracted boundaries are limited to the size of an edge element which achieves accuracy of at most one pixel. Although some algorithms [33, 38, 58, 78] can locate edges with subpixel accuracy in theory, boundary extraction with subpixel accuracy is rarely achieved. Eua-Anant [22] and Eua-Anant and Udpa [24] introduced the new concept of boundary extraction based on particle motion in a vector field that yields results with subpixel accuracy. However, the particle model in [22] is not optimized and suffers from an inertial effect causing shape distortion near object corners.

2.3.1.3 Template matching techniques

The more rigorous approaches for edge-oriented image segmentation have been implemented using template matching. In template matching, the estimated boundary is modeled as a deformable curve. The regularity of the estimated boundaries is controlled by structures and parameters of the model. Parameters of the model are adjusted to maximize the likelihood of the model and image data. Cooper [15] and Cooper *et al* [16] modeled the sequence of edge elements in a boundary as a Markov process estimated iteratively using a ripple filter. The boundary is deformed along pixel grids in such a way as to increase the joint likelihood between the hypothesized boundary and image data. Goshtasby and Shyu [33] used the rational Gaussian curve to fit the extracted high gradient regions. Kass [41] introduced the spline-based active contour models or snakes. Finding the optimal boundary is equivalent to minimizing the energy function of the contour including internal forces imposing contour rigidity and elasticity and image forces attracting the contour to salient features such as edges in the image. Ronfard [70] and Brzakovic [7] proposed an alternative region-based active contour model in which image forces depend on local regions partitioned by the contour. To prevent the contour from being trapped at weak local energy minima, Cohen [11, 12] proposed a model with an additional inflation force. The normalized image force is also used to avoid the problem of instability. Gunn and Nixon [35] proposed a dual active contour model that prevents the contour from getting stuck at local energy minima by balancing expansion and contraction forces between interlinked contours. Lai and Chin [44] proposed the global contour model using a regenerative shape matrix combined with a Markov random field to deal with local deformations. Geiger *et al* [30] applied dynamic programming to the active contour model with multiscale boundary extraction capability. General concepts, algorithms, applications, and other work on active contours can be found in [6]. In these models, constraints of smoothness and continuity as well as boundary topology are imposed without prior knowledge of image data which sometimes

Table 2.1 Comparison: template matching versus edge linking.

	Template matching	Edge linking
Computational cost	High. Optimization of a template is computationally intensive.	Low. Edge linking requires a simple search for extracting connected components of the boundary.
Initialization	Complex. A template requires complex initialization procedures.	Simple. The maximum gradient points or zero crossing points can be used as starting points of edge linking.
Flexibility	Low. Template matching fails when the template topology does not match the boundary topology.	High. Because of the sequential nature of search, edge linking is topologically unrestricted.
Accuracy	High. An optimization algorithm employed in template matching yields accurate results.	Low. Edge linking methods do not exploit subpixel optimization techniques resulting in less accurate boundaries.
Noise Immunity	High. Due to optimization, template matching is more robust to noise.	Low. Difference operators used for generating edge maps enhance noise.
Boundary quality	High. The constraints of continuity and smoothness imposed on the template ensure that the extracted boundary is always smooth and connected.	Low. Edge linking usually results in broken boundaries.

are not matched to actual image structures.

Besides optimization, initialization is also crucial for template matching. In order to allow the contour to converge to the image features, the initial contour must be placed close to the solution. For example, in region-based models [7, 15, 16, 70], the initial boundary must overlap or encompass the object. In active contour models, all nodes must be placed along the path lying near the desired image feature. In [35], one contour must be placed inside the object while another must wrap around the object. Lia and Chin [44] initialized the curve using the generalized Hough transform. Cohen and Ron [13] initialized the curve by placing endpoints at image features. Due to this constraint, in many cases, initialization is performed manually. The comparison between template matching and edge linking techniques is summarized in Table 2.1.

2.3.2 Boundary Representation

The boundary can be represented in two ways using either an analytic descriptor of the boundary commonly used in template matching or a sequence of boundary points that are used in edge linking. As dimension, size, and complexity of an image increase, identifying a suitable number of parameters representing the boundary becomes critical and cannot be ignored. The efficiency of boundary representation scheme can be determined by the number of parameters used in the representation compared with reconstruction error. Ideally, one desires a boundary representation scheme that uses a small number of parameters while maintaining structural details of the object boundary

In template matching, since the template is selected without any knowledge of the image feature, the number of parameters is usually fixed and the template can overfit or underfit the image feature. In other words, the template with the fixed number of parameters is not guaranteed to be an efficient boundary representation. In addition, in template matching, locations of boundary points are optimal along normals of the contour and not always optimal along the contour. In contrast to template matching, in edge linking, the number of boundary points varies linearly with the length of the extracted boundary. However, in general, these points can be collinear and hence redundant. Therefore, this boundary representation scheme is less efficient. After boundary extraction is completed, the boundary information is used for different high level image processing applications. Construction of the global image boundary model representing the overall relation between the objects usually requires additional information of the application objectives and interpretation of the retrieved boundary information. In most techniques, the organization of the resultant boundaries is left to users.

CHAPTER 3. GLOBAL SCALE ANALYSIS

In general, an image consists of objects of several scales. Image analysis performed at different scales can result in different interpretation of the same image. As demonstrated in Figures 3.1 and 3.2, images of the same objects observed at different resolutions reveal different structures. Images of a tree in Figure 3.1 can represent leaves, branches, and the whole tree depending on the observation scale. By analyzing objects at the scales starting from the finest scale to the largest scale, the overall relation between objects and scales can be perceived. Global scale analysis is the quantitative evaluation of structural details with respect to scales. This chapter is primarily devoted to the development of a mathematical framework for global scale analysis.

3.1 Edge Based Global Scale Analysis

Detection of objects in an image can be done by observing edges since the existence of edges indicates the existence of structures. In scale-space analysis [32, 52, 56, 83], detection of dominant structures in a scale-space image is done by observing behavior of edges in terms of zero crossings of the second order differential structures at various scales. However, the methods do not provide a quantitative estimate of existing scales. One way to estimate global changes of edge patterns is to measure statistical parameters such as intensity average, root-mean-square, and variance of normalized multiscale edge images $e(\sigma; x, y)$ as proposed in the previous research [22] where the edge characteristic-scale curves (ECS) are statistical functions of the normalized edge images:

$$\begin{aligned} Mean(\sigma) &= E(e(\sigma; x, y)), \\ RMS(\sigma) &= E(e^2(\sigma; x, y)), \\ SD(\sigma) &= \sqrt{E(e^2(\sigma; x, y)) - E(e(\sigma; x, y))^2}. \end{aligned}$$

The main idea underlying the use of statistical parameters of multiscale edge images at optimum scale σ^* is that sharpest edges are reflected by the minimum value of statistical measures at σ^*. If the original image contains multiscale structures, embedded scales

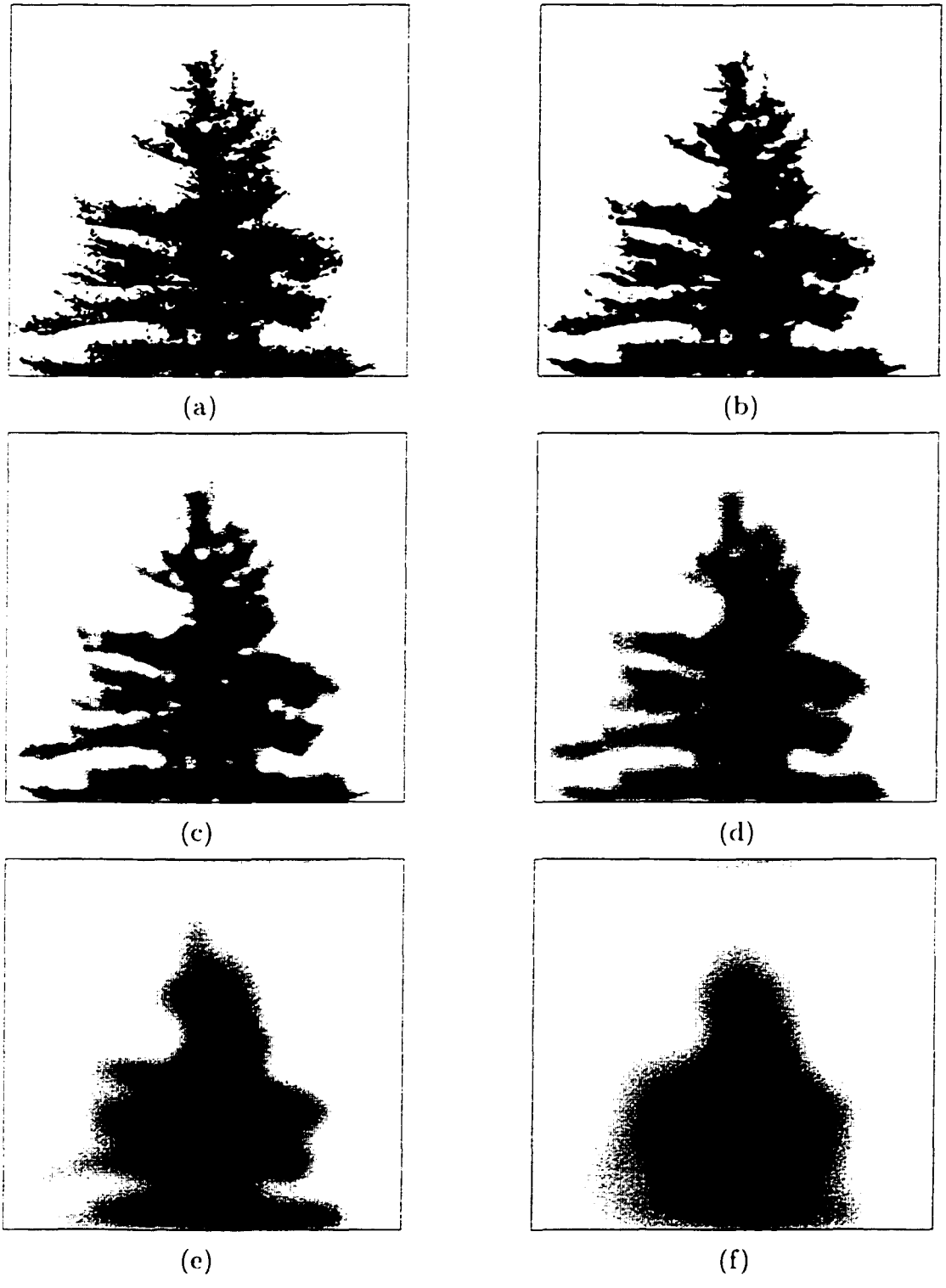


Figure 3.1 A tree observed at different resolutions: (a) original image. (b) to (f) smoothed images obtained using the Gaussian smoothing functions with $\sigma = 1, 2, 4, 8,$ and 16 pixels, respectively.

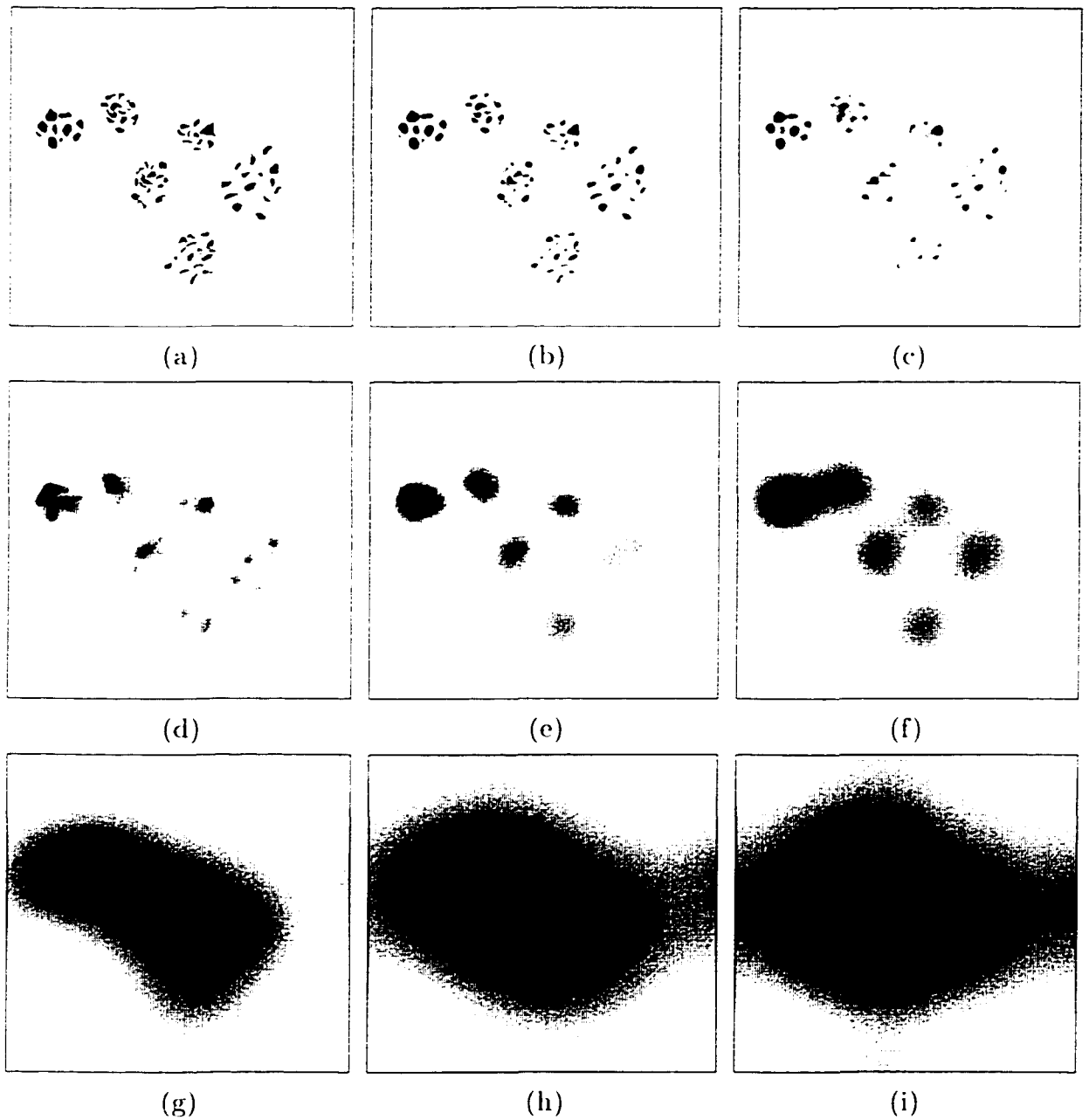


Figure 3.2 Cluster-like objects observed at different resolutions: (a) original image, (b) to (i) smoothed images obtained using the Gaussian smoothing functions with $\sigma = 1, 2, 4, 8, 16, 32, 64,$ and 128 pixels, respectively.

are expected to be correlated with local minima of the ECS curves. However, the ECS curves are seen to be sensitive to noise and are not always successful in detecting all the scales, accurately.

In contrast to spatial domain analysis, detection of dominant scales could be effectively obtained by analyzing the signal in frequency domain. The following sections describe the frequency domain approach for global scale analysis. Some aspects of multiscale representation followed by scale-frequency analysis are given next.

3.2 Multiscale Representation and Multiscale Differential Operators

Generally, in scale-space analysis, multiscale low pass filtering is used to separate information at different scales. The combination of multiscale low pass filters and single scale differential operators yields multiscale differential operators. In frequency domain, multiscale differential operators are band pass filters with variable bandwidth and center frequencies.

3.2.1 Multiscale Differential Operators

One-dimensional multiscale differential masks $o_t(\sigma; t)$ can be expressed by derivative of the multiscale, isotropic, linear, low pass filters, namely the multiscale smoothing functions $o(\sigma; t)$:

$$o_t(\sigma; t) = \frac{d}{dt} o(\sigma; t).$$

Using the inverse operation, the multiscale smoothing function $o(\sigma; t)$ can be computed from

$$o(\sigma; t) = \int o_t(\sigma; t) dt.$$

As in wavelet analysis, the smoothing function $o(\sigma; t)$ is the scaled version of the mother smoothing function $o(t)$:

$$o(\sigma; t) = \frac{1}{n(\sigma)} o(\frac{t}{\sigma}). \tag{3.1}$$

where $\frac{1}{n(\sigma)}$ is the normalization factor. For discrete signals, the continuous multiscale smoothing function o and differential operators are replaced by the sampled version of o and difference operators that approximate an ideal mathematical differential operator.

3.2.2 Multiscale Representation

Convolving the multiscale smoothing kernels of gradually increasing scales with the underlying signal results in a series of smoothed signals. This scale-parameter family of smoothed signals is called a scale-space image.

$$\{p(\sigma_i; t): i = 0, \ldots, N \text{ and } \sigma_0 < \ldots < \sigma_N\}. \tag{3.2}$$

where a smoothed image $p(\sigma: t)$ is obtained by convolving the signal $p(t)$ with the multiscale smoothing kernel of scale σ:

$$p(\sigma: t) = p(t) * o(\sigma: t).$$

while the scale parameter σ_i is expressed in the exponential form as

$$\sigma_i = \sigma_0 \cdot e^{\alpha \cdot i}$$

where $\sigma_0 \in R^+$ and $\alpha > 0$. In frequency domain analysis, data is represented by the Fourier transform:

$$\{P(\sigma_i: \omega): i = 0, \ldots, N\}. \tag{3.3}$$

where

$$P(\sigma: \omega) = P(\omega) \cdot \Phi(\sigma: \omega).$$

Analysis of multiscale representation requires the operators that are causal, linear, and invariant with respect to scale. The choice of scale invariant normalization schemes depends upon type of information to be investigated. In Fourier analysis, frequency components of the signal can be used for describing global patterns embedded in the signal. Therefore, global scale analysis can be performed effectively in frequency domain. In conjunction with multiscale differential operators, analysis of global scales in frequency domain will be discussed next.

3.2.3 Multiscale Structures

In general, objects at smaller scales are parts of objects at larger scales. For example, leaves, roots, and branches form a tree. There are several models for multiscale objects such as fractals and wavelets. A simple model of multiscale objects based on set theoretical operations is given here.

Let B_n be a window at scale n defined as

$$B = \{x : x = 1 \text{ if } x \in B \text{ otherwise } x = 0\}, \tag{3.4}$$

an object V_n at scale n. V_n can be obtained using

$$V_n = B_n \cap (\cup^{\infty} V_{n-1}) \tag{3.5}$$

$$= B_n \cap (\cup^{\infty}(B_{n-1} \cap (\cup^{\infty}(B_{n-2} \cap \ldots (\cup^{\infty} V_0))))). \tag{3.6}$$

where $\cup^{\infty} V_{n-1}$ represents a set of objects at scale $n-1$ spreading over the entire spatial domain. In other words, the window B_n is used to "crop" $\cup^{\infty} V_{n-1}$ to create the objects V_n. Equation 3.6 refers to a multiplicative multiscale pattern. This model simplifies analysis of the objects in frequency domain since the multiplicative multiscale representation in Equation 3.6 corresponds to the convolution between Fourier transforms of the envelopes B_n and $\cup^{\infty} V_{n-1}$. An example of a multiplicative multiscale object is illustrated in Figure 3.3. As seen, Figure 3.3-e is created by masking Figure 3.3-a with Figure 3.3-c. Equivalently, in frequency domain, the spectrum in Figure 3.3-f of the multiscale signal in Figure 3.3-e is obtained by convolving the spectrum in Figure 3.3-b with the spectrum in Figure 3.3-d which smears the impulses in Figure 3.3-b to sinc functions in Figure 3.3-f.

3.3 Scale Detection Based on Moments of Multiscale Power Spectra

From the model of multiplicative multiscale objects, the relation between spectrum of multiscale objects and existing scales is expressed by the convolution between spectra of objects of different scales. The spectrum usually contains several frequency lobes which can be analyzed using moments of multiscale spectra.

3.3.1 New Measure for Scale Analysis

For 1-dimensional signal $f(t)$, the n^{th} order moment of $f(t)$ is defined as

$$m_n(f(t)) = \int_{\mathbf{R}} t^n f(t) dt. \tag{3.7}$$

where $\mathbf{R} = (-\infty, \infty)$. If $f(t)$ is the mass distribution function along t-axis, the center of mass of $f(t)$ can be computed from moments by

$$\bar{t} = \frac{m_1(f(t))}{m_0(f(t))} \tag{3.8}$$

$$= \frac{\int_{\mathbf{R}} t f(t) dt}{\int_{\mathbf{R}} f(t) dt}. \tag{3.9}$$

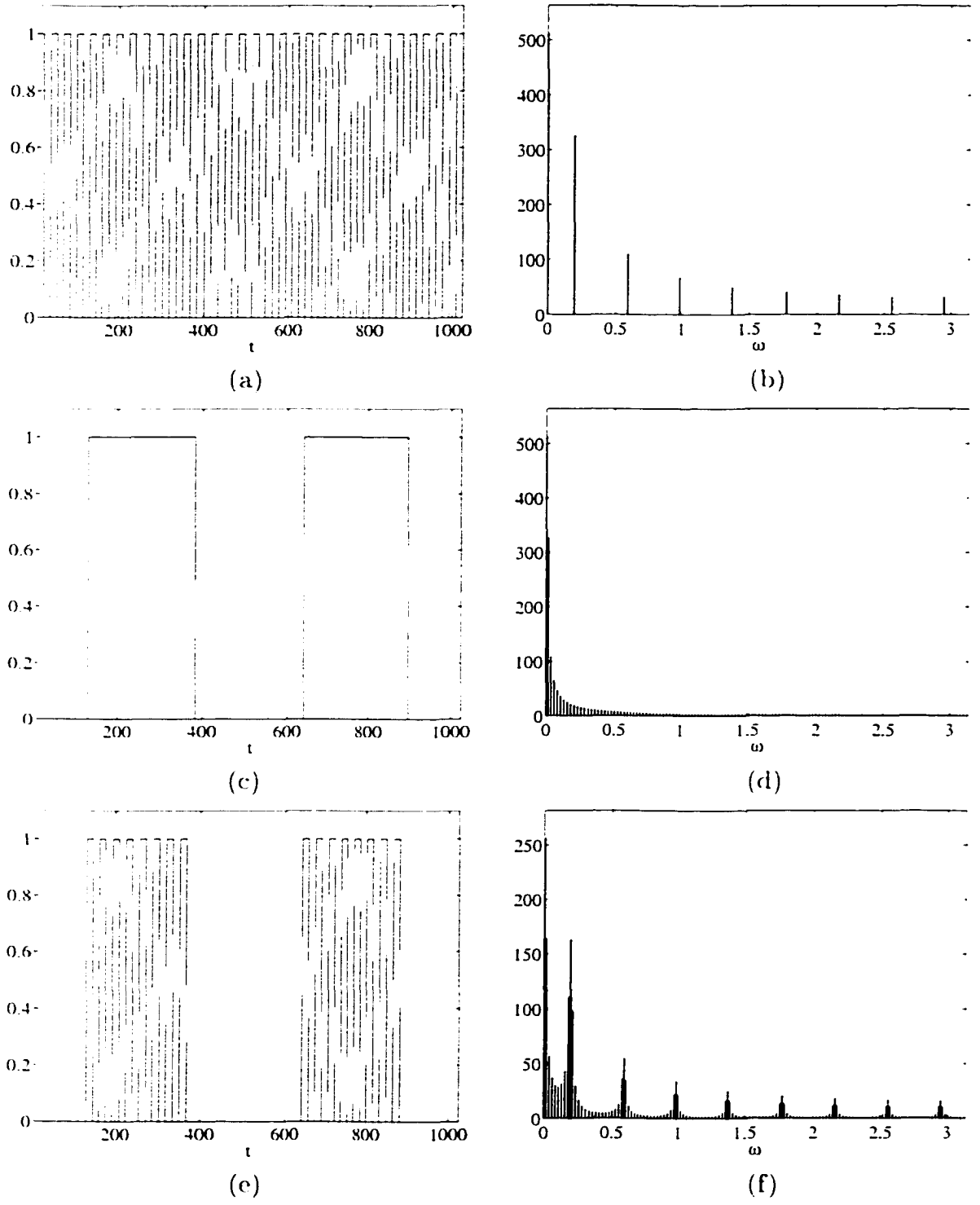


Figure 3.3 (a) and (b) A square wave signal of period 32 and its Fourier transform. (c) and (d) a square wave signal of period 512 and its Fourier transform. (e) and (f) a multiscale square wave signal and its Fourier transform.

The central moment of order n is then defined as

$$\mu_n = \int_{\mathbf{R}} (t - \bar{t})^n f(t) dt. \tag{3.10}$$

The second order normalized central moment, namely the *normalized variance*, is related to the spread of $f(t)$ and is given by

$$\rho^2 = \frac{\mu_2}{\mu_0} \tag{3.11}$$

$$= \frac{\int_{\mathbf{R}} (t - \bar{t})^2 f(t) dt}{\int_{\mathbf{R}} f(t) dt}. \tag{3.12}$$

In frequency domain, for a signal $f(t)$ with Fourier transform $F(\omega)$, the center frequency $\overline{\omega}$ of a power spectrum $|F(\omega)|^2$, $-\infty < \omega < \infty$, is given by,

$$\overline{\omega} = \frac{m_1(|F(\omega)|^2)}{m_0(|F(\omega)|^2)} \tag{3.13}$$

$$= \frac{\int_{\mathbf{R}} \omega |F(\omega)|^2 d\omega}{\int_{\mathbf{R}} |F(\omega)|^2 d\omega}. \tag{3.14}$$

For all real signals, the center frequency $\overline{\omega}$ is always zero because the power spectrum of a real signal is always symmetric with respect to the zero frequency. From Equation 3.12, the normalized variance ρ^2 of the power spectrum is given by

$$\rho^2(|F(\omega)|^2) = \frac{\mu_2(|F(\omega)|^2)}{\mu_0(|F(\omega)|^2)} \tag{3.15}$$

$$= \frac{\int_{\mathbf{R}} (\omega - \overline{\omega})^2 |F(\omega)|^2 d\omega}{\int_{\mathbf{R}} |F(\omega)|^2 d\omega}. \tag{3.16}$$

For a real signal, ρ^2 becomes

$$\rho^2(|F(\omega)|^2) = \frac{\int_{\mathbf{R}} \omega^2 |F(\omega)|^2 d\omega}{\int_{\mathbf{R}} |F(\omega)|^2 d\omega}. \tag{3.17}$$

For the discrete signal, integrations in Equation 3.17 are replaced by the summations:

$$\rho^2(|F(\omega)|^2) = \frac{\sum_i \omega_i^2 |F(\omega_i)|^2}{\sum_i |F(\omega_i)|^2}, \tag{3.18}$$

where $\omega_i \in [-\pi, \pi]$. Applying Equation 3.17 to the scale-frequency representation $P(\sigma : \omega)$, ρ^2 is a function of σ:

$$\rho^2(|P|^2 : \sigma) = \frac{\mu_2(|P(\sigma : \omega)|^2)}{\mu_0(|P(\sigma : \omega)|^2)}. \tag{3.19}$$

Consider a signal $p(t)$ with spectrum, $P(\omega) = c$. The scale-frequency representation using the multiscale kernel $o(\sigma; t)$ is obtained by

$$P(\sigma;\omega) = P(\omega) \cdot \Phi(\sigma;\omega) \tag{3.20}$$

$$= c \cdot \Phi(\sigma;\omega). \tag{3.21}$$

where $\Phi(\sigma;\omega)$ is the spectrum of the kernel $o(\sigma; t)$. Applying Equation 3.19 to Equation 3.21, we get

$$\rho^2(|P|^2;\sigma) = \frac{\mu_2(|c \cdot \Phi(\sigma;\omega)|^2)}{\mu_0(|c \cdot \Phi(\sigma;\omega)|^2)} \tag{3.22}$$

$$= \frac{\mu_2(|\Phi(\sigma;\omega)|^2)}{\mu_0(|\Phi(\sigma;\omega)|^2)} \tag{3.23}$$

$$= \rho^2(|\Phi|^2;\sigma) \tag{3.24}$$

which is equivalent to the normalized variance of the power spectrum $\Phi(\sigma;\omega)$ of the kernel $o(\sigma; t)$.

Defining the power spectrum normalized variance ratio (PSNVR) as

$$PSNVR(P;\sigma) = \frac{\rho^2(|P|^2;\sigma)}{\rho^2(|\Phi|^2;\sigma)}. \tag{3.25}$$

where $P(\sigma;\omega)$ is the spectrum of the multiscale representation $p(\sigma; t)$ of the underlying signal $p(t)$ and $\Phi(\sigma;\omega)$ is the spectrum of the multiscale kernel $o(\sigma; t)$, we see that PSNVR = 1 when $P(\omega)$ is constant, i.e., if the signal is composed of just white noise, PSNVR = 1. In contrast, if $P(\omega)$ is not constant but has a peak at ω^*, we have

$$\rho^2(|P|^2;\sigma) < \rho^2(|\Phi|^2;\sigma) \text{ if } |\omega^*| < \rho(|\Phi|^2;\sigma)$$

and

$$\rho^2(|P|^2;\sigma) > \rho^2(|\Phi|^2;\sigma) \text{ if } |\omega^*| > \rho(|\Phi|^2;\sigma).$$

This implies that PSNVR $\neq$ 1 indicates the presence of a scale or feature in the image.

If $p_t(t)$ is of interest, we can further define the differential power spectrum normalized variance ratio (DPSNVR) as

$$DPSNVR(P;\sigma) = PSNVR(P_t;\sigma) \tag{3.26}$$

$$= \frac{\rho^2(|P_t|^2;\sigma)}{\rho^2(|\Phi_t|^2;\sigma)}. \tag{3.27}$$

where $P_t(\sigma;\omega)$ is the spectrum of the multiscale differential representation $p_t(\sigma; t)$ of $p(t)$ and $\Phi_t(\sigma;\omega)$ is the spectrum of the multiscale differential kernel $o_t(\sigma; t)$. Note that the normalization factor $\frac{1}{n(\sigma)}$ in Equation 3.1 does not affect the PSNVR and DPSNVR.

Consider the case of an ideal multiscale low pass filter $o(\sigma: t)$ with the cutoff frequency $\omega_c = \frac{1}{\sigma}$:

$$|\Phi(\sigma: \omega)|^2 = \begin{cases} 1. & |\omega| \in [0. \omega_c] \\ 0 & \text{otherwise.} \end{cases}$$

The normalized variance of $|\Phi(\sigma: \omega)|^2$ is given by

$$\begin{aligned} \rho^2(|\Phi(\sigma: \omega)|^2) &= \frac{\int_{-\omega_c}^{\omega_c} \omega^2 d\omega}{\int_{-\omega_c}^{\omega_c} d\omega} \\ &= \frac{\omega_c^2}{3} \\ &= \frac{1}{3\sigma^2}. \end{aligned}$$

which is a monotonically decreasing function of σ. Figures 3.4-a and 3.4-b show $|\Phi(\sigma: \omega)|^2$ with $\omega_c = 2$ and $\rho^2(|\Phi(\sigma: \omega)|^2)$.

Consider the signal $p(t)$ with its power spectrum $|P(\omega)|^2$ given by

$$|P(\omega)|^2 = \begin{cases} 1. & |\omega| \in [\omega_1. \omega_2] \cup [\omega_3. \omega_4] \text{ where } 0 < \omega_1 < \omega_2 < \omega_3 < \omega_4 \\ 0 & \text{otherwise.} \end{cases}$$

Applying $o(\sigma: t)$ to $p(t)$ yields a multiscale smoothed signal

$$p(\sigma: t) = p(t) * o(\sigma: t)$$

with the corresponding power spectrum

$$|P(\sigma: \omega)|^2 = |P(\omega)| \cdot |\Phi(\sigma: \omega)|.$$

The normalized variance of $|P(\sigma: \omega)|^2$ is given by

1. for $\omega_c > \omega_1$.

$$\rho^2(|P(\sigma: \omega)|^2) = \frac{\omega_4^3 - \omega_3^3 + \omega_2^3 - \omega_1^3}{3(\omega_4 - \omega_3 + \omega_2 - \omega_1)}.$$

2. for $\omega_3 < \omega_c < \omega_4$.

$$\rho^2(|P(\sigma: \omega)|^2) = \frac{\omega_c^3 - \omega_3^3 + \omega_2^3 - \omega_1^3}{3(\omega_c - \omega_3 + \omega_2 - \omega_1)}.$$

3. for $\omega_2 < \omega_c < \omega_3$.

$$\rho^2(|P(\sigma: \omega)|^2) = \frac{\omega_2^3 - \omega_1^3}{3(\omega_2 - \omega_1)}.$$

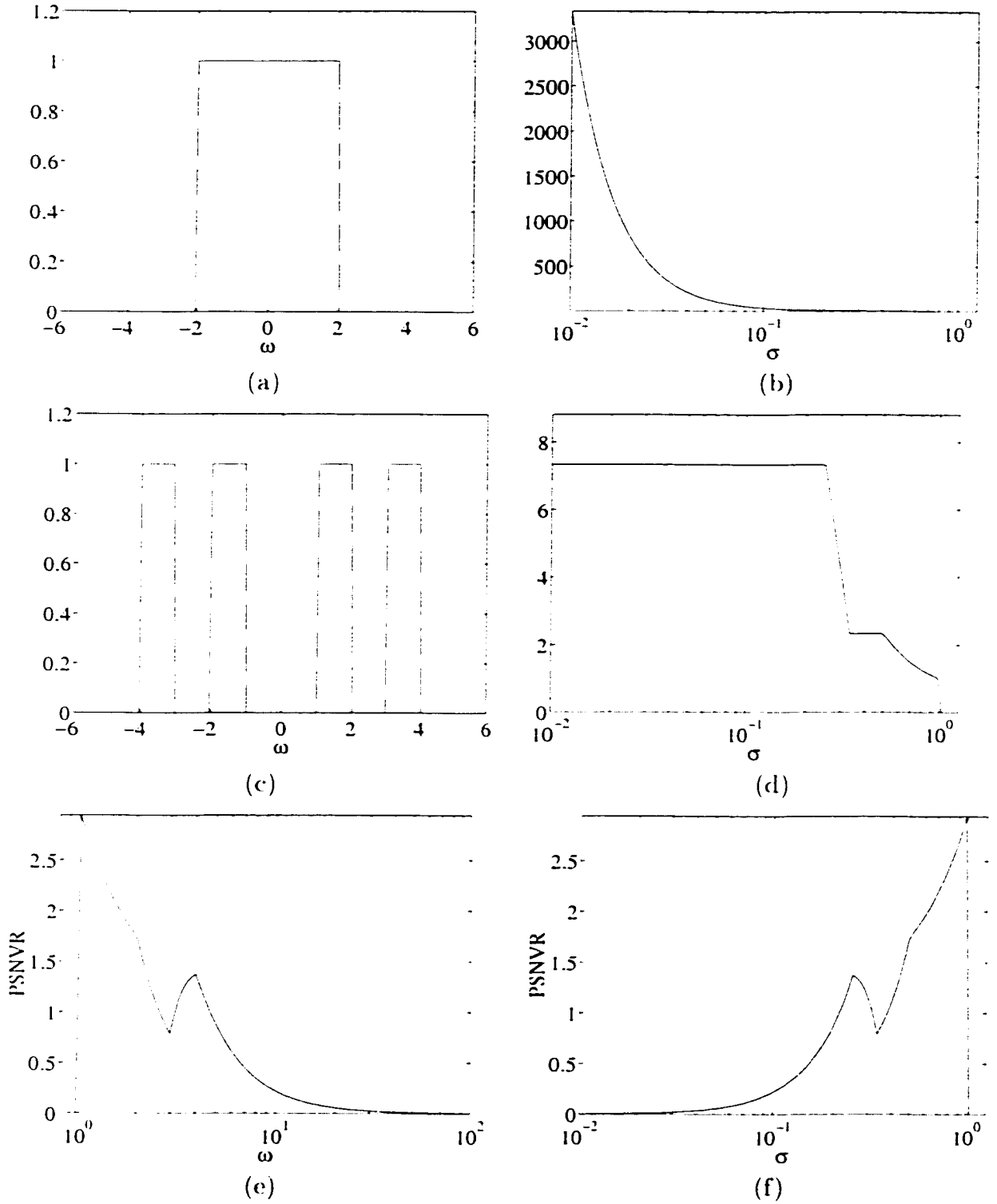


Figure 3.4 (a) $|\Phi(\sigma;\omega)|^2$ with $\omega_c = 2$. (b) $\rho^2(|\Phi(\sigma;\omega)|^2)$. (c) $|P(\omega)|^2$ with $\omega_1 = 1, \omega_2 = 2, \omega_3 = 3$, and $\omega_4 = 4$. (d) $\rho^2(|P(\sigma;\omega)|^2)$. (e) $PSNVR(\omega_c)$. (f) $PSNVR(\sigma)$.

4. for $\omega_1 < \omega_c < \omega_2$,

$$\rho^2(|P(\sigma;\omega)|^2) = \frac{\omega_c^3 - \omega_1^3}{3(\omega_c - \omega_1)}.$$

5. for $\omega_c < \omega_1$,

$$\rho^2(|P(\sigma;\omega)|^2) = 0.$$

Figures 3.4-c and 3.4-d show $|P(\sigma;\omega)|^2$ and $\rho^2(|P(\sigma;\omega)|^2)$ with $\omega_1 = 1, \omega_2 = 2, \omega_3 = 3$, and $\omega_4 = 4$. Figures 3.4-e and 3.4-f show the corresponding PSNVR graphs as functions of σ and ω_c respectively where there exist 2 local minima due to frequency lobes of $|P(\omega)|^2$. This example shows that local minima of the PSNVR graph indicate the existence of frequency lobes in the power spectrum of the signal. Moreover, in Figure 3.4-f, local minima of $PSNVR(\omega_c)$ occur at $\omega_c = \omega_3$ and $\omega_c > \omega_4$ which indicates that the values of ω_c at local minima of the graph PSNVR are ideal for separating frequency lobes. For the application of multiscale edge detection, the values of σ at local minima of the DPSNVR graph can be used for separating edges of different scales.

3.3.2 Experimental Results: Global Scale Analysis of 1-Dimensional Multiscale Edge Signals

In this section, several 1-dimensional simulated signals are analyzed using the proposed method and is compared to the edge characteristic-scale (ECS) analysis introduced in the previous research [22]. In all examples, the multiscale Gaussian differential kernel is used:

$$o_t(\sigma;t) = -\frac{t}{\sqrt{2\pi}\sigma^3} e^{-\frac{t}{2\sigma^2}}$$

with the corresponding Fourier transform:

$$\Phi_t(\sigma;\omega) = j\omega e^{-\frac{\sigma^2\omega^2}{2}}.$$

Figure 3.5 illustrates the example of a multiplicative double scale pattern where the signal in Figure 3.5-a is the multiplication between square waves of periods 16 and 128. Figure 3.5-b shows the corresponding magnitude spectrum. Figure 3.5-c shows the ECS curves where AVG, RMS, and SD denote average, root-mean-square and standard deviation respectively. The graph DPSNVR is shown in Figure 3.5-d. The values of σ obtained from both global scale analysis schemes are displayed in Table 3.1. As seen in Figure 3.5-b, the corresponding spectrum consists of several frequency lobes due to harmonic terms and signal aliasing. Most of the graphs contain two distinct local minima reflecting two existing scales while the $SD(\sigma)$ curve has additional spurious

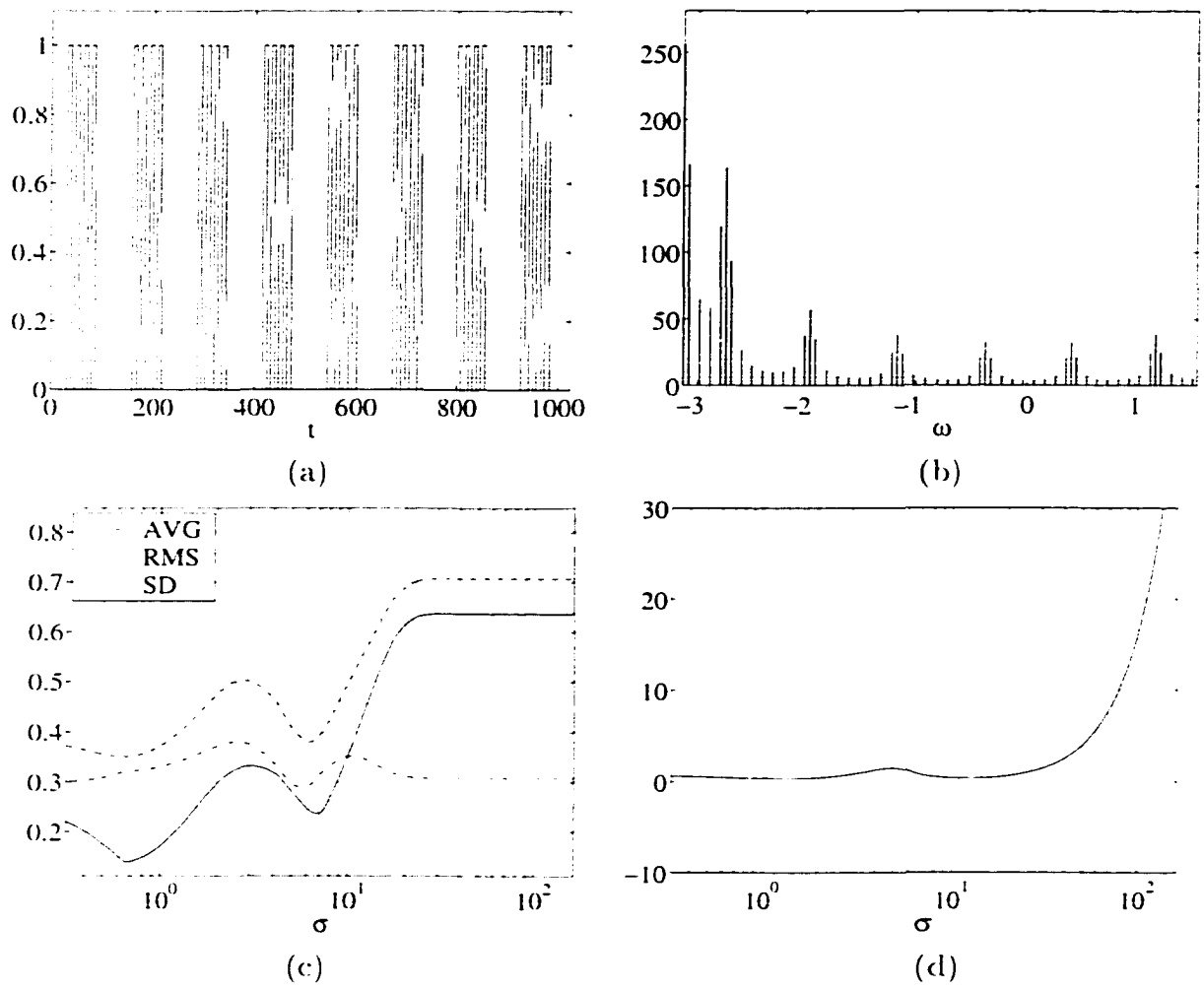


Figure 3.5 Double scale signal: (a) original signal. (b) Fourier transform magnitude. (c) ECS curves. (d) DPSWVR graph.

Table 3.1 Global scale analysis of the signal in Figure 3.5.

	Scale 1	Scale 2	
σ at local minima of $AVG(\sigma)$	0.66	6.85	93.56
σ at local minima of $RMS(\sigma)$	0.64	6.21	
σ at local minima of $SD(\sigma)$	0.32 0.69	5.73	60.52
σ at local minima of $DPSWVR(\sigma)$	1.41	12.41	

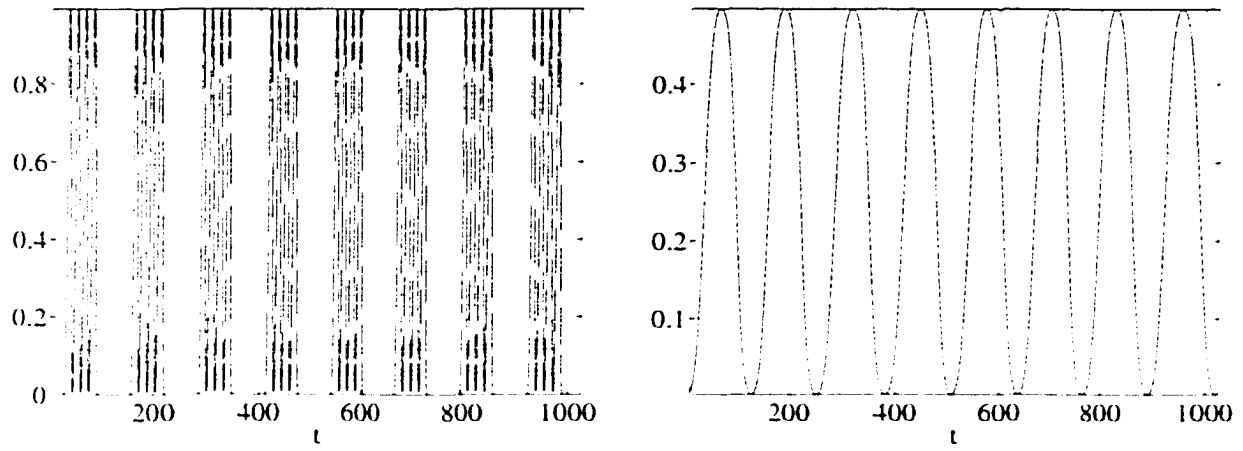


Figure 3.6 Smoothed signals of the signal in Figure 3.5-a obtained using multiscale Gaussian smoothing filters with σ selected from local minima in the DPSNVR graph: (a) $\sigma = 1.41$, (b) $\sigma = 12.41$.

local minima. From Table 3.1, all schemes detect both existing scales. The values of σ obtained from ECS analysis are relatively smaller than those obtained from the frequency domain approach. Figure 3.6 shows smoothed signals at two scales obtained using the multiscale Gaussian smoothing kernels with the values of σ selected from local minima of the DPSNVR graph. While the signal in Figure 3.6-a represents the pattern with fine details, the signal in Figure 3.6-b represents coarse structures.

The next example shown in Figure 3.7 is the multiplicative triple scale signal generated by the product between square waves of periods 16, 128, and 512. Tables 3.2 shows the values of σ obtained from both global scale analysis schemes. Similar to the previous example, the spectrum in Figure 3.7-b consists of several frequency lobes. Clearly, 3 distinct local minima can be seen in all ECS and DPSNVR curves except the edge average-scale curve. These 3 peaks are consistent with 3 existing scales of the signal. The smoothed signals obtained using the multiscale Gaussian smoothing filters with σ selected from the local minima of the DPSNVR graph in Figure 3.7-d are shown in Figure 3.8.

The next example shown in Figure 3.9 is the degraded version of the signal in Figure 3.7-a corrupted by additive white Gaussian noise $N(0, 0.5)$ (signal to noise ratio -3.70 dB). The values of σ obtained from global scale analysis schemes are shown in Table 3.3. As seen, ECS analysis method yields several false indications of scales introduced by noise while no spurious scale is detected using the DPSNVR. This example shows

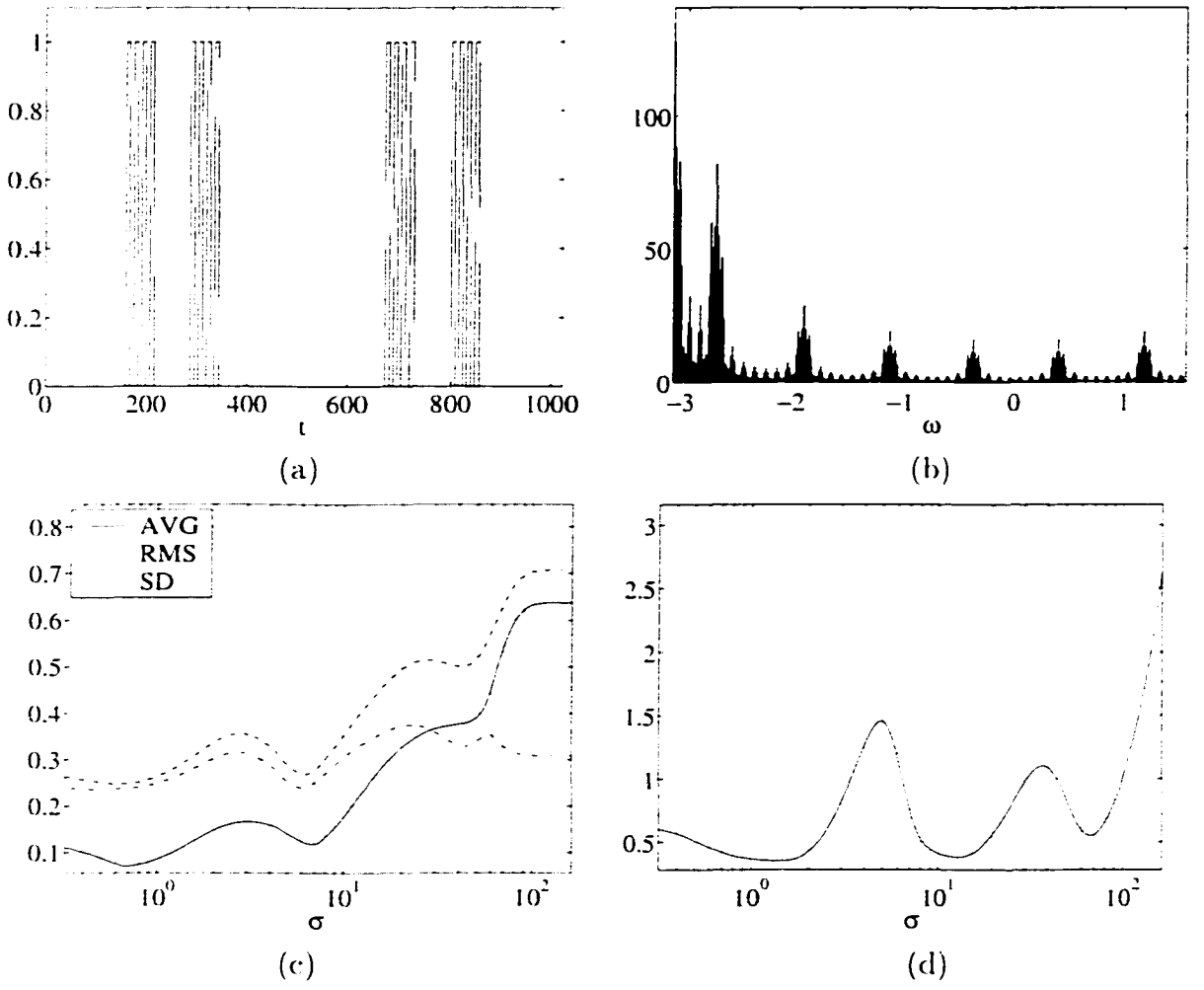


Figure 3.7 Triple scale signal: (a) original signal. (b) Fourier transform magnitude. (c) ECS curves. (d) DPSNVR graph.

Table 3.2 Global scale analysis of the signal in Figure 3.7.

	Scale 1	Scale 2	Scale 3
σ at local minima of $AVG(\sigma)$	0.66	6.85	
σ at local minima of $RMS(\sigma)$	0.64	6.21	41.54
σ at local minima of $SD(\sigma)$	0.50	5.97	42.37
σ at local minima of $DPSNVR(\sigma)$	1.41	12.66	66.82

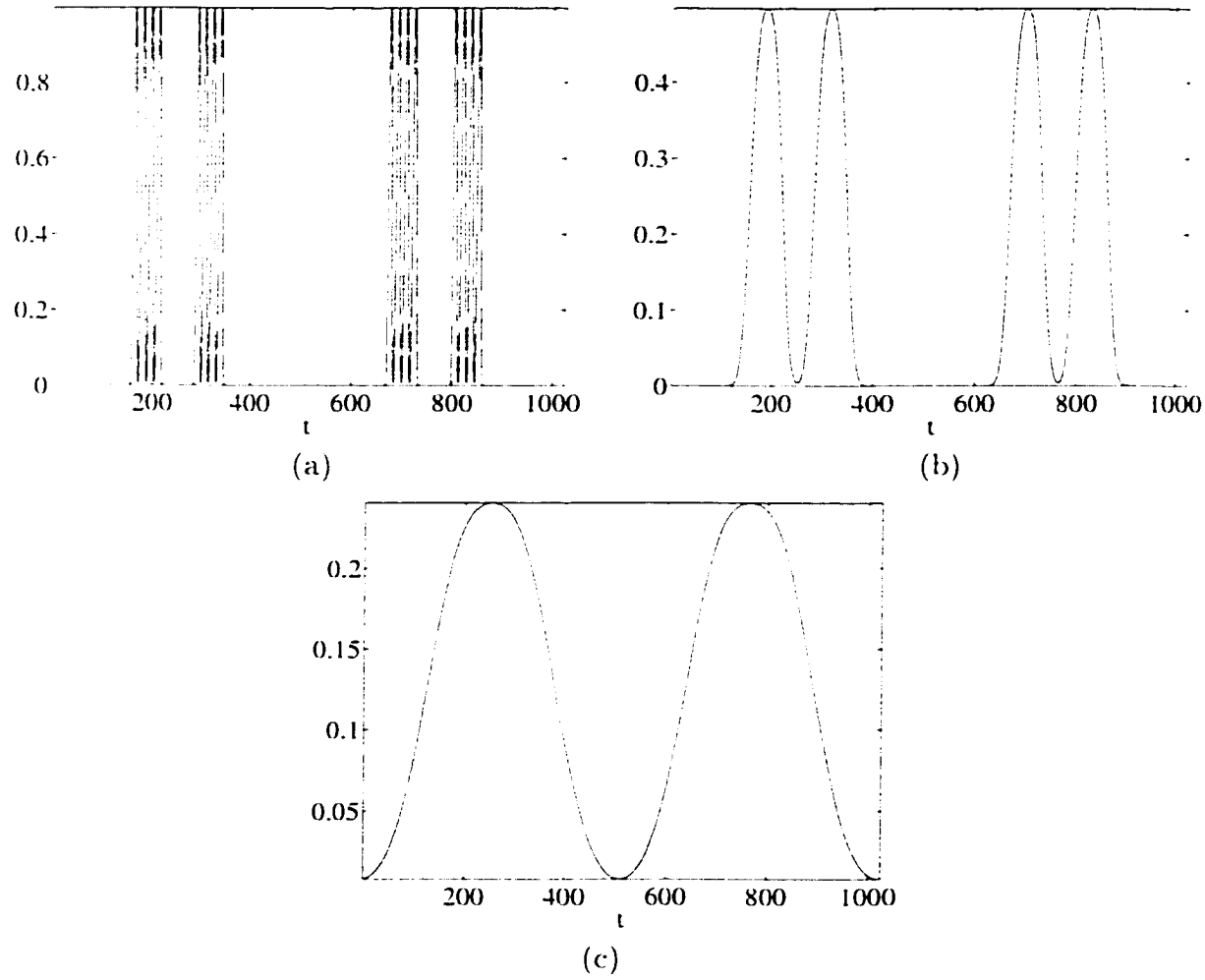


Figure 3.8 Smoothed signals of the signal in Figure 3.7-a obtained using multiscale Gaussian smoothing filters with σ selected from local minima in the DPSNVR graph: (a) $\sigma = 1.41$. (b) $\sigma = 12.66$. (c) $\sigma = 66.82$.

that the noise has little effect on the DPSNVR. This is because frequency components of noise spreading uniformly over the entire spectrum is averaged out in the computation of moments of the spectrum and hence has little effect on the DPSNVR. For ECS analysis, normalization factors are based on strength of the strongest edges which depend on data itself and cannot be predicted. Therefore, ECS analysis is less reliable. Figure 3.10 shows smoothed signals obtained using the Gaussian smoothing filters with σ obtained from local minima in the DPSNVR graph in Figure 3.9-d. In summary, global scale analysis based on the DPSNVR is a very robust method.

3.4 Extension to Multidimensional Signals

In order to extend the proposed global scale analysis method to images, several issues must be taken into account. First, the operator must be homogeneous and isotropic, i.e., translation invariant and directional invariant. In the frequency domain, this requirement is equivalent to rotational invariance. In continuous mathematics, the N-dimensional multiscale differential mask can be expressed as the convolution of multiscale isotropic smoothing kernel $o(\sigma; x_1, \ldots, x_N)$ and directional differential operators $\frac{d}{dx_i}$ as follows:

$$o_{x_i}(\sigma; x_1, \ldots, x_N) = \frac{d}{dx_i} * o(\sigma; x_1, \ldots, x_N), \; i = 1, \ldots, N. \tag{3.28}$$

where o_{x_i} is the multiscale differential mask in x_i direction. Examples of continuous 2-dimensional multiscale differential masks and corresponding multiscale smoothing functions are shown in Table 3.4. In the discrete case, the continuous multiscale smoothing kernel is replaced by its sampled version and the directional differential operator is replaced by the directional difference operator as demonstrated by the examples in Figure 3.11. Figure 3.12 shows examples of multiscale differential masks and the corresponding multiscale smoothing functions.

In general, in the N-dimensional spatial domain, the position is expressed in Cartesian coordinate by a vector $\vec{x} = (x_1, \ldots, x_N)$. Equivalently, the frequency component is expressed by the frequency vector $\vec{\omega} = (u_1, \ldots, u_N)$ where u_i represents the frequency measured in x_i-direction. It can be seen in Table 3.4 that, due to the constraint of isotropy imposed on the multiscale smoothing function, o can be written as a function of a single variable r where $r = \sqrt{x^2 + y^2}$. In general, the N-dimensional multiscale isotropic smoothing function can be written as $o(\sigma; r)$ with the center at $r = 0$, where $r = \sqrt{x_1^2 + \ldots + x_N^2}$ is the spatial radius. Similarly, the Fourier transform of an isotropic

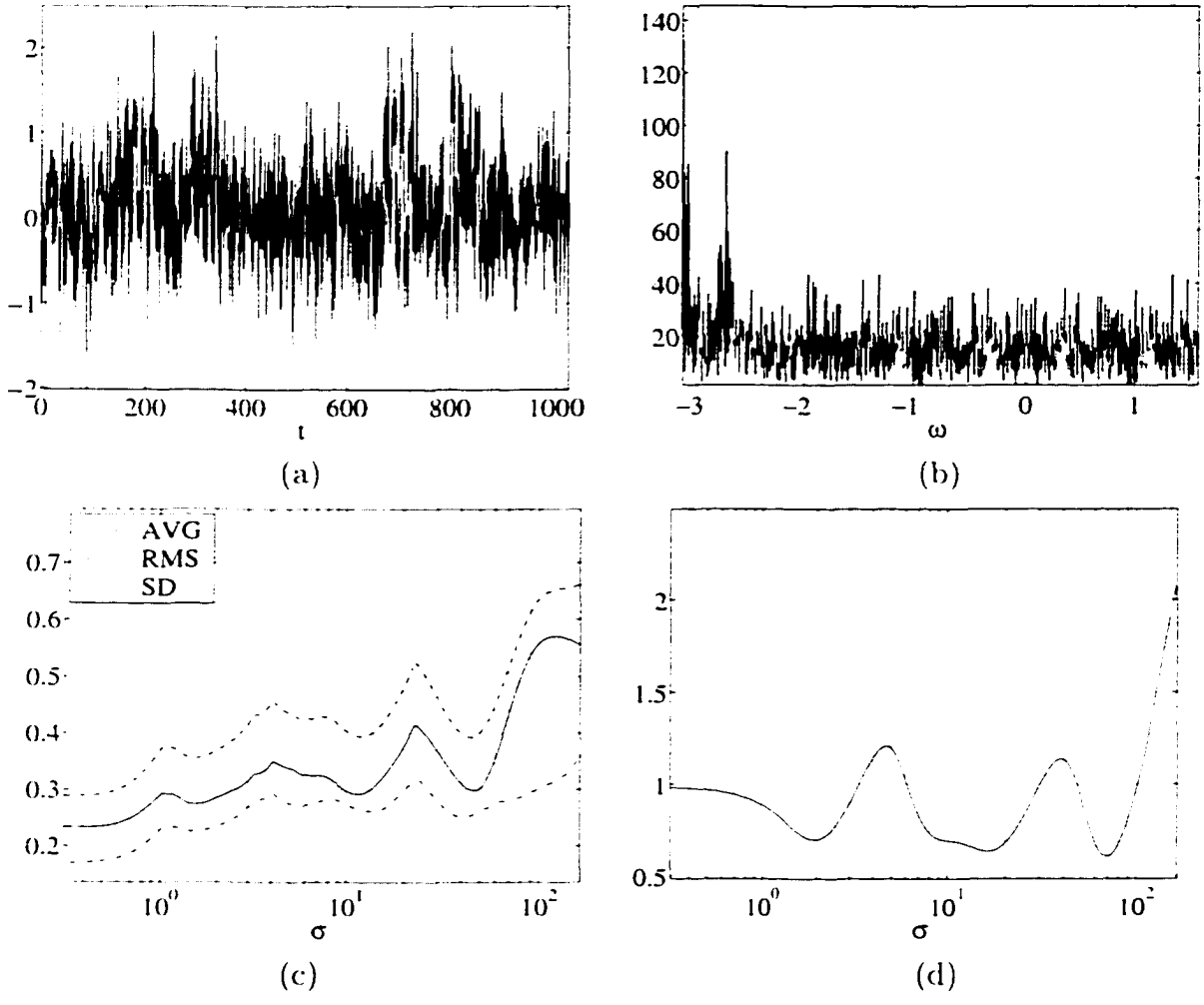


Figure 3.9 Triple scale signal with noise: (a) degraded version of the signal in Figure 3.7-a with SNR −3.70 dB. (b) Fourier transform magnitude. (c) ECS curves. (d) DPSNVR graph.

Table 3.3 Global scale analysis of the signal in Figure 3.9.

	Scale 1	Scale 2	Scale 3
σ at local minima of $AVG(\sigma)$	0.40 1.11 1.52	3.36 6.09 6.59 11.47	46.78
σ at local minima of $RMS(\sigma)$	0.34 1.11 1.49	5.97 6.46 7.27 12.17	44.97
σ at local minima of $SD(\sigma)$	0.32 1.07 1.46	5.73 13.98	40.73
σ at local minima of $DPSNVR(\sigma)$	1.93	16.38	69.52

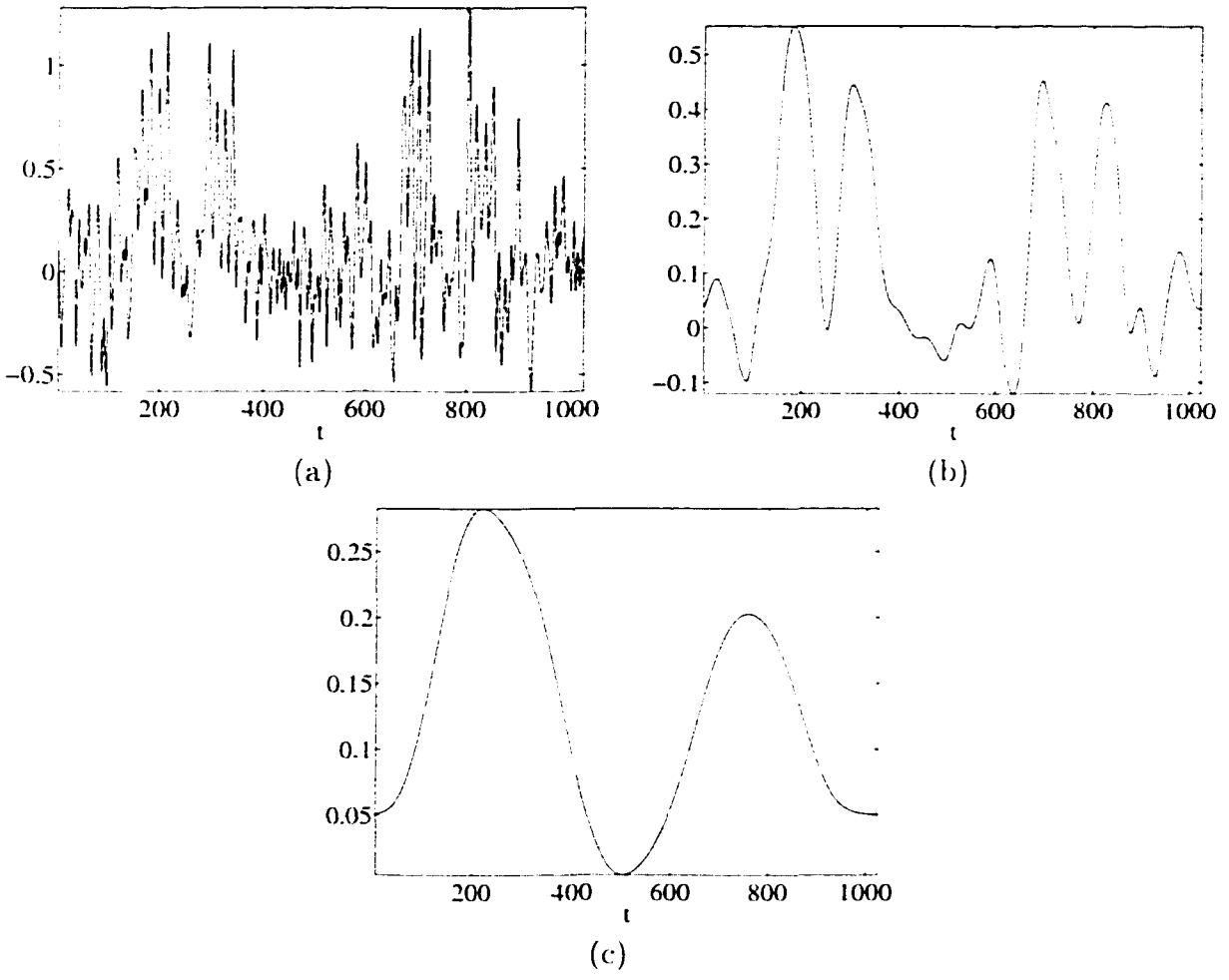


Figure 3.10 Smoothed signals of the signal in Figure 3.9-a obtained using multiscale Gaussian smoothing filters with σ obtained from local minima in the DPSNVR graph: (a) $\sigma = 1.93$. (b) $\sigma = 16.38$. (c) $\sigma = 69.52$.

(a) (b)

Figure 3.11 Example of differential masks (a) in x-direction and (b) y-direction.

smoothing function o can be expressed as $\Phi(\sigma; R)$ with the center frequency at $R = 0$ where $R = \sqrt{u_1^2 + \ldots + u_N}$ refers to the *frequency radius*. However, the Fourier transform of the multiscale directional differential mask, $o_{x_i}(\sigma; x_1, \ldots, x_N)$, is given by

$$\Phi_{x_i}(\sigma; u_1, \ldots, u_N) = \mathcal{F}\{o_{x_i}(\sigma; x_1, \ldots, x_N)\} \quad (3.29)$$

$$= ju_i\Phi(\sigma; u_1, \ldots, u_N). \quad (3.30)$$

where $\Phi(\sigma; u_1, \ldots, u_N)$ is the Fourier transform of the multiscale smoothing function $o(\sigma; x_1, \ldots, x_N)$. Due to the factor ju_i on the right side of Equation 3.30, Φ_{x_i} is not isotropic. One can create an operator with rotational invariance property as follows.

Consider the power spectrum function $|\Psi|^2$ given below:

$$|\Psi|^2 = |\Phi_{x_1}|^2 + \ldots + |\Phi_{x_N}|^2 \quad (3.31)$$

$$= (u_1^2 + \ldots + u_N^2) \cdot |\Phi|^2 \quad (3.32)$$

$$= R^2 \cdot |\Phi|^2. \quad (3.33)$$

Since $|\Phi|^2$ is rotationally invariant, $|\Psi|^2$ is also rotationally invariant and can be expressed as a function of R. Moreover, the multiplication by R^2 on the right hand side also results in another desirable property of $|\Psi|^2$ for global scale analysis, namely localization with respect to R. Figure 3.13 illustrates examples of 2-dimensional $|\Psi|^2$ generated from the Gaussian smoothing functions of different scales. As demonstrated by images in Figures 3.13-a and 3.13-c, $|\Psi|^2$ is refered to as the *multiscale power spectrum ring* (MSPSR) of order 1, or simply, the ring. Varying the scale σ creates the rings of different diameters resulting in the property of localization with respect to R.

Table 3.4 2-Dimensional multiscale differential masks and multiscale smoothing functions.

$o_x(\sigma; x, y)$	$o_y(\sigma; x, y)$	$o(\sigma; x, y)$
Gaussian		
$-\frac{x}{2\pi\sigma^4} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$	$-\frac{y}{2\pi\sigma^4} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$	$\frac{e^{-\frac{x^2+y^2}{2\sigma^2}}}{2\pi\sigma^2}$
GWIMV*		
$-\frac{x \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}}{2\pi\sigma^2\sqrt{x^2+y^2}}$	$-\frac{y \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}}{2\pi\sigma^2\sqrt{x^2+y^2}}$	$-\frac{e^{-\frac{x^2+y^2}{2\sigma^2}}}{2\pi\sigma}\left(\sum_{n=0}^{\infty}\frac{(\frac{\sqrt{x^2+y^2}}{\sigma})^{2n+1}}{\prod_{i=0}^{n}(2i+1)}\right) + C$
Moment*†		
$-2x$	$-2y$	$-(x^2+y^2) + C$
Distance*†		
$-\frac{2x}{\sqrt{x^2+y^2}}$	$-\frac{2y}{\sqrt{x^2+y^2}}$	$-\sqrt{x^2+y^2} + C$

* C is a constant such that $\min o(\sigma; x, y) = 0$.

† $(x, y) \in \{(x, y): \sqrt{x^2+y^2} \leq r\}$ where r is the radius of the mask.

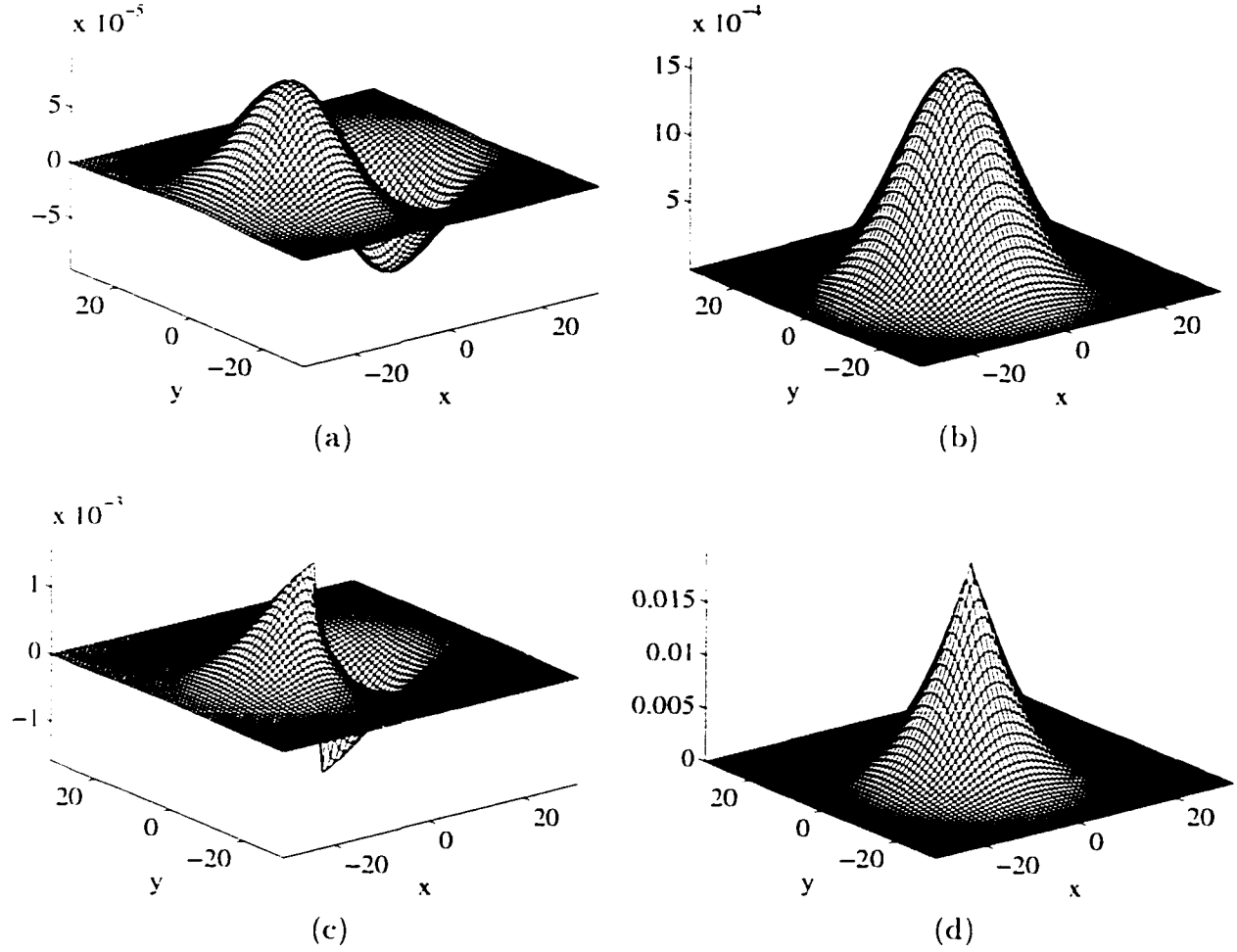


Figure 3.12 Examples of 2-dimensional multiscale differential masks and multiscale smoothing functions: (a) the Gaussian differential masks in x-direction and (b) the Gaussian function with $\sigma = 20$. (c) the GWIMV mask [22] in x-direction and (d) the corresponding multiscale smoothing function of the GWIMV mask with $\sigma = 20$.

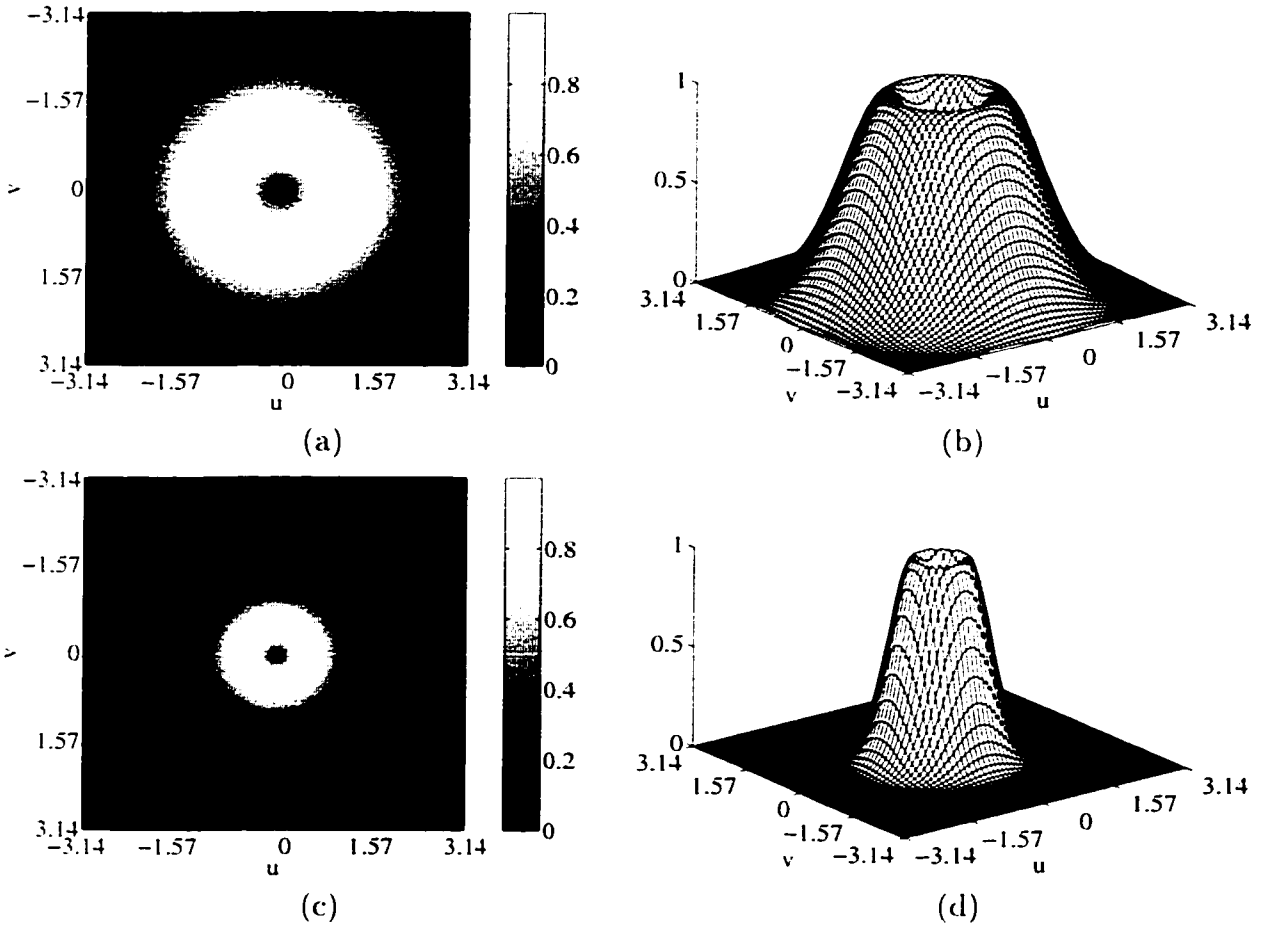


Figure 3.13 Magnitude images and surface plots of the multiscale Gaussian power spectrum rings: (a) and (b) $\sigma = 1$. (c) and (d) $\sigma = 2$.

3.4.1 Application of Multiscale Power Spectrum Rings to Global Scale Analysis

The multiscale power spectrum ring possesses several properties including localization with respect to scale and frequency radius and rotational invariance. Using the MSPSR as the kernel, the generalized N-dimensional global scale analysis method based on the DPSNVR is established as follows.

For an N-dimensional signal $f(x_1, \ldots, x_N)$, the generalized moment of order $(n_1, \ldots, n_N)$ is expressed as

$$m_{n_1,\ldots,n_N} = \int_{\mathbf{R}^N} x_1^{n_1} \cdot x_2^{n_2} \cdots x_N^{n_N} \cdot f(x_1, \ldots, x_N) dx_1 dx_2 \cdots dx_N,$$

where $\mathbf{R} = (-\infty, \infty)$. Similar to the 1-dimensional case, the center of mass $(\overline{x}_1, \ldots, \overline{x}_N)$ in the N-dimensional space is computed from

$$\overline{x}_i = \frac{m_{0,\ldots,i=1,\ldots,0}}{m_{0,\ldots,0}}.$$

Using $(\overline{x}_1, \ldots, \overline{x}_N)$ as the reference point, the central moment, μ, of order $(n_1, \ldots, n_N)$ is given by

$$\mu_{n_1,\ldots,n_N} = \int_{\mathbf{R}^N} (x_1 - \overline{x}_1)^{n_1} \cdot (x_2 - \overline{x}_2)^{n_2} \cdots (x_N - \overline{x}_N)^{n_N} \cdot f(x_1, \ldots, x_N) dx_1 dx_2 \cdots dx_N.$$

The normalized central moment, η, of order $(n_1, \ldots, n_N)$ is defined by

$$\eta_{n_1,\ldots,n_N} = \frac{\mu_{n_1,\ldots,n_N}}{\mu_{0,\ldots,0}}.$$

In the N-dimensional invariance theory [73], there are several combinations of normalized central moments that generate moments which are invariant to translation, rotation, and scale change. One invariant moment is the normalized variance ρ^2:

$$\rho^2 = \eta_{2,0,\ldots,0} + \eta_{0,2,\ldots,0} + \ldots + \eta_{0,0,\ldots,2}. \tag{3.34}$$

which can be expanded as

$$\begin{aligned} \rho^2 &= \frac{\int_{\mathbf{R}^N} ((x_1 - \overline{x}_1)^2 + \ldots + (x_N - \overline{x}_N)^2) f(x_1, \ldots, x_N) dx_1 \cdots dx_N}{\int_{\mathbf{R}^N} f(x_1, \ldots, x_N) dx_1 \cdots dx_N} \\ &= \frac{\int_{\mathbf{R}^N} r^2 f(x_1, \ldots, x_N) dx_1 \cdots dx_N}{\int_{\mathbf{R}^N} f(x_1, \ldots, x_N) dx_1 \cdots dx_N}. \end{aligned}$$

In fact, ρ^2 measures the average square radius with respect to the center of f.

In the global scale analysis of a 1-dimensional signal, $\rho^2(|P|^2;\sigma)$ is compared with $\rho^2(|\Phi|^2;\sigma)$ to obtain the PSNVR. Similarly, in N-dimensional space, the PSNVR is defined by

$$PSNVR(P;\sigma) = \frac{\rho^2(|P(\vec{\omega}) \cdot \Phi(\sigma;\vec{\omega})|^2)}{\rho^2(|\Phi(\sigma;\vec{\omega})|^2)}. \tag{3.35}$$

where P and Φ are the spectra of the signal $p(\vec{x})$ and the multiscale kernel $\phi(\vec{x})$ respectively. For global scale analysis of differential signals, Φ is replaced by the multiscale power spectrum ring $|\Psi|^2$ in Equation 3.33 resulting in the DPSNVR:

$$DPSNVR(P;\sigma) = \frac{\rho^2(|P(\vec{\omega})|^2|\Psi(\sigma;\vec{\omega})|^2)}{\rho^2(|\Psi(\sigma;\vec{\omega})|^2)}. \tag{3.36}$$

The scale information can be retrieved by observing local minima of the graph of DPSNVR vs σ. Due to rotational invariance, both $|\Phi|^2$ and $|\Psi|^2$ have center frequency vector $\vec{\omega} = (\omega_1, \ldots, \omega_N)$ at $(0, \ldots, 0)$. For a real signal $f(\vec{x})$, $|F(\vec{\omega})|^2$ also has zero center frequency. Therefore, ρ^2 in Equations 3.35 and 3.36 can be computed from

$$\rho^2(|F(u_1, \ldots, u_N)|^2) = \frac{\int_{\mathbf{R}^N}(u_1^2 + \ldots + u_N^2)|F(u_1, \ldots, u_N)|^2 du_1 \cdots du_N}{\int_{\mathbf{R}^N}|F(u_1, \ldots, u_N)|^2 du_1 \cdots du_N}. \tag{3.37}$$

For a discrete case, the integration in Equation 3.37 is approximated by summation.

3.5 Experimental Results of Global Scale Analysis of 2-Dimensional Multiscale Edge Signals

This section presents examples of global scale analysis of simulated and real images. The 2-dimensional multiscale smoothing kernel used in this experiment is the multiscale Gaussian function defined as:

$$\phi(\sigma; x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{3.38}$$

with the corresponding Fourier transform

$$\Phi(\sigma; u, v) = e^{-\frac{\sigma^2(u^2+v^2)}{2}}. \tag{3.39}$$

The associated multiscale Gaussian power spectrum ring is given by

$$|\Psi(\sigma; u, v)|^2 = (u^2 + v^2)e^{-\sigma^2(u^2+v^2)} \tag{3.40}$$

$$= R^2 e^{-\sigma^2 R^2}, \tag{3.41}$$

where $R = \sqrt{u^2 + v^2}$.

To eliminate the need for computing the Fourier transformation, Ψ is generated directly from Equation 3.41. The DPSNVR is calculated according to Equation 3.36 and the edge characteristic-scale analysis is performed using the multiscale normalized gradient magnitude image, $\hat{e}$, defined by

$$\hat{e}(\sigma; x, y) = \frac{1}{n}\sqrt{p_x(\sigma; x, y)^2 + p_y(\sigma; x, y)^2},$$

where p_x and p_y are the first order derivatives of the smoothed image $p(\sigma; x, y)$ in x and y directions and the normalization factor n is given by

$$n = \max_{x,y} \sqrt{p_x(\sigma; x, y)^2 + p_y(\sigma; x, y)^2}.$$

The edge characteristic-scale functions are computed as follows:

$$\begin{aligned} AVG(\sigma) &= \frac{1}{N \times M} \sum_{x,y} \hat{e}(\sigma; x, y), \\ RMS(\sigma) &= |\frac{1}{N \times M} \sum_{x,y} \hat{e}^2(\sigma; x, y)|^{\frac{1}{2}}, \\ SD(\sigma) &= |RMS(\sigma)^2 - AVG(\sigma)^2|^{\frac{1}{2}}, \end{aligned}$$

where $N \times M$ is the image size. In all experiments, local minima in the curves correspond to the scale values σ.

3.5.1 Simulated Images

The first two examples show two types of multiscale structures, a cluster of objects displayed in Figure 3.14-a and a snowflake shown in Figure 3.16-a. The cluster object can be modeled using the multiplicative model proposed in the previous section while fractal based method is suitable for modeling the snowflake.

The image in Figure 3.14-a has two scales, the small scale of circles and the large scale of the hexagonal cluster. Table 3.5 shows results using the different global scale analysis techniques where two scales are detected. Clearly, there are two local minima in each of the ECS and DPSNVR graphs. Edge images with σ selected from local minima of $SD(\sigma)$ and $DPSNVR(\sigma)$ are displayed in Figure 3.15. Since the values of σ obtained from local minima of $SD(\sigma)$ are too small to separate the 2 different scales, some small details appear in the large scale edge image in Figure 3.15-b. The most accurate edge images in Figure 3.15 are obtained using the local minima of the DPSNVR graph. For example, as illustrated by Figure 3.15-d, small scale edges are completely removed while the hexagonal outline is well preserved.

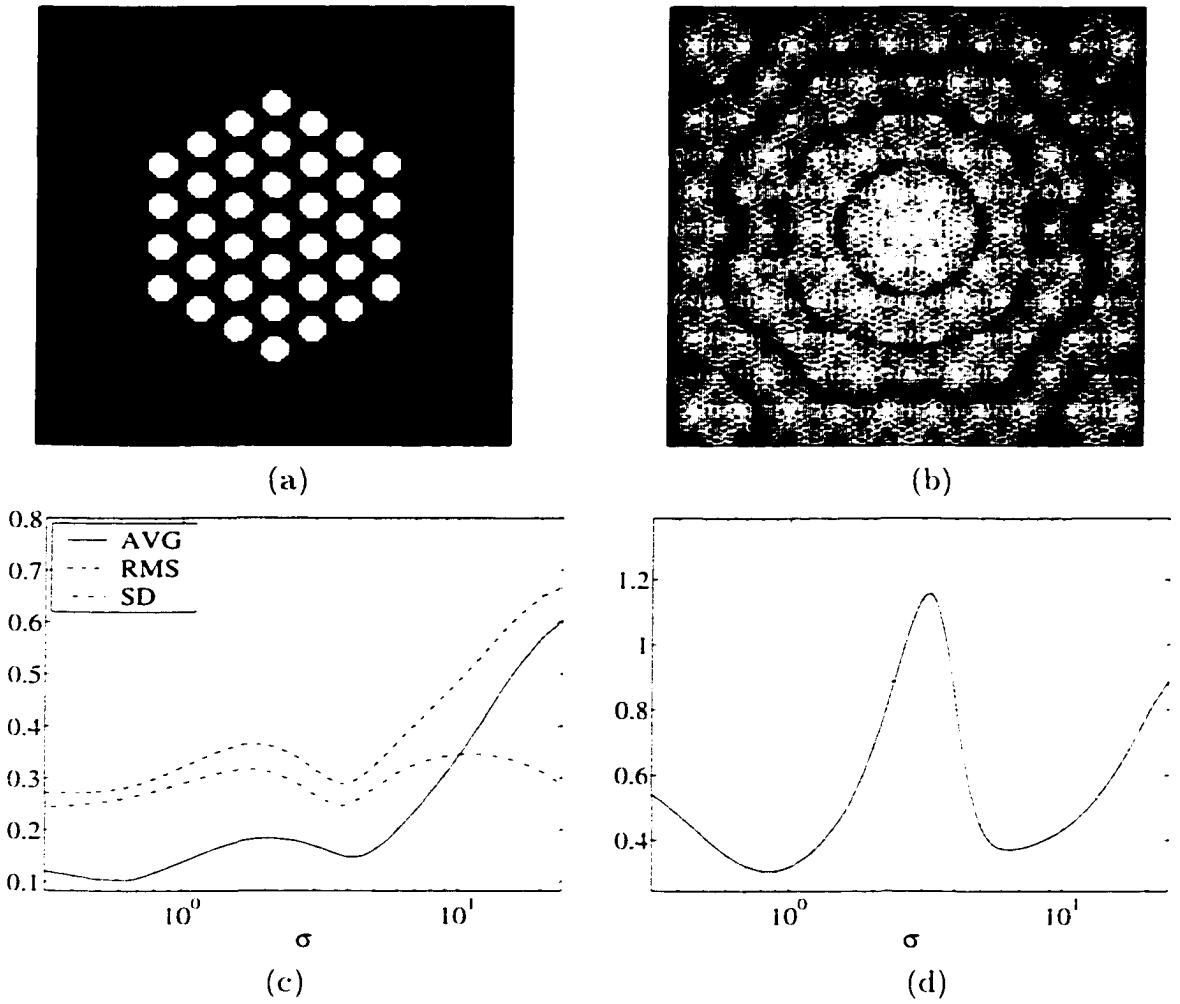


Figure 3.14 Double scale image: (a) original image. (b) log magnitude of the Fourier transform. (c) ECS curves. (d) DPSNR graph.

Table 3.5 Global scale analysis of the image in Figure 3.14.

	Scale 1	Scale 2
σ at local minima of $AVG(\sigma)$	0.62	4.07
σ at local minima of $RMS(\sigma)$	0.34	3.77
σ at local minima of $SD(\sigma)$	0.32	3.77
σ at local minima of $DPSNR(\sigma)$	0.85	6.27

The snowflake shown in Figure 3.16-a is a triple scale tree structure. The global scales detected are summarized in Table 3.6. As seen, only one scale is detected using $AVG(\sigma)$ while the rest of the graphs detect all three scales. Edge images with σ selected from local minima of $SD(\sigma)$ and $DPSNVR(\sigma)$ are displayed in Figure 3.17. Clearly, while edge images in Figures 3.17-a to 3.17-c obtained using σ from the local minima of $SD(\sigma)$ are too detailed, results presented in Figures 3.17-d to 3.17-f obtained using the local minima of the DPSNVR graph represent structures at all 3 scales.

3.5.1.1 Effect of spatial separation of objects

The next three examples in Figures 3.18 through 3.20 show the effect of spatial distance between objects in global scale analysis. In these examples, all images have 144 small squares of size 4×4 pixels while the number of global scales are controlled by varying the separation between the 4×4 pixel squares. For example, with 2 pixel separation in Figure 3.18-a, the image contains only two scales, that of the small square and the whole cluster. The third scale is introduced by increasing separation between the groups of 2×2 squares as seen in Figures 3.19-a and 3.20-a respectively. In Figure 3.18, both scales are detected by all the global scale analysis techniques as shown in Table 3.7. In Figure 3.19, the separation distance between the 2×2 squares increases but is not enough to yield the next scale, and hence is not detected by any of the techniques. As shown in Table 3.8, all ECS methods, $AVG(\sigma)$, $RMS(\sigma)$, and $SD(\sigma)$, miss the largest scale while the method based on $DPSNVR(\sigma)$ misses the middle scale. However, a change in the slope of the DPSNVR graph can be seen around $\sigma = 3.6$. In the graph of $\frac{d(DPSNVR(z))}{dz}$ in Figure 3.19-e, where z is the natural scale parameter, $z = \ln\sigma$, the additional local maximum point can be seen at $\sigma = 3.6$. This feature is due to the appearance of the middle scale. Hence, the presence of weak scales can be detected using the derivative $\frac{d(DPSNVR(z))}{dz}$.

$$\sigma_{weak} = \begin{cases} & \sigma \text{ at local minima of } \frac{d(DPSNVR(z))}{dz}, \quad \frac{d(DPSNVR(z))}{dz} > 0 \\ \text{or} & \\ & \sigma \text{ at local maxima of } \frac{d(DPSNVR(z))}{dz}, \quad \frac{d(DPSNVR(z))}{dz} < 0 \end{cases} \tag{3.42}$$

As the the separation distance increases, the middle scale becomes more distinct as seen in Figure 3.20-a. In this case, all three scales can be detected by the graphs $SD(\sigma)$ and $DPSNVR(\sigma)$. Corresponding changes in $\frac{d(DPSNVR(z))}{dz}$ can also be observed in Figures 3.18-e, 3.19-e, and 3.20-e where the middle scale is indicated at the local extrema at $\sigma = 3.62$ in Figure 3.19-e. Edge images using σ listed in Table 3.9 are shown

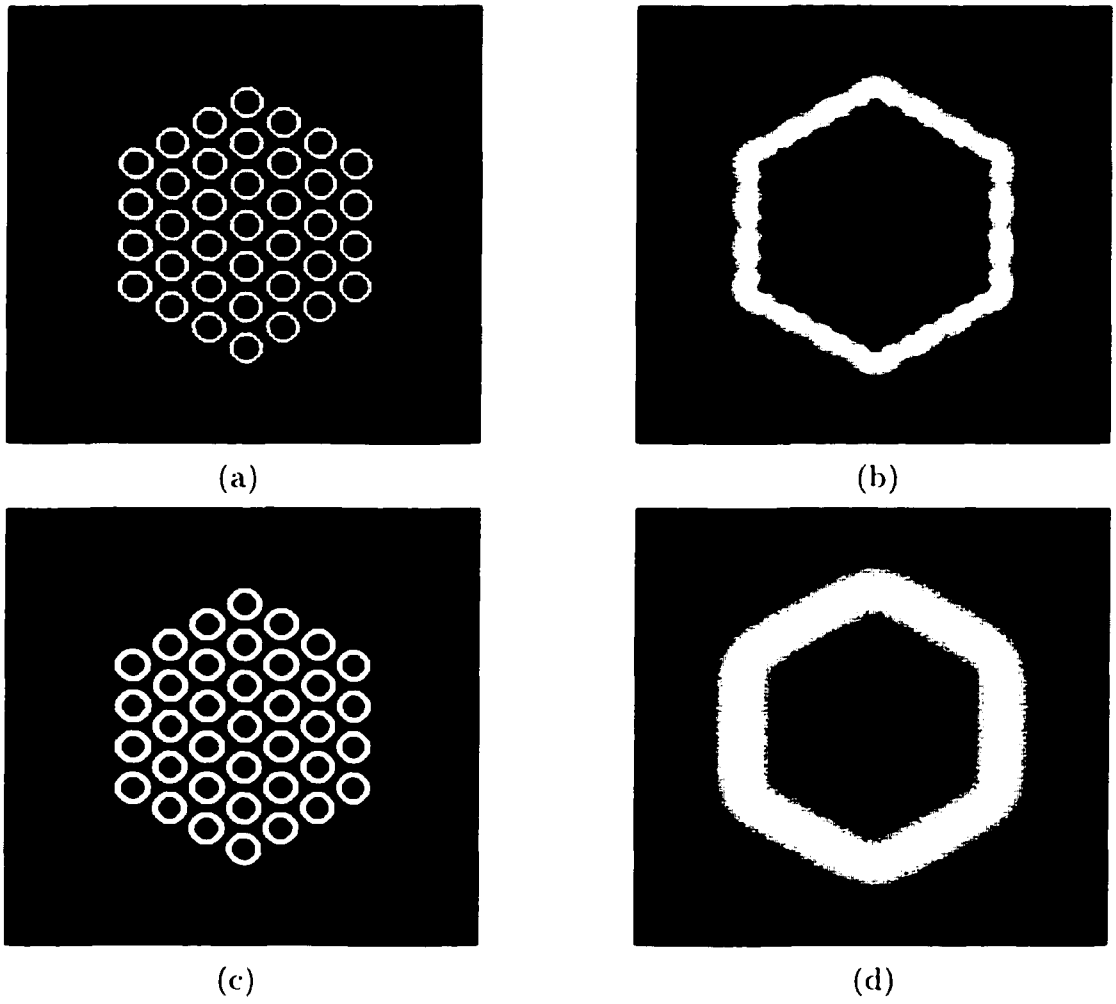


Figure 3.15 Edge images with σ obtained from Table 3.5: local minima of $SD(\sigma)$: (a) $\sigma = 0.32$. (b) $\sigma = 3.77$: local minima of the DPSNVR graph: (c) $\sigma = 0.85$. (d) $\sigma = 6.27$.

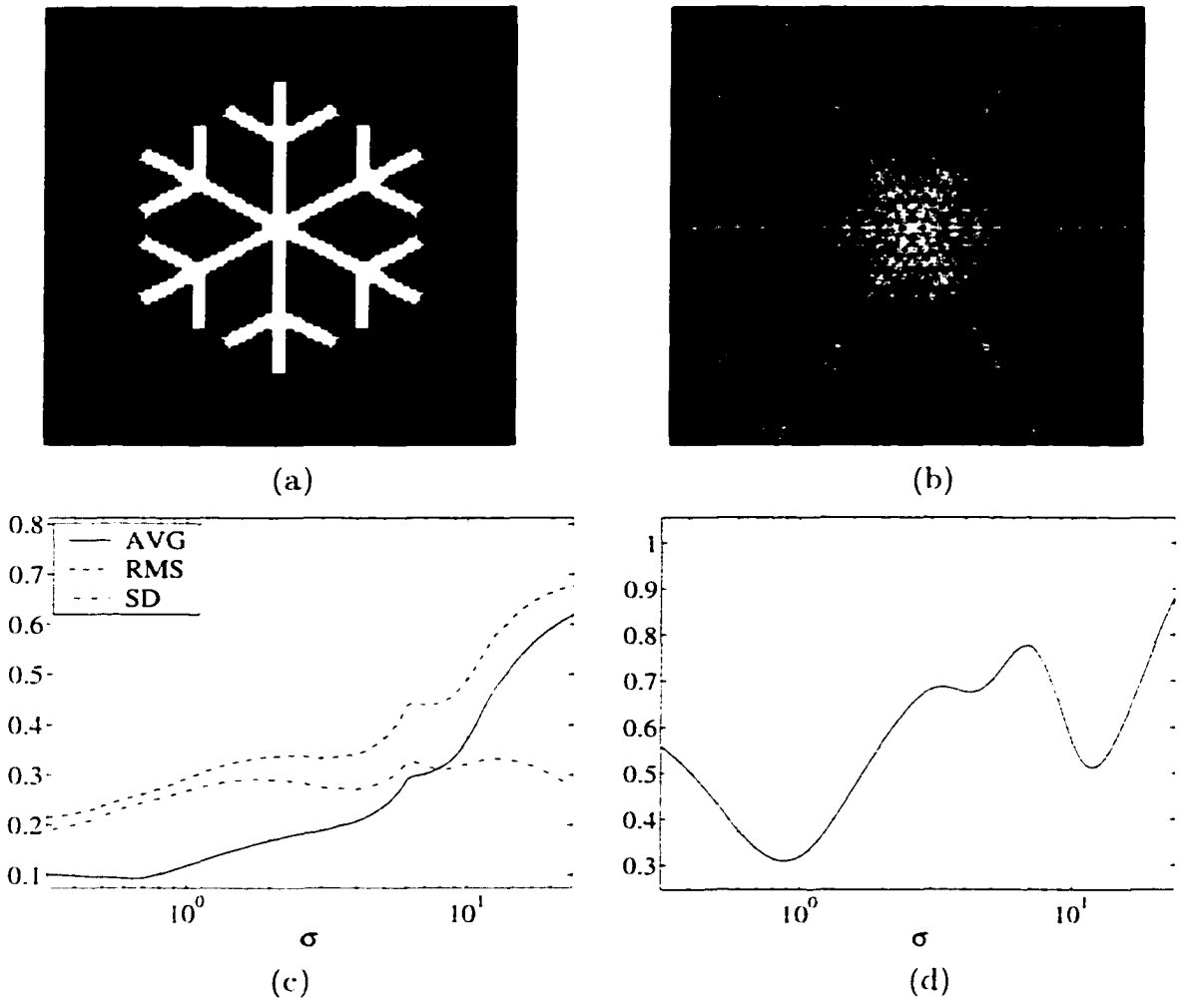


Figure 3.16 Triple scale image: (a) original image. (b) log magnitude of the Fourier transform. (c) ECS curves. (d) DPSNVR graph.

Table 3.6 Global scale analysis of the image in Figure 3.16.

	Scale 1	Scale 2	Scale 3
σ at local minima of $AVG(\sigma)$	0.64		
σ at local minima of $RMS(\sigma)$	0.32	3.10	7.34
σ at local minima of $SD(\sigma)$	0.32	3.35 3.92	8.25
σ at local minima of $DPSNVR(\sigma)$	0.88	4.07	11.75

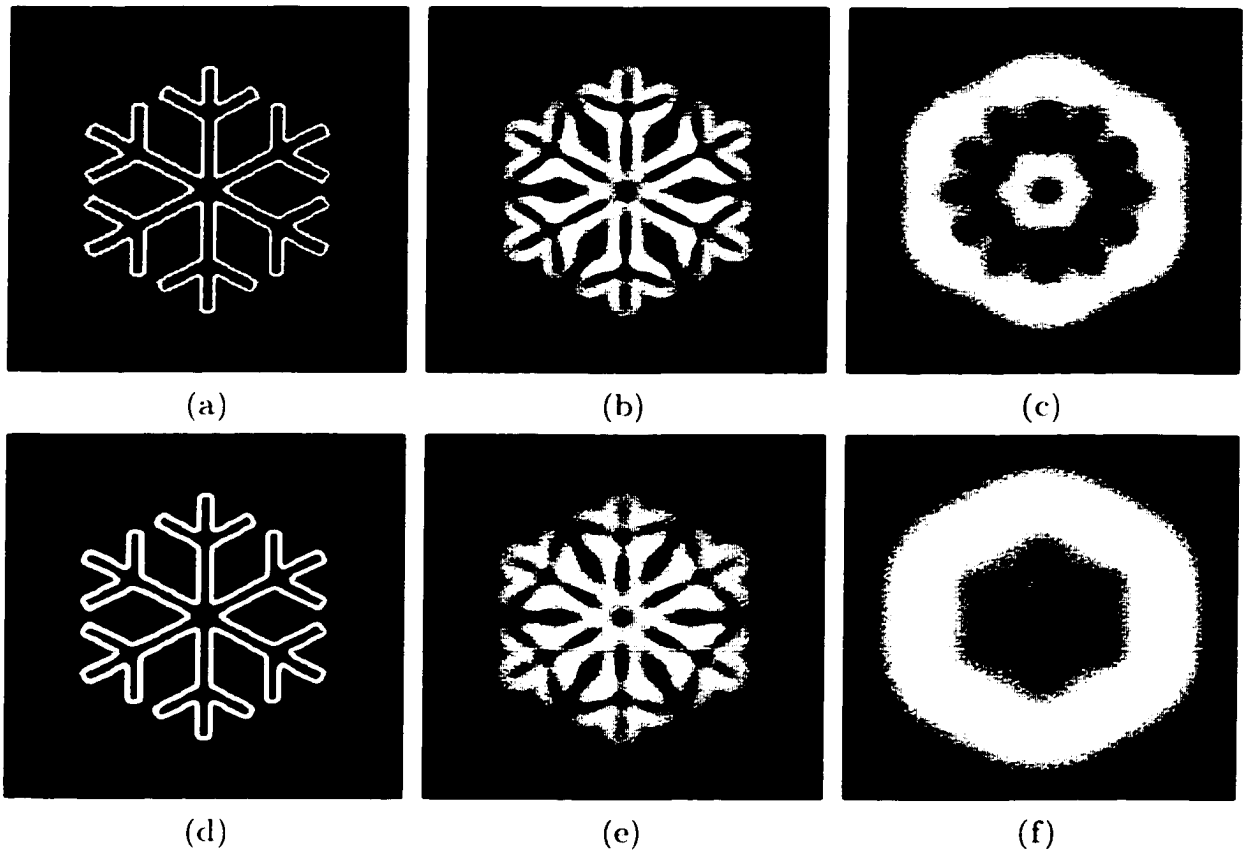


Figure 3.17 Edge images with σ obtained from Table 3.6: local minima of $SD(\sigma)$: (a) $\sigma = 0.32$. (b) $\sigma = 3.35$. (c) $\sigma = 8.25$: local minima of the DPSNVR graph: (d) $\sigma = 0.88$. (e) $\sigma = 4.07$. (f) $\sigma = 11.75$.

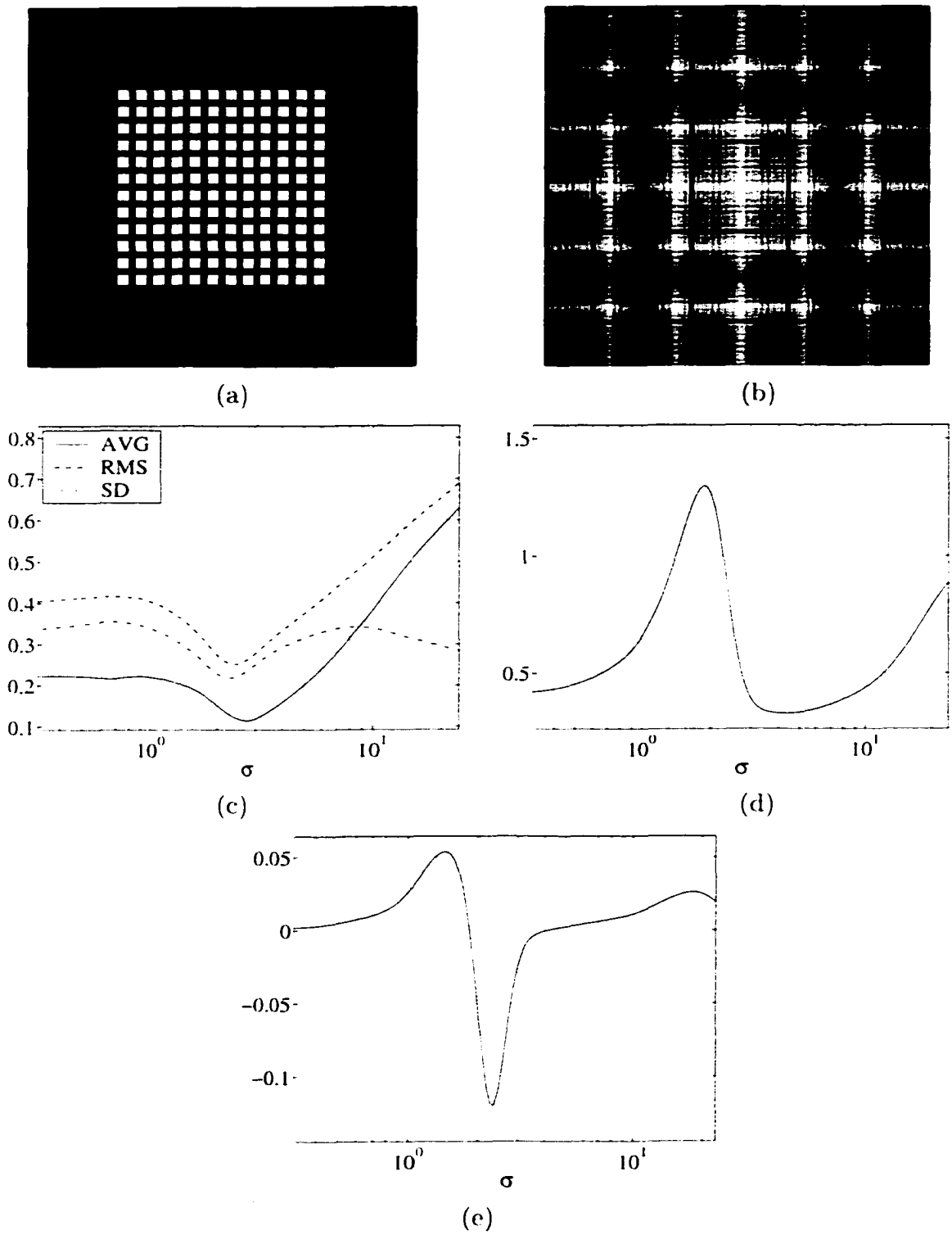


Figure 3.18 (a) Original image. (b) log magnitude of the Fourier transform. (c) ECS curves. (d) DPSNVR graph. (e) $\frac{d(DPSNVR(z))}{dz}$.

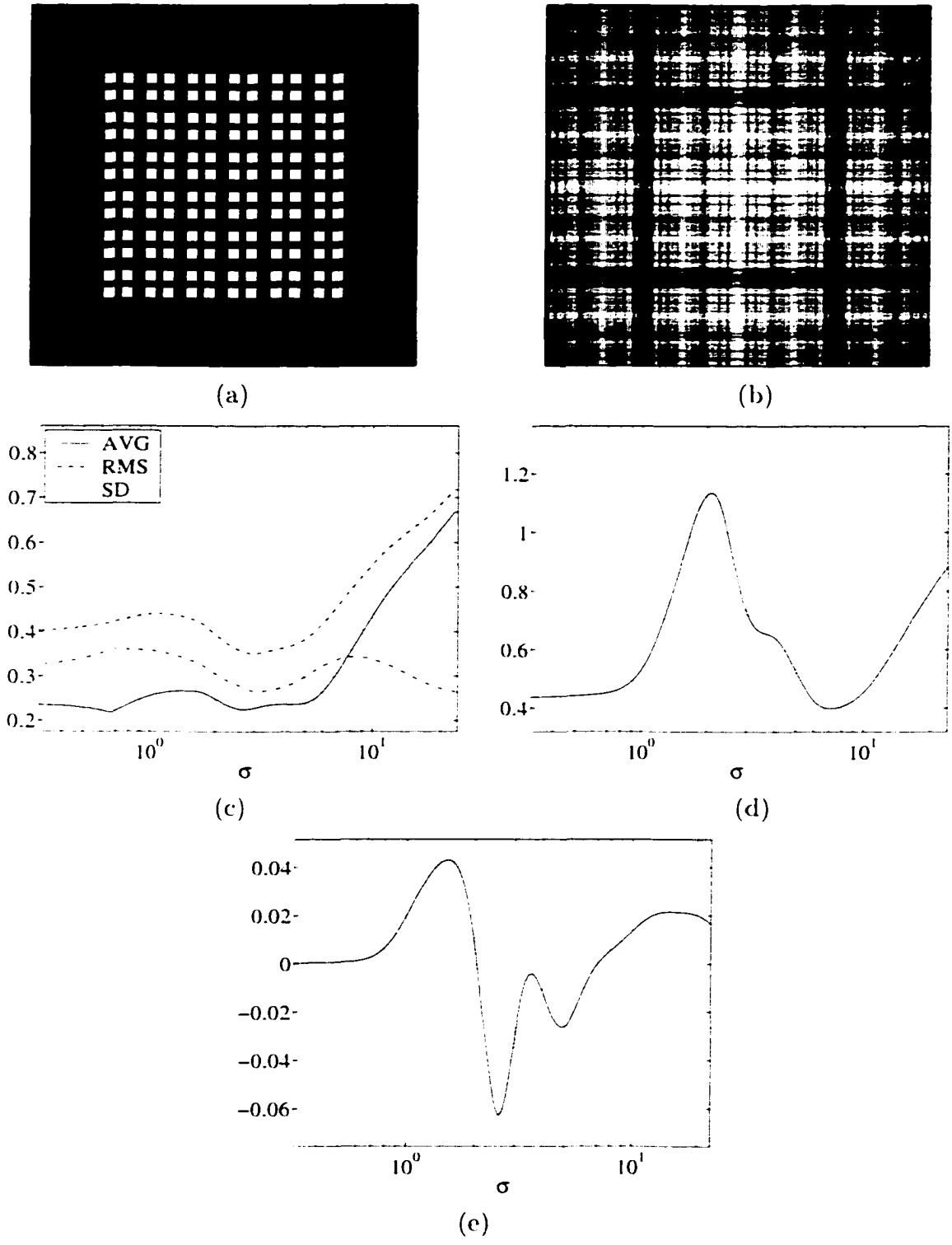


Figure 3.19 (a) Original image. (b) log magnitude of the Fourier transform. (c) ECS curves. (d) DPSNVR graph. (e) $\frac{d(DPSNVR(z))}{dz}$.

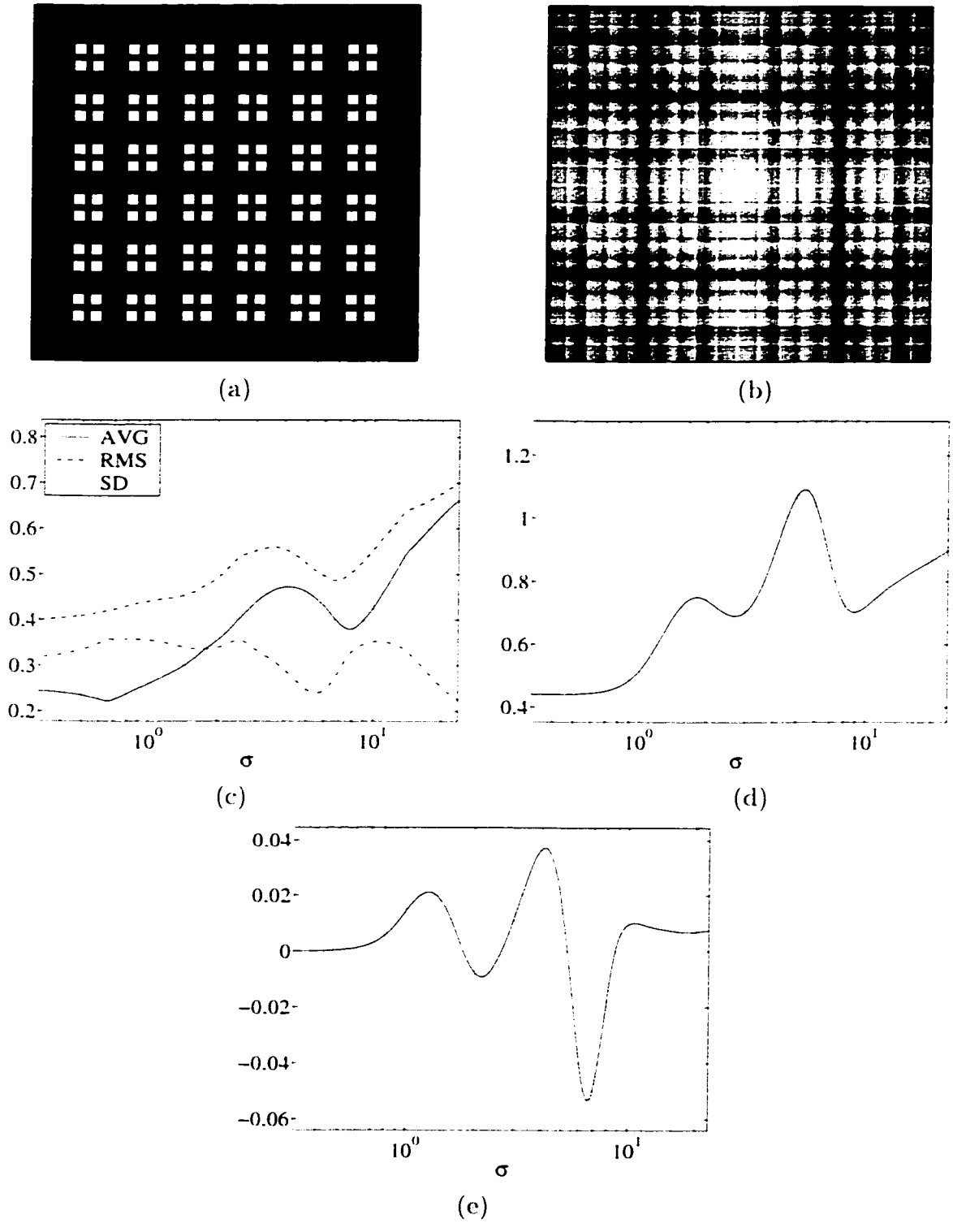


Figure 3.20 (a) Original image, (b) log magnitude of the Fourier transform. (c) ECS curves. (d) DPSNVR graph, (e) $\frac{d(DPSNVR(z))}{dz}$.

Table 3.7 Global scale analysis of the image in Figure 3.18.

	Scale 1	Scale 2
σ at local minima of $AVG(\sigma)$	0.32 0.64	2.65
σ at local minima of $RMS(\sigma)$	0.32	2.35
σ at local minima of $SD(\sigma)$	0.32	2.26
σ at local minima of $DPSNR(\sigma)$	0.32	4.24

Table 3.8 Global scale analysis of the image in Figure 3.19.

	Scale 1	Scale 2	Scale 3
σ at local minima of $AVG(\sigma)$	0.67	2.65 4.24	
σ at local minima of $RMS(\sigma)$	0.32	2.98	
σ at local minima of $SD(\sigma)$	0.32	3.10	
σ at local minima of $DPSNR(\sigma)$	0.32		7.05
σ_{weak} of $\frac{d(DPSNR(z))}{dz}$		3.62	

Table 3.9 Global scale analysis of the image in Figure 3.20.

	Scale 1	Scale 2	Scale 3	
σ at local minima of $AVG(\sigma)$	0.64		7.94	
σ at local minima of $RMS(\sigma)$	0.32		6.78	
σ at local minima of $SD(\sigma)$	0.32	1.65 1.79	5.58	
σ at local minima of $DPSNR(\sigma)$	0.34	2.65	8.93	
σ_{weak} of $\frac{d(DPSNR(z))}{dz}$				18.08

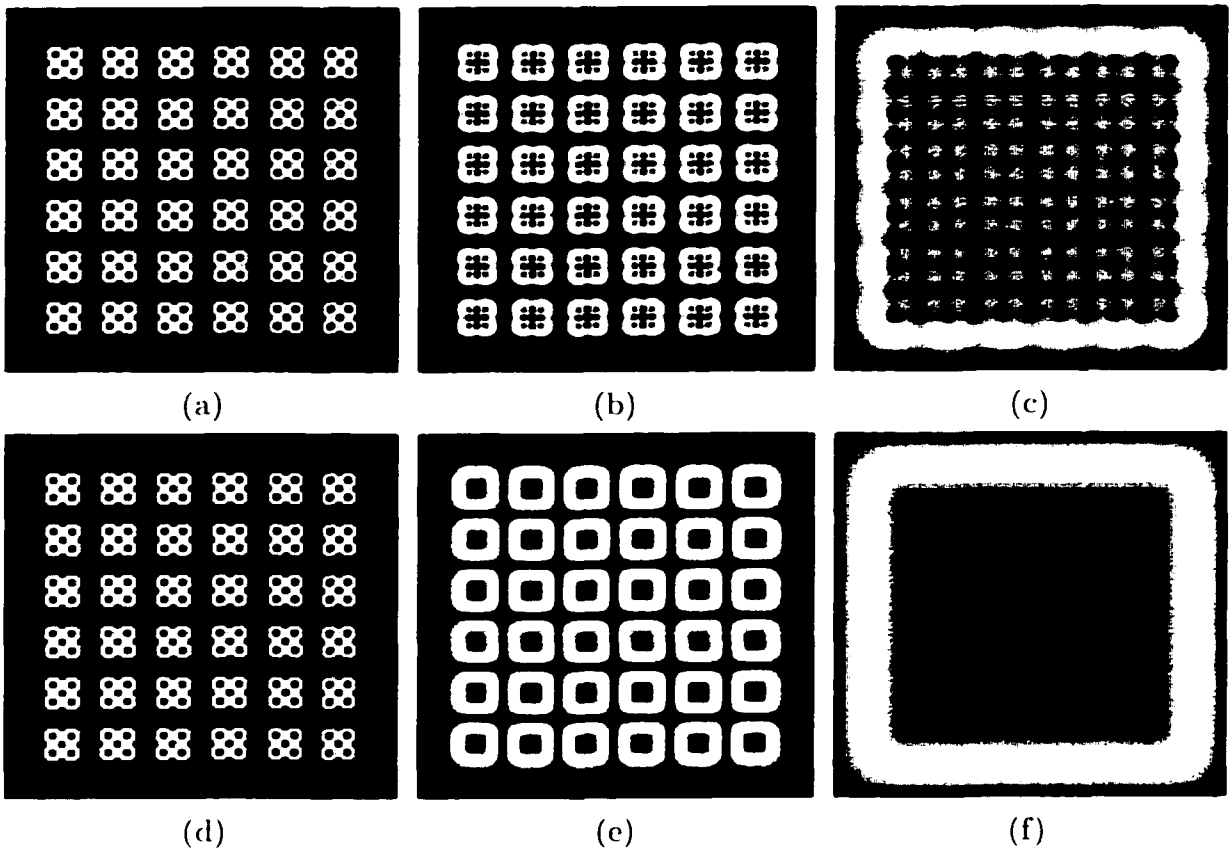


Figure 3.21 Edge images with σ obtained from Table 3.9: local minima of $SD(\sigma)$: (a) $\sigma = 0.32$, (b) $\sigma = 1.65$, (c) $\sigma = 5.58$: local minima of the DPSNVR graph: (d) $\sigma = 0.34$, (e) $\sigma = 2.65$, (f) $\sigma = 8.93$.

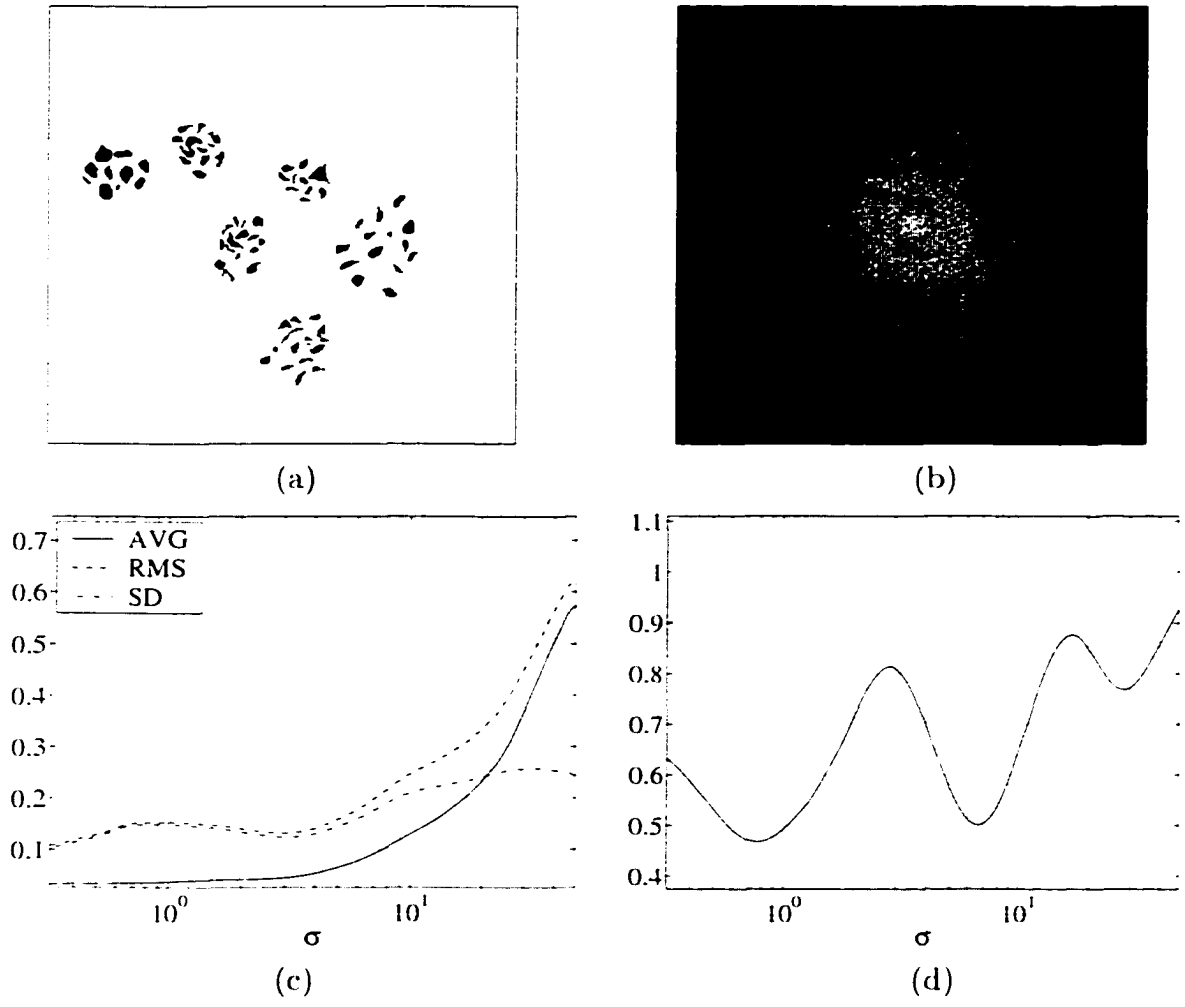


Figure 3.22 (a) Original image. (b) log magnitude of the Fourier transform. (c) ECS curves. (d) DPSNVR graph.

Table 3.10 Global scale analysis of the image in Figure 3.22.

	Scale 1	Scale 2	Scale 3
σ at local minima of $AVG(\sigma)$	0.32 0.47 0.53		
σ at local minima of $RMS(\sigma)$	0.32 2.98		
σ at local minima of $SD(\sigma)$	0.32 2.98		
σ at local minima of $DPSNVR(\sigma)$	0.75	6.52	27.84

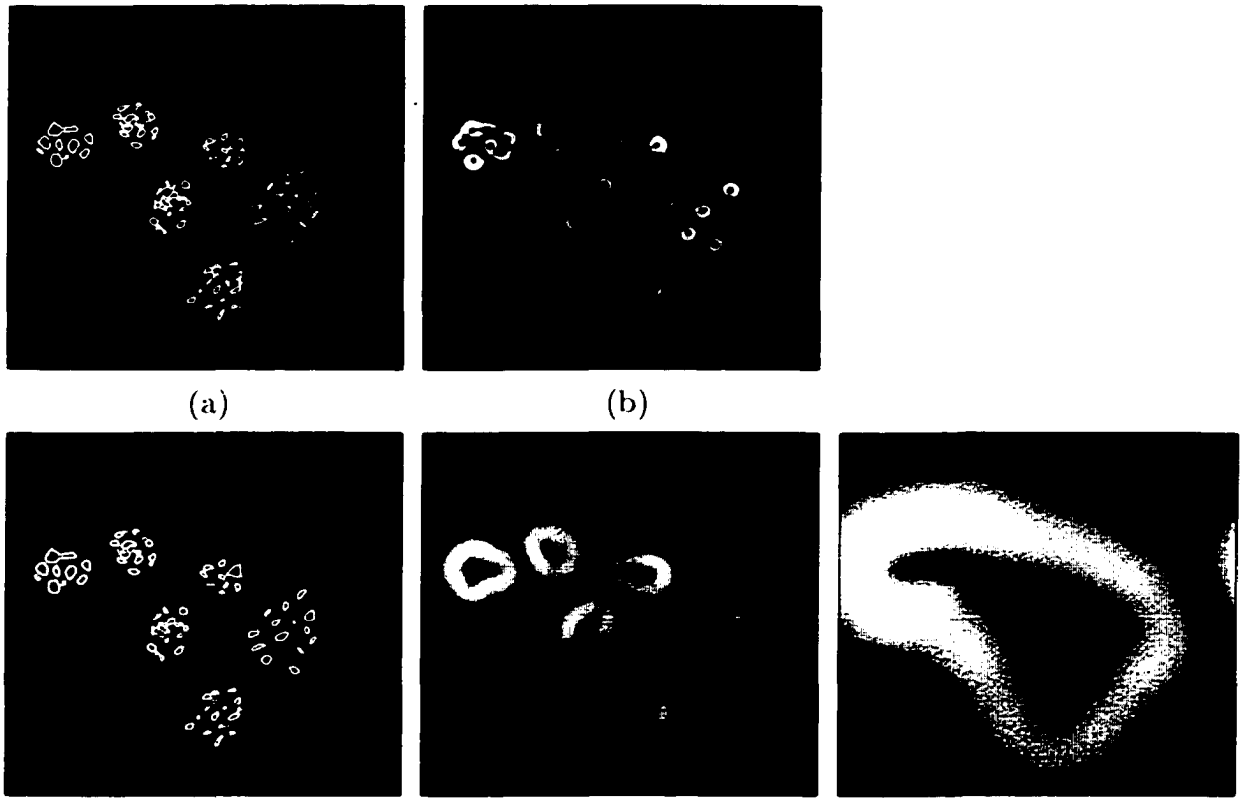


Figure 3.23 Edge images with σ obtained from Table 3.11: local minima of $SD(\sigma)$: (a) $\sigma = 0.32$, (b) $\sigma = 2.98$: local minima of the DPSNVR graph: (c) $\sigma = 0.75$, (d) $\sigma = 6.52$, (e) $\sigma = 27.84$.

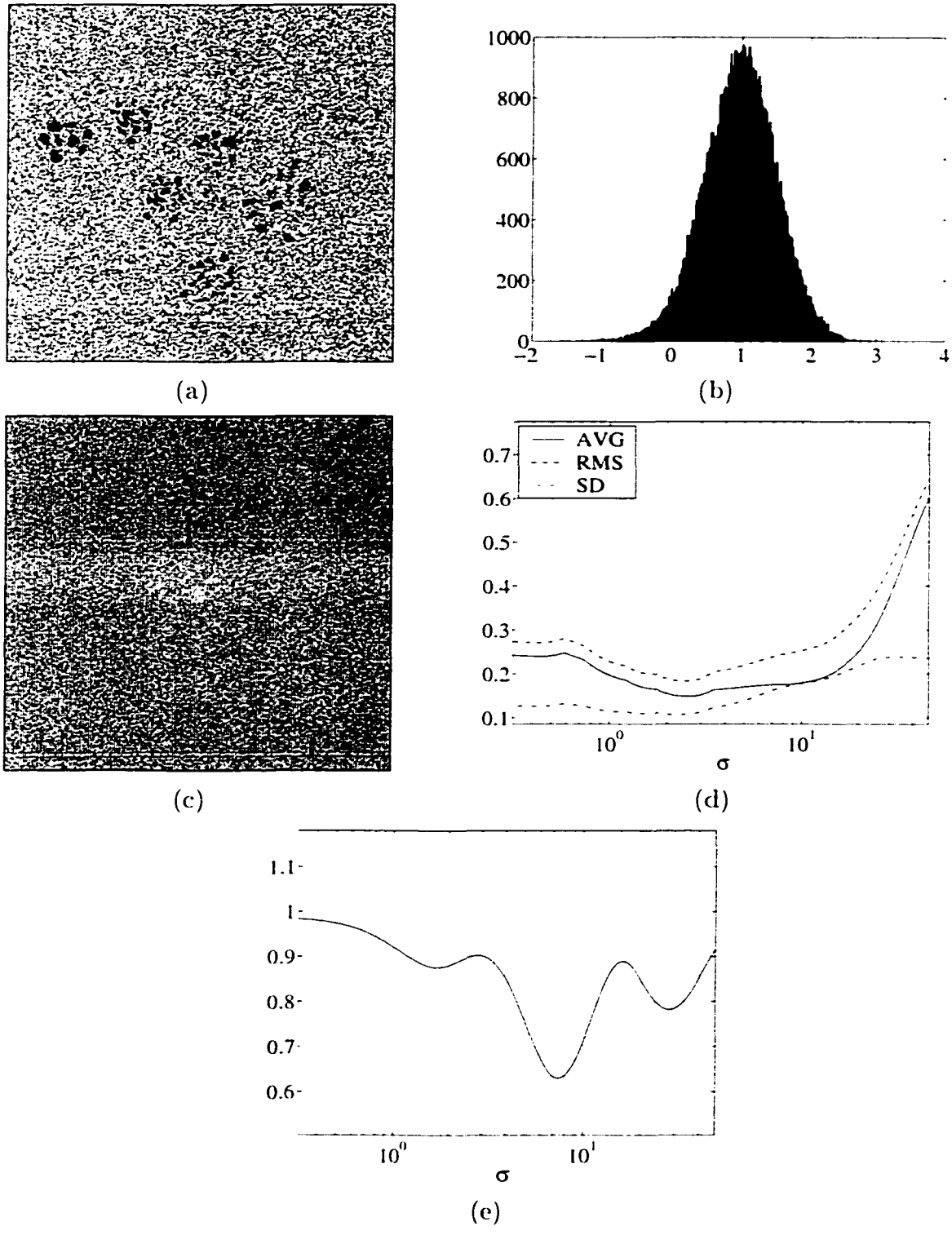
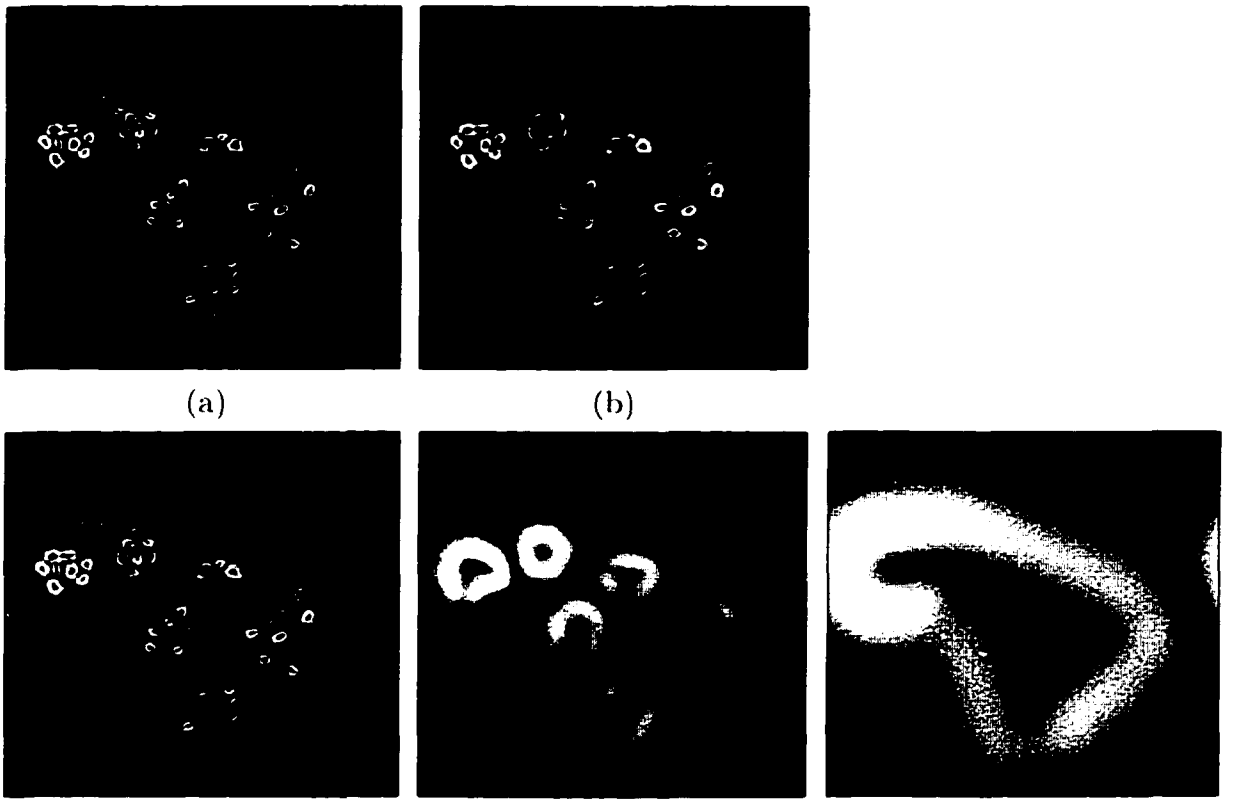


Figure 3.24 (a) Degraded image of the image in Figure 3.14-a with SNR = 2. (b) intensity histogram. (c) log magnitude of the Fourier transform. (d) ECS curves. (e) DPSNVR graph.

Table 3.11 Global scale analysis of the image in Figure 3.24.

	Scale 1	Scale 2	Scale 3
σ at local minima of $AVG(\sigma)$	0.44 2.54 3.62	7.94	
σ at local minima of $RMS(\sigma)$	0.42 2.54		
σ at local minima of $SD(\sigma)$	0.40 1.12 1.47 2.18		
σ at local minima of $DPSNVR(\sigma)$	1.72	7.34	27.84



(a) (b) (c) (d) (e)

Figure 3.25 Edge images with σ obtained from Table 3.11: local minima of $SD(\sigma)$: (a) $\sigma = 1.47$, (b) $\sigma = 2.18$; local minima of the DPSNVR graph: (f) $\sigma = 1.72$, (g) $\sigma = 7.34$, (h) $\sigma = 27.84$.

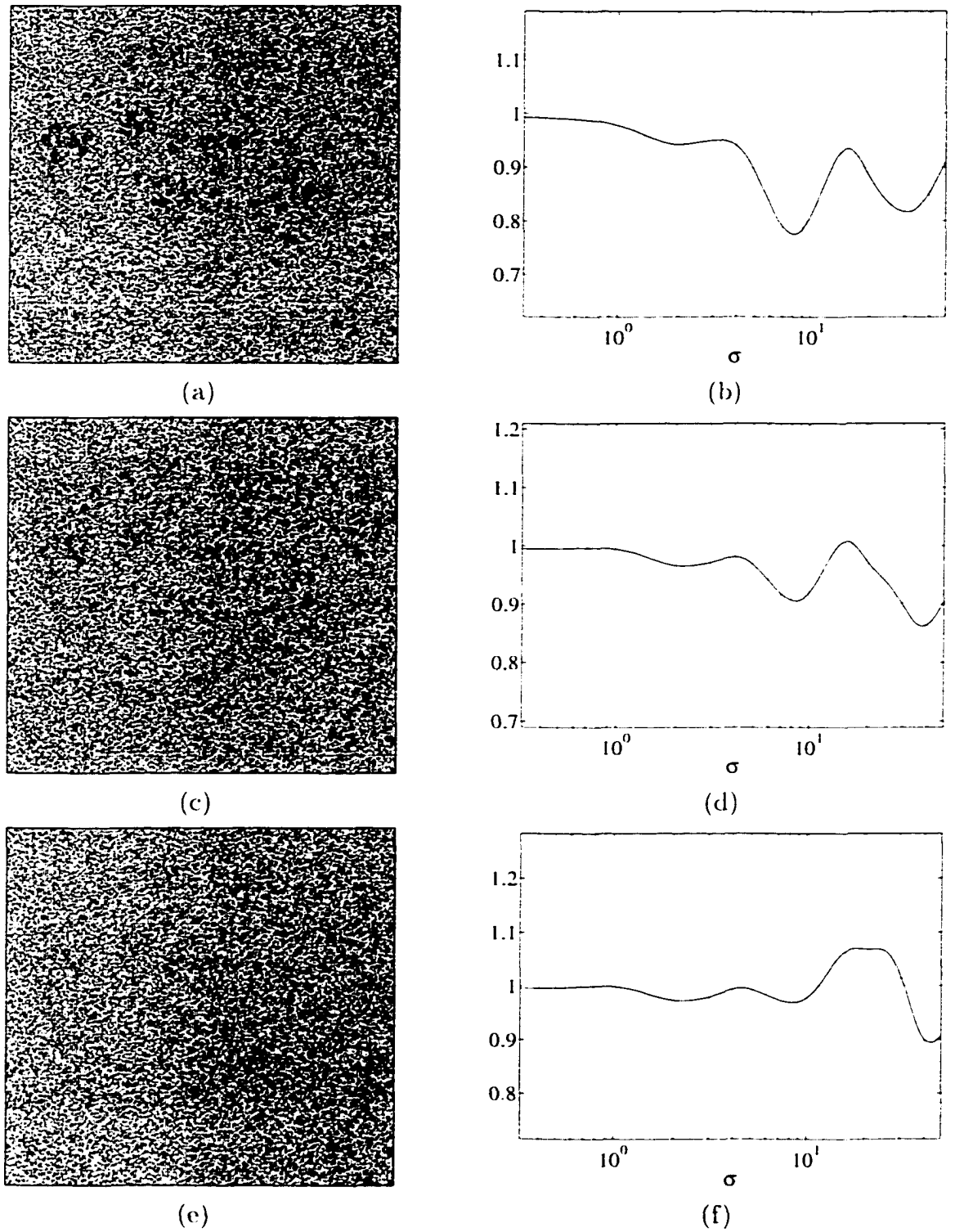


Figure 3.26 Degraded images of the image in Figure 3.22-a and the corresponding DPSNVR curves: (a) and (b) SNR = 1, (c) and (d) SNR = 0.5, (e) and (f) SNR = 0.25.

in Figure 3.21. As in Figure 3.14, edge images in Figures 3.21-d to 3.21-f with σ obtained using local minima of the DPSNVR graph are the most accurate.

3.5.1.2 Effect of Noise on Global Scale Analysis

The next three examples in Figures 3.22, 3.24, and 3.26 demonstrate the effect of noise on the global scale analysis. In these examples, signal to noise ratio is defined as

$$SNR = \frac{I_{obj} - I_{bgnd}}{\sigma_{noise}},$$

where I_{obj} and I_{bgnd} are object and background intensities and σ_{noise} is the standard deviation of noise. The image in Figure 3.24-a is the noisy version of the image in Figure 3.22-a degraded by additive white Gaussian noise with $SNR = 2.0$. The unimodal histogram of the degraded image shown in Figure 3.24-b illustrates that object and background pixels are heavily corrupted by noise and cannot be separated using histogram analysis. Tables 3.10 and 3.11 report the scales detected by the 4 methods. $DPSNVR(\sigma)$ detects all three scales in Figure 3.22 while the ECS methods miss the second and third scales. Edge images obtained using σ at local minima of $SD(\sigma)$ and $DPSNVR(\sigma)$ are shown in Figure 3.23. Clearly, edge images in Figures 3.23-c to 3.23-e are best at presenting structures at 3 scales. Table 3.11 reports scales detected in the noisy image in Figure 3.24-a. In this case, the ECS methods still miss the largest scale while there are several false indications of smaller scales. Remarkably, in this example, the most accurate results are obtained using the DPSNVR graph where all scales are detected without any false indication. This is due to the fact that frequency components of noise distributed over the entire spectrum have little effect on the computation of the second order normalized central moment of the spectrum of the image. This example indicates the superior performance of the DPSNVR based global scale analysis.

In the case of very low signal to noise ratios, the method based on DPSNVR still performs consistently as demonstrated in Figure 3.26 where DPSNVR curves corresponding to all degraded images with various signal to noise ratios reveal similar patterns. In all examples in Figure 3.26, 3 existing scales are detected even though objects in images can barely be seen. The results of global scale analysis at various signal to noise ratios are summarized in Table 3.12. Note that, as signal to noise ratio decreases, false indications of global scales are produced.

Table 3.12 σ at local minima of DPSNVR curves in Figures 3.22, 3.24, and 3.26.

		Scale 1	Scale 2		Scale 3
Figure 3.22-d (SNR = ∞)		0.75	6.52		27.84
Figure 3.24-e (SNR = 2)		1.72	7.34		27.84
Figure 3.26-b (SNR = 1)		2.01	7.94		30.11
Figure 3.26-d (SNR = 0.5)	0.42	2.18	8.25		38.10
Figure 3.26-f (SNR = 0.25)	0.36	2.18	8.25	21.16	42.86

3.5.2 Results of Implementation on Natural Images

The images of living things such as trees and cells are good examples of natural multiscale structures. The first example is an image of a tree with 3 distinct scales as shown in Figure 3.27-a. Results obtained using the 4 methods are summarized in Table 3.13. It is seen that only DPSNVR based method detects all three scales. Edge images with σ obtained from local minima of $SD(\sigma)$ and $DPSNVR(\sigma)$ are shown in Figure 3.28. As demonstrated by Figures 3.28-c to 3.28-e, edge images with σ obtained from $DPSNVR(\sigma)$ clearly represent structures at 3 scales. The next example in Figure 3.29-a is an image of bone marrow cells. Unlike previous examples, the number of scales in this image is not easily to discern. Table 3.14 displays the corresponding results. Edge images with σ obtained from local minima of $SD(\sigma)$ and $DPSNVR(\sigma)$ are displayed in Figure 3.30. As seen in Table 3.14, all methods detect Scales 1 and 2. However, only the edge image in Figure 3.30-e with $\sigma = 2.09$ obtained from the local minimum point of $DPSNVR(\sigma)$ clearly exhibits boundaries of the white cells.

The next example shown in Figure 3.31-a is the image of a printed page where the scales range from characters and lines to paragraphs and columns. In terms of separation distances, the number of scales in this example is not obvious. However, from the results displayed in Table 3.15, scales of characters, paragraphs, and columns are detected using the DPSNVR and $\frac{d(DPSNVR(z))}{dz}$ graphs. Edge images at detected scales are shown in Figure 3.32. The last example shown in Figure 3.33-a is the CT scan of a cylindrical soil sample where grey areas represent soil porosity. Table 3.16 shows results of global scale analysis where 3 scales are detected from the DPSNVR and $\frac{d(DPSNVR(z))}{dz}$ graphs. Clearly, structures at different scales including details of soil porosity are reflected by edge images in Figure 3.34.

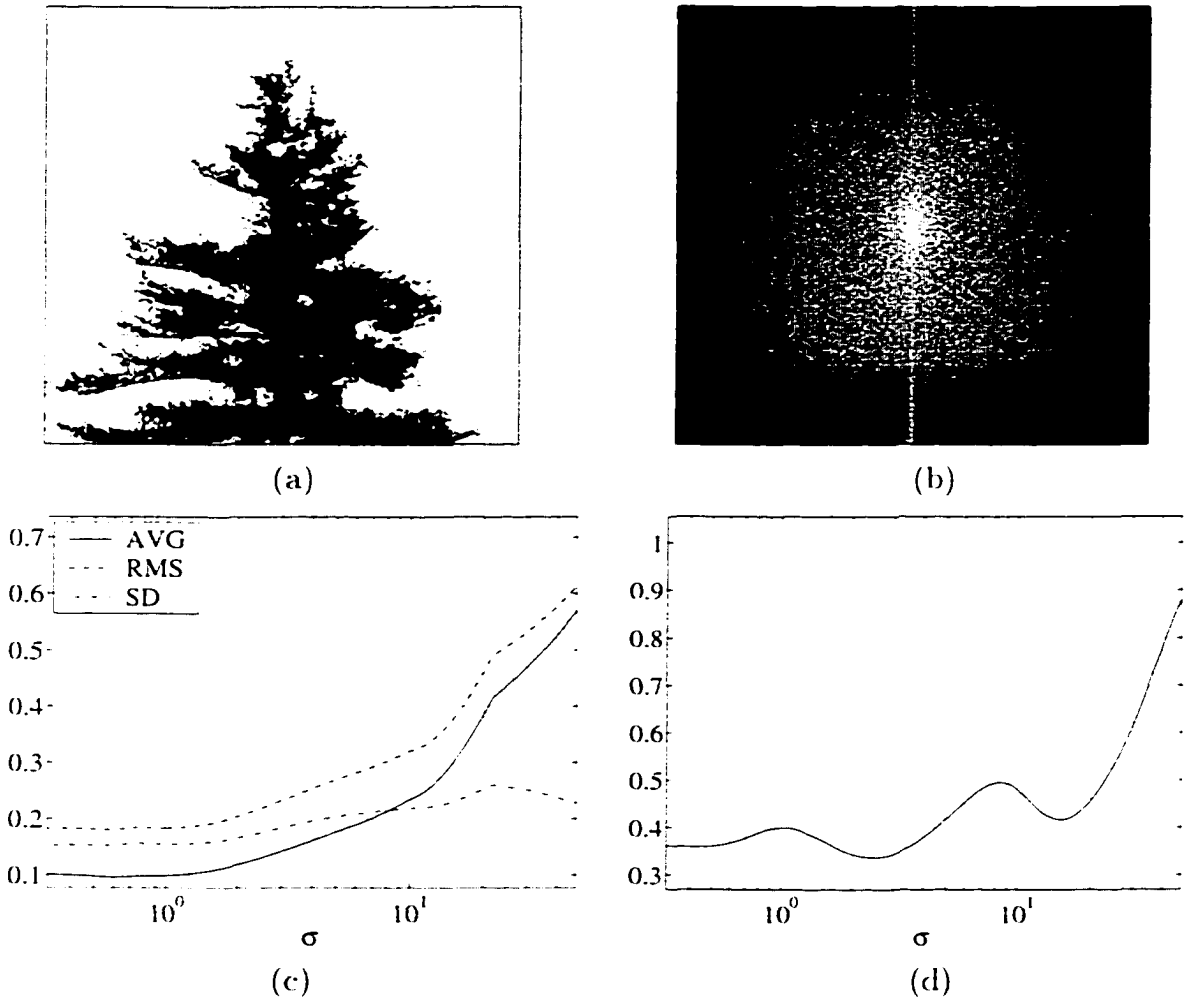


Figure 3.27 (a) Original image. (b) log magnitude of the Fourier transform. (c) ECS curves. (d) DPSNVR graph.

Table 3.13 Global scale analysis of the image in Figure 3.27.

	Scale 1	Scale 2	Scale 3
σ at local minima of $AVG(\sigma)$	0.62 0.82		
σ at local minima of $RMS(\sigma)$	0.33 0.57 0.95		
σ at local minima of $SD(\sigma)$	0.32 0.53 1.03		10.44
σ at local minima of $DPSNVR(\sigma)$	0.42	2.45	14.86

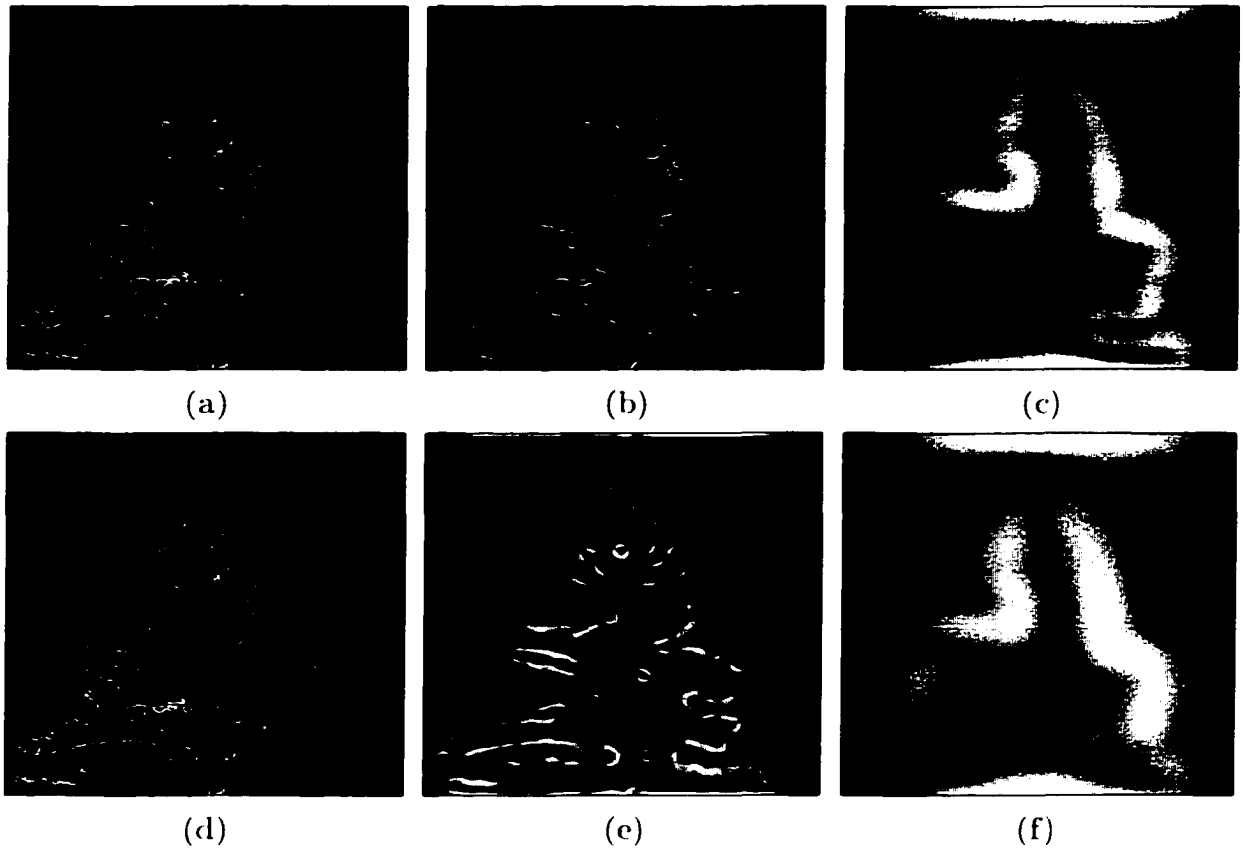


Figure 3.28 Edge images with σ obtained from Table 3.13: local minima of $SD(\sigma)$: (a) $\sigma = 0.53$. (b) $\sigma = 1.03$. (c) $\sigma = 10.44$: local minima of the DPSNVR graph: (f) $\sigma = 0.42$. (g) $\sigma = 2.45$. (h) $\sigma = 14.86$.

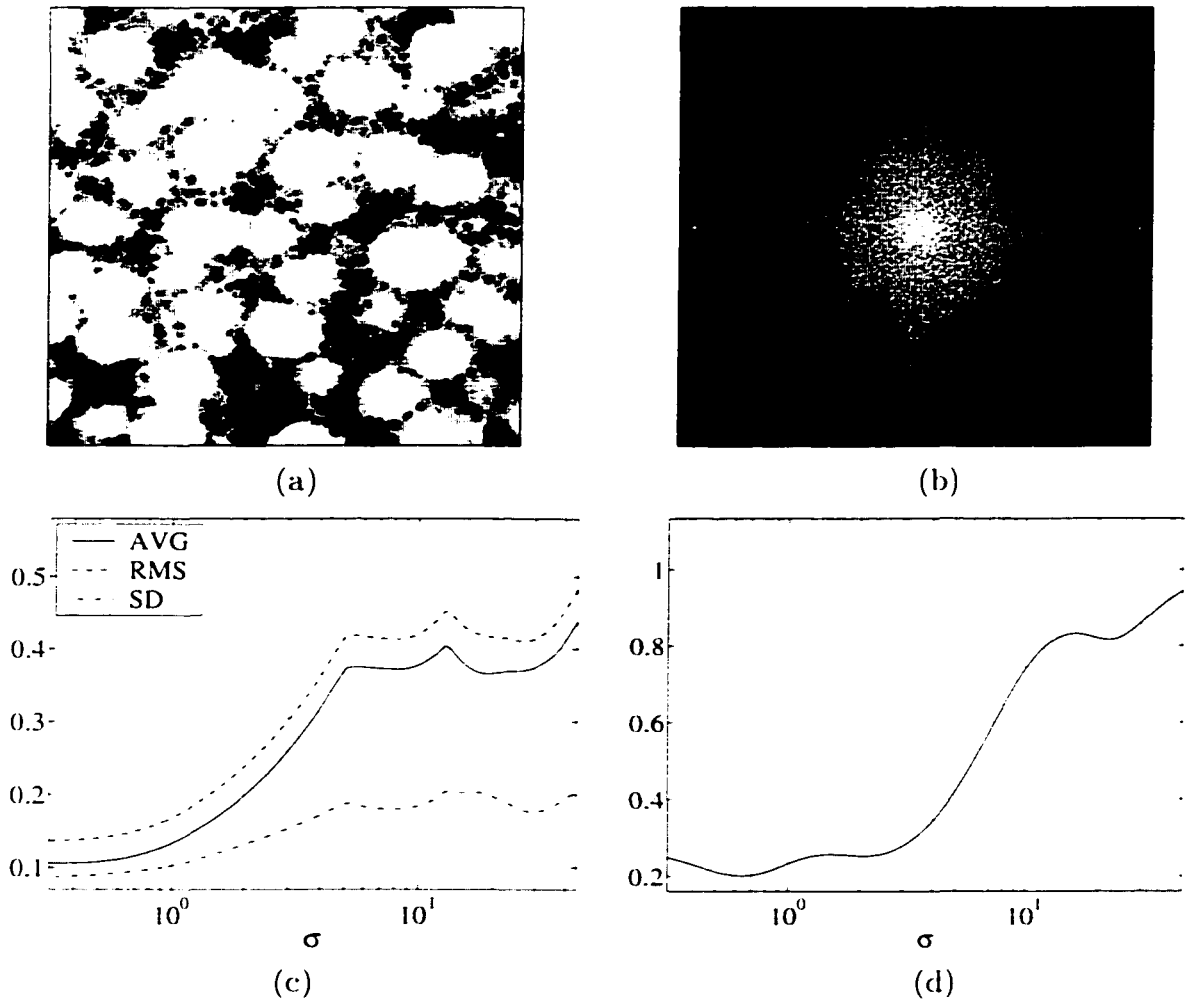


Figure 3.29 (a) Original image, (b) log magnitude of the Fourier transform, (c) ECS curves, (d) DPSNVR graph.

Table 3.14 Global scale analysis of the image in Figure 3.29.

..	Scale 1	Scale 2	Scale 3	
σ at local minima of $AVG(\sigma)$	0.33	8.25	18.81	
σ at local minima of $RMS(\sigma)$	0.32	7.94		26.77
σ at local minima of $SD(\sigma)$	0.32	7.63	14.86	28.95
σ at local minima of $DPSNVR(\sigma)$	0.64	2.09	22.00	

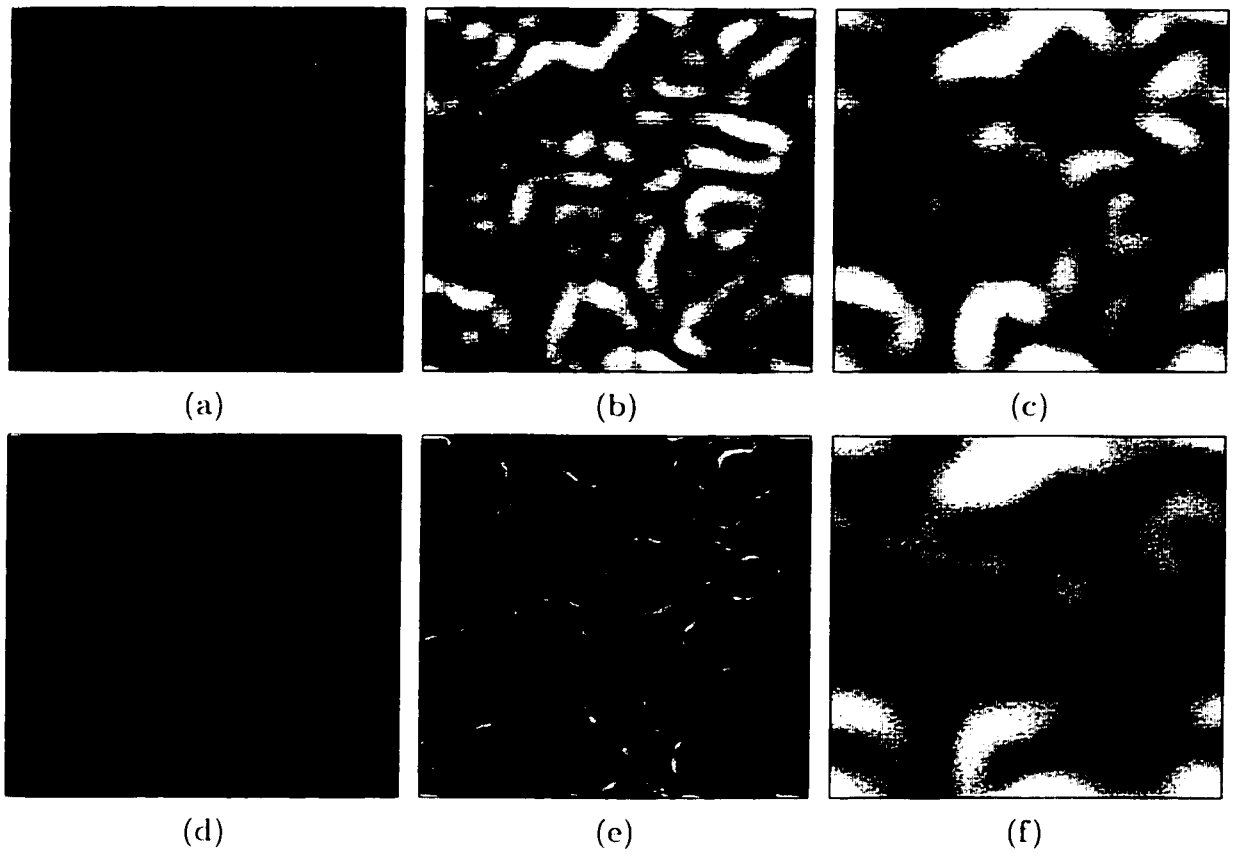


Figure 3.30 Edge images with σ obtained from Table 3.14: local minima of $SD(\sigma)$: (a) $\sigma = 0.32$. (b) $\sigma = 7.63$. (c) $\sigma = 14.86$: local minima of the DPSNR graph: (d) $\sigma = 0.64$. (e) $\sigma = 2.09$. (f) $\sigma = 22.0$.

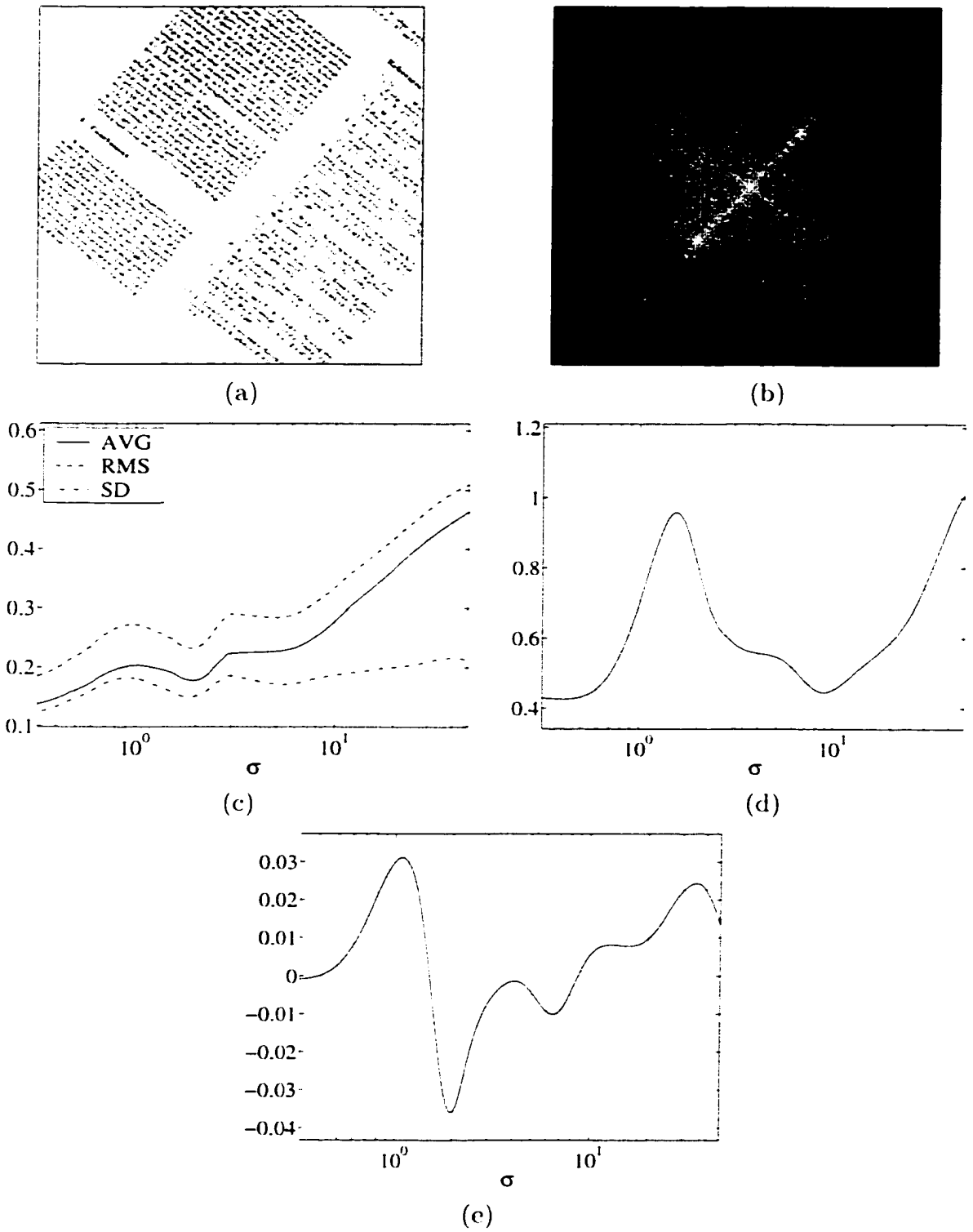
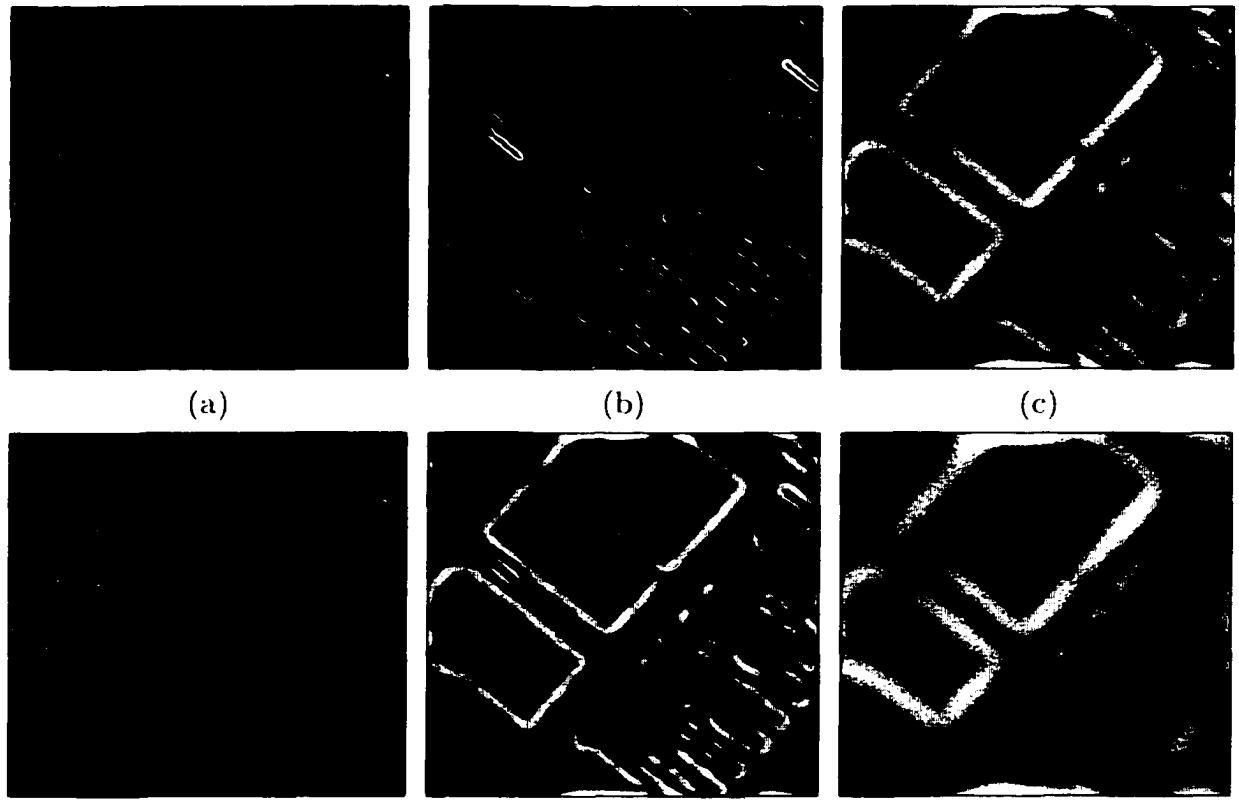


Figure 3.31 (a) Original image. (b) log magnitude of the Fourier transform. (c) ECS curves. (d) DPSNVR graph. (e) $\frac{d(DPSNVR(z))}{dz}$.

Table 3.15 Global scale analysis of the image in Figure 3.31.

	Scale 1	Scale 2	Scale 3	Scale 4
σ at local minima of $AVG(\sigma)$	0.32	1.93		
σ at local minima of $RMS(\sigma)$	0.32	1.86 4.96		
σ at local minima of $SD(\sigma)$	0.32	1.86 5.58		
σ at local minima of $DPSNVR(\sigma)$	0.40		8.93	
σ_{weak} of $\frac{d(DPSNVR(z))}{dz}$		4.07		15.46



(a) (b) (c)

(d) (e) (f)

Figure 3.32 Edge images with σ obtained from Table 3.14: local minima of $SD(\sigma)$: (a) $\sigma = 0.32$. (b) $\sigma = 1.86$. (c) $\sigma = 5.58$; local minima of the DPSNVR graph and σ_{weak} of $\frac{d(DPSNVR(z))}{dz}$: (d) $\sigma = 0.40$. (e) $\sigma = 4.07$. (f) $\sigma = 8.93$.

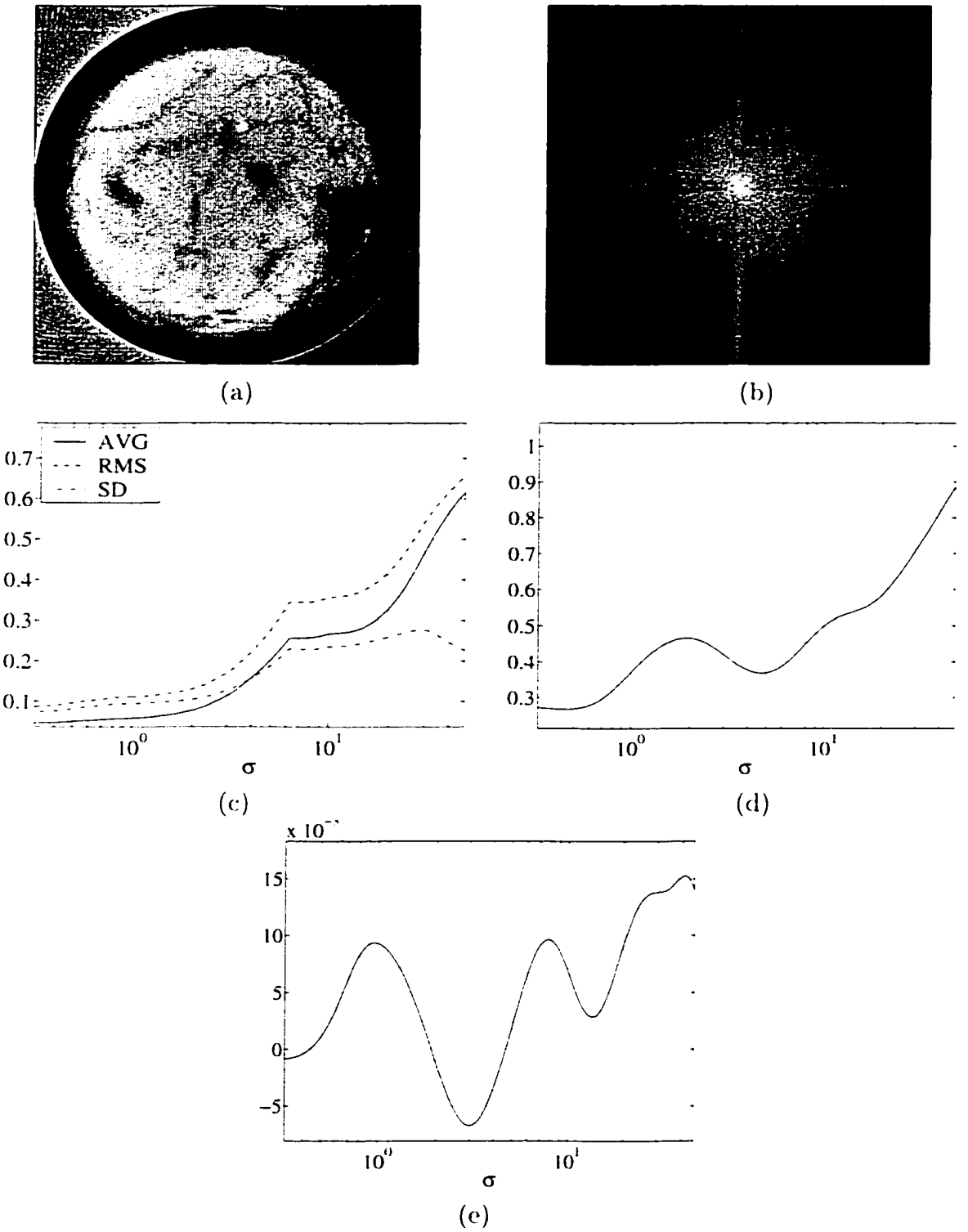
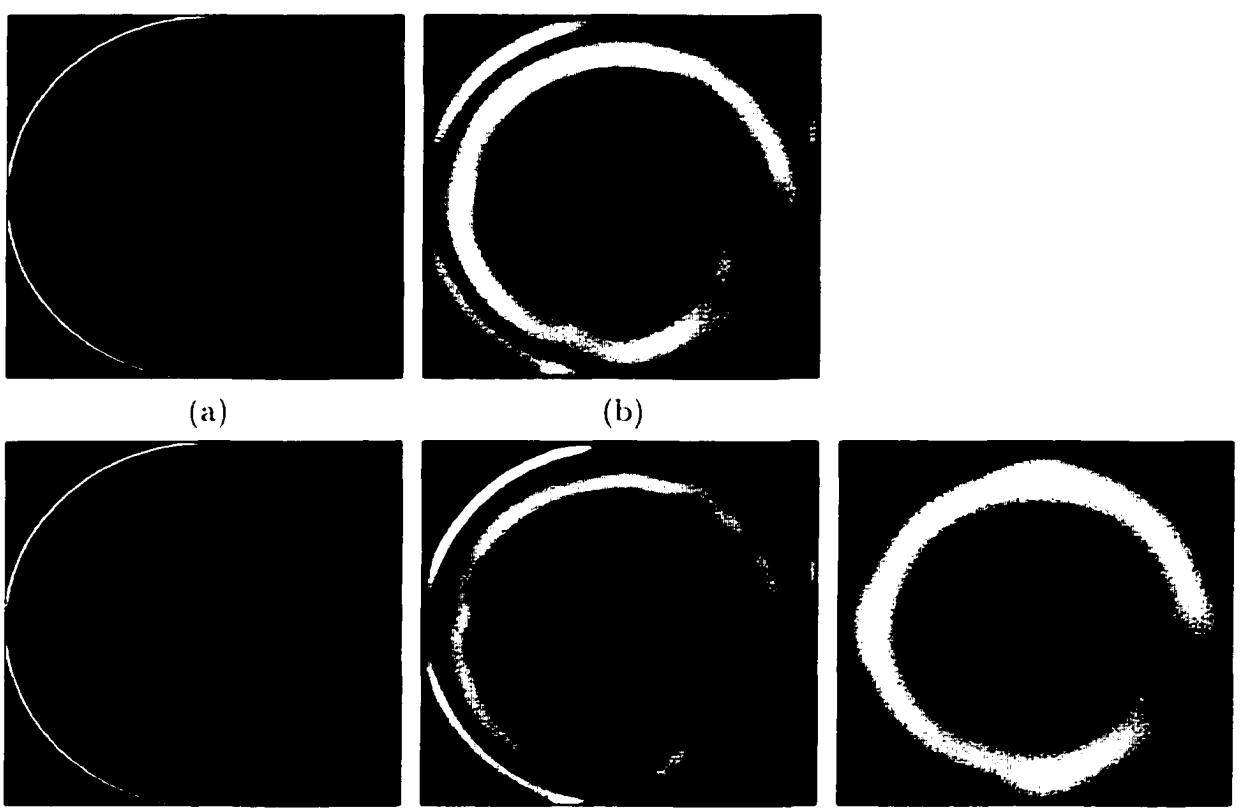


Figure 3.33 CT scan of a soil sample: (a) original image, (b) log magnitude of the Fourier transform, (c) ECS curves, (d) DPSNVR graph, (e) $\frac{d(DPSNVR(z))}{dz}$.

Table 3.16 Global scale analysis of the image in Figure 3.33.

	Scale 1	Scale 2	Scale 3
σ at local minima of $AVG(\sigma)$	0.32	6.78	
σ at local minima of $RMS(\sigma)$	0.32	7.34	
σ at local minima of $SD(\sigma)$	0.32	7.63	
σ at local minima of $DPSNVR(\sigma)$	0.44	4.77	
σ_{weak} of $\frac{d(DPSNVR(z))}{dz}$			13.74



(a) (b)

(c) (d) (e)

Figure 3.34 Edge images with σ obtained from Table 3.16: local minima of $SD(\sigma)$: (a) $\sigma = 0.32$. (b) $\sigma = 7.63$; local minima of the DPSNVR graph and σ_{weak} of $\frac{d(DPSNVR(z))}{dz}$: (c) $\sigma = 0.44$. (d) $\sigma = 4.77$. (e) $\sigma = 13.74$.

3.6 Conclusions

In this chapter, the problem of global scales is addressed. The global scale is defined as the resolution at which objects tend to exhibit some distinct observable characteristics. Based on the concept of scale-frequency representation, a frequency domain approach for global scale analysis is proposed. The method relies on the assumption that the spectrum of a multiscale signal contains several frequency lobes related to global scales. The differential power spectrum normalized variance ratio (DPSNVR) which is the ratio between the second order normalized central moments of the power spectra of the multiscale signal representation and the multiscale differential mask is used as a measure of global scales. The local minimum point in the DPSNVR graph is used as a global scale indication. Since the computation of moments is less sensitive to uniformly distributed noise, the method performs well even in a low signal to noise ratio situation. In the situation when the separation distance between objects is too small to create the distinct global scale, the weak scale can be detected by observing the derivative of the DPSNVR graph with respect to the natural scale parameter. In comparison, the edge characteristic-scale analysis performs poorly in several cases while the DPSNVR based technique performs extremely well in all cases.

CHAPTER 4. LOCAL SCALE ANALYSIS OF EDGES AND MULTISCALE DATA FUSION

In the previous chapter, a study of the global behavior of object edges established a framework for scale analysis. The formulation was carried out in the frequency domain. Global statistics of edges at different scales were used for scale detection while information at each detected scale was rendered separately. As the analysis focuses on smaller details, local information becomes more important but local behavior of each object may not be reflected in the global statistics. Local scale analysis presented in this chapter is specially designed for estimating suitable scale parameters to deal with local structures.

4.1 1-Dimensional Local Scale Analysis

Let $L^2(\mathbf{R})$ denote the space of square-integrable functions such that for $f \in L^2(\mathbf{R})$,

$$\int_{\mathbf{R}} f(t)^2 dt < \infty. \tag{4.1}$$

For all $f, g \in L^2(\mathbf{R})$, the inner product between f and g on $L^2(\mathbf{R})$ is defined as

$$\langle f, g \rangle = \int_{\mathbf{R}} f(t)g(t)dt, \tag{4.2}$$

while the L^2 norm $\|f\|$ is defined as

$$\begin{aligned} \|f\| &= \langle f, f \rangle^{\frac{1}{2}} && (4.3) \\ &= (\int_{\mathbf{R}} f(t)^2 dt)^{\frac{1}{2}}. && (4.4) \end{aligned}$$

From Cauchy-Schwarz inequality,

$$|\langle f, g \rangle| \leq \|f\|\|g\|. \tag{4.5}$$

it can be easily seen that

$$\max |\langle \frac{f}{\|f\|}, \frac{g}{\|g\|} \rangle| = 1 \tag{4.6}$$

which occurs when $f = c \cdot g$, where $c, c \neq 0$, is a constant coefficient. Let $\hat{o}(\sigma; t)$ be a multiscale normalized template described as

$$\hat{o}(\sigma; t) = \frac{1}{\|o(\frac{t}{\sigma})\|} o(\frac{t}{\sigma}).$$

where $o(t)$ is a symmetric mother smoothing function used to derive $\hat{o}$. From (4.6),

$$\max_{\sigma} |\langle f(t), \hat{o}(\sigma; t) \rangle| \tag{4.7}$$

occurs when $\hat{o}(\sigma; t)$ matches $f(t)$. Hence

$$\sigma_{max} = \arg\max_{\sigma} |\langle f(t), \hat{o}(\sigma; t) \rangle| \tag{4.8}$$

can be used to reveal scale information of f. Sizing by this method is analogous to measuring an unknown diameter of a hole by inserting gauges of known sizes into the hole and finding the one that fits.

An example in Figure 4.1 demonstrates the application to size 1-dimensional rectangular pulses based on Equation 4.8. The multiscale normalized Gaussian function used in this example is given by

$$\hat{g}(\sigma; t) = \frac{1}{\sqrt{\sigma\sqrt{\pi}}} e^{-\frac{t^2}{2\sigma^2}}. \tag{4.9}$$

where $\|\hat{g}\|$ is always 1. Figure 4.1-a shows the Gaussian functions $\hat{g}$ with different values of σ. The inner product between a rectangular pulse $p(W; t)$:

$$p(W; t) = \begin{cases} 1. & \text{for } |t| \leq W. \\ 0. & \text{otherwise.} \end{cases}$$

and $\hat{g}$ is given by

$$\begin{aligned} \langle p(W; t), \hat{g}(\sigma; t) \rangle &= \int_{-W}^{W} \frac{1}{\sqrt{\sigma\sqrt{\pi}}} e^{-\frac{t^2}{2\sigma^2}} dt \\ &= \sqrt{2\sigma\sqrt{\pi}} \operatorname{erf}(\frac{W}{\sqrt{2}\sigma}). \end{aligned}$$

where

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\tau^2} d\tau.$$

The graph $\langle p(10; t), \hat{g}(\sigma; t) \rangle$ as a function of σ with the peak at $\sigma_{max} = 7.17$ is shown in Figure 4.1-b. The relation between W and σ_{max} is found to be

$$\sigma_{max} = 0.717W$$

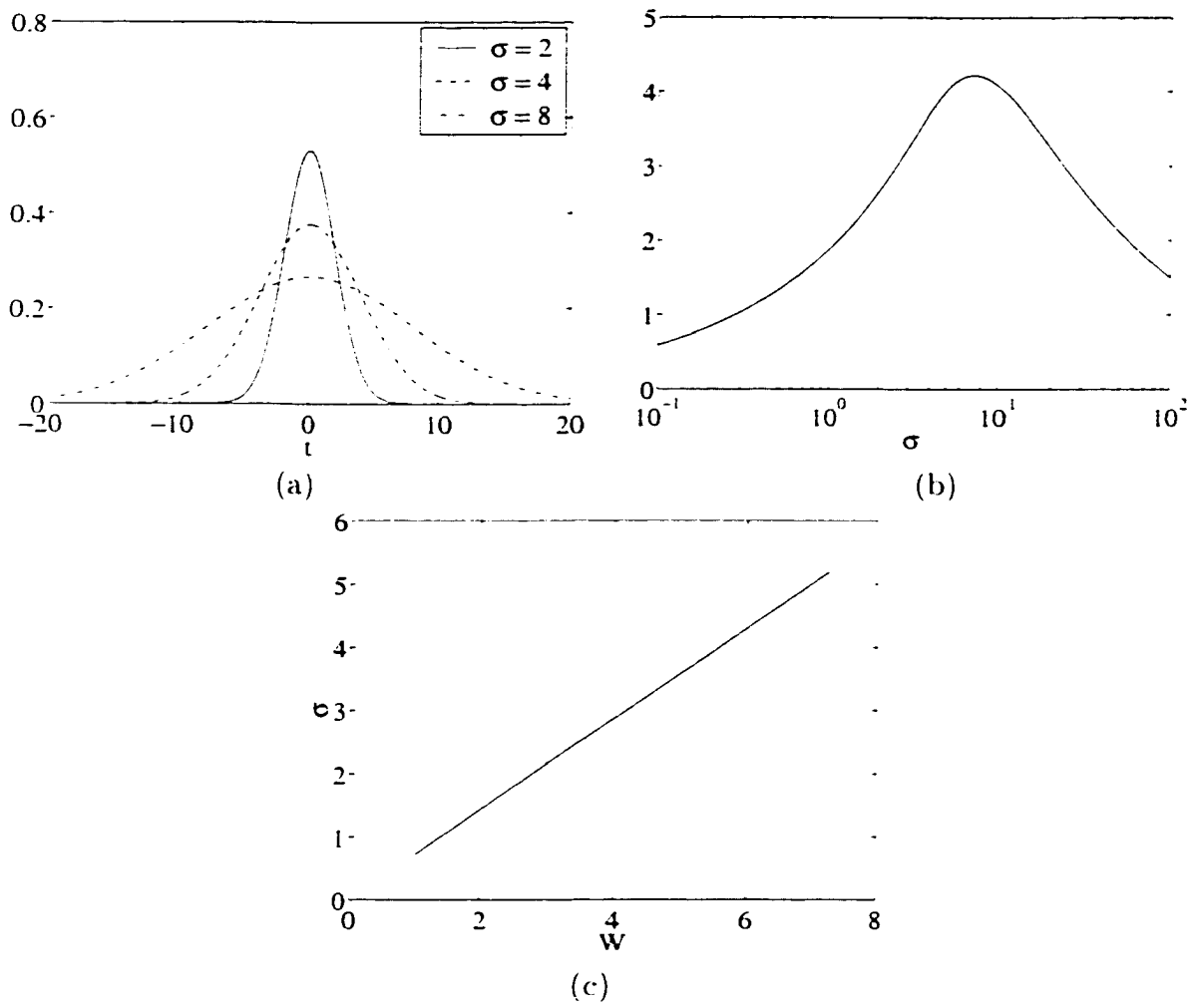


Figure 4.1 (a) Multiscale normalized Gaussian functions $\hat{g}(\sigma; t)$. (b) $\langle p(10; t), \hat{g}(\sigma; t)\rangle$ as a function of σ, and (c) graph between W and σ_{max}.

as illustrated by the graph in Figure 4.1-c.

Consider a scale-space image $f(\sigma, t)$ obtained from

$$f(\sigma, t) = f(t) * \hat{o}(\sigma; t) \tag{4.10}$$

$$= \int_{\mathbf{R}} f(\tau)\hat{o}(\sigma; t - \tau)d\tau. \tag{4.11}$$

Since $o(t) = o(-t)$, $f(\sigma, t)$ can be written as

$$f(\sigma, t) = \int_{\mathbf{R}} f(\tau)\hat{o}(\sigma; \tau - t)d\tau \tag{4.12}$$

$$= \langle f(\tau), \hat{o}(\sigma; \tau - t)\rangle. \tag{4.13}$$

According to Cauchy-Schwarz inequality, local maxima of $|f(\sigma, t)|$ occur when locations and sizes of $\hat{o}$ match those of local structures of f. Therefore, complete information of sizes, locations, and strengths of local structures of f can be obtained from local maxima of $|f(\sigma, t)|$. Equation 4.13 is called the *local scale analysis* of f with respect to $\hat{o}(\sigma; t)$. An example shown in Figure 4.2-a is a signal $x(t)$ with 2 rectangular pulses with $W = 10$ and 20 centered at $t = 110$ and 240 respectively. In Figure 4.2-b, $y(t)$ is a degraded version of $x(t)$ corrupted by additive white Gaussian noise $N(0, 0.25)$. The discrete convolution is used to compute the scale-space image $|y(\sigma, t)|$. Figures 4.2-c and 4.2-d show $|y(\sigma, t)|$ and its contour plot where there are 2 distinct peaks at $(t, \sigma) = (110, 7.244)$ and $(240, 13.66)$ corresponding to 2 rectangular pulses while spurious peaks due to noise can be seen in the lower part of the pictures.

4.1.1 Multiscale Differential Operators

Consider a multiscale first order derivative given by

$$p_t(\sigma, t) = p(t) * \hat{o}_t(\sigma; t) \tag{4.14}$$

$$= \int_{\mathbf{R}} p(\tau) \frac{d}{dt} \hat{o}(\sigma; t - \tau) d\tau \tag{4.15}$$

where $\hat{o}(\sigma; t)$ is a normalized multiscale smoothing function. Using associative property of convolution, Equation 4.14 can be rearranged as

$$p_t(\sigma, t) = \int_{\mathbf{R}} (\frac{d}{dt} p(t))_{t=\tau} \hat{o}(\sigma; t - \tau) d\tau \tag{4.16}$$

$$= \int_{\mathbf{R}} p_t(\tau) \hat{o}(\sigma; t - \tau) d\tau \tag{4.17}$$

$$= \int_{\mathbf{R}} p_t(\tau) \hat{o}(\sigma; \tau - t) d\tau \tag{4.18}$$

$$= \langle p_t(\tau), \hat{o}(\sigma; \tau - t) \rangle. \tag{4.19}$$

Consequently, the multiscale first order differential operation is equivalent to local scale analysis of $p_t(t)$ using the integration of the multiscale differential mask, $\int \hat{o}_t(\sigma; t) dt$, as a template. The local maxima of $|p_t(\sigma, t)|$ reveal scale information of differential structures of $p(t)$. For a given multiscale first order differential mask o_t, the normalization of o_t can be done as

$$\hat{o}_t = \frac{1}{\| \int_{\mathbf{R}} o_t(\sigma; t) dt \|} o_t(\sigma; t). \tag{4.20}$$

In a trivial case, a Gaussian diffused step function $p(t)$ is obtained by smoothing a unit step function $u(t)$ with the Gaussian function $g(\sigma_0; t)$,

$$p(t) = g(\sigma_0; t) * u(t)$$

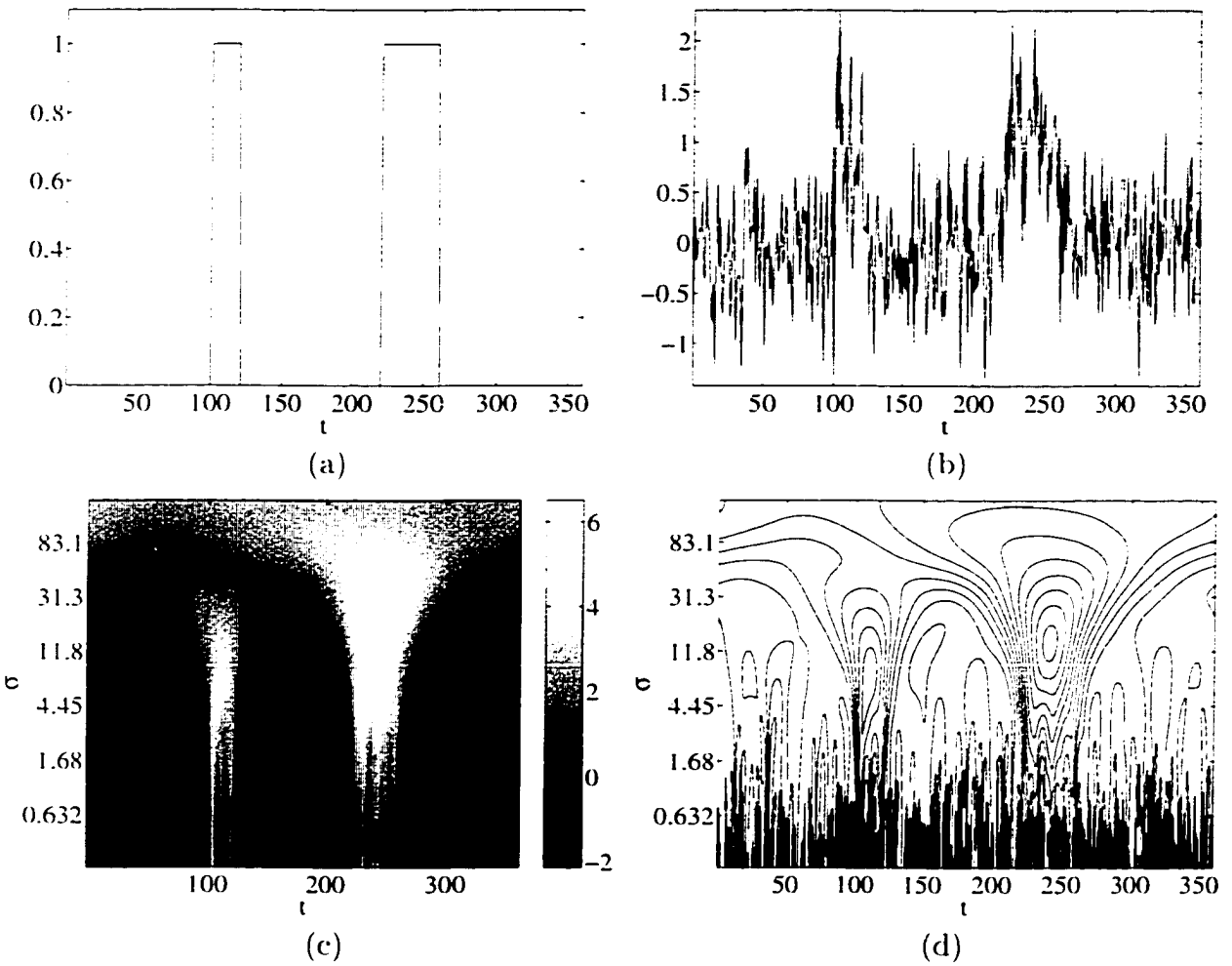


Figure 4.2 (a) Original signal $x(t)$. (b) degraded signal $y(t)$. (c) $y(\sigma, t)$. (d) contour plot of $y(\sigma, t)$.

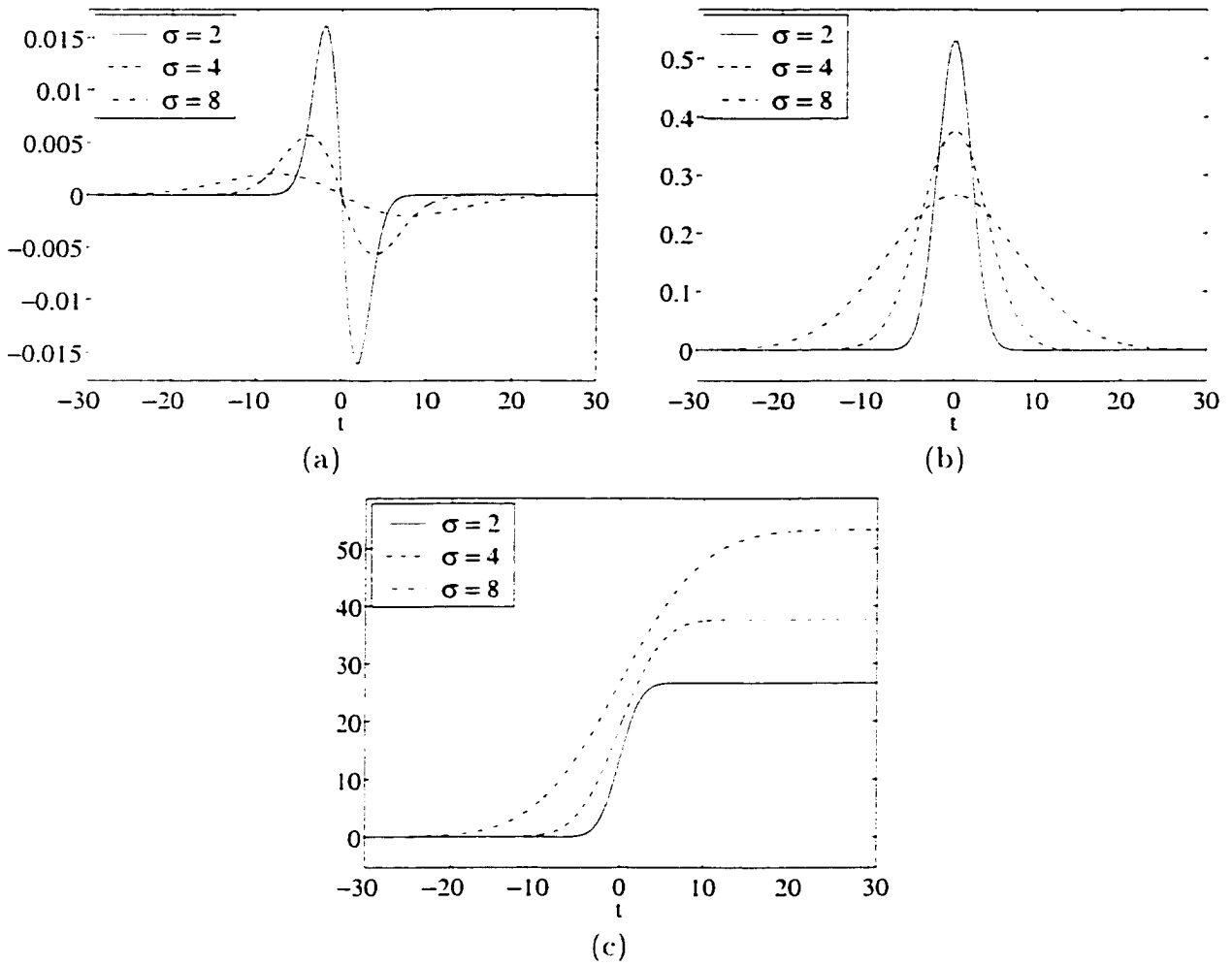


Figure 4.3 (a) The first order normalized Gaussian differential masks. (b) integrations of the masks. (c) the smoothed step functions.

$$= \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{t^2}{2\sigma_0^2}} dt.$$

Here, σ_0 represents a degree of smoothness of the signal. The first order normalized Gaussian differential function is given by

$$\hat{g}_t(\sigma; t) = \frac{d}{dt} \frac{1}{\sqrt{\sigma\sqrt{\pi}}} e^{-\frac{t^2}{2\sigma^2}} \tag{4.21}$$

$$= \frac{-t}{\sqrt{\sigma^5\sqrt{\pi}}} e^{-\frac{t^2}{2\sigma^2}}. \tag{4.22}$$

Figure 4.3 shows examples the first order normalized Gaussian differential masks. Equa-

tion 4.14 yields

$$\begin{aligned} p_t(\sigma, t) &= f(t) * \hat{g}_t(\sigma; t) \\ &= (\int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{t^2}{2\sigma_0^2}} dt) * (\frac{d}{dt} \frac{1}{\sqrt{\sigma\sqrt{\pi}}} e^{-\frac{t^2}{2\sigma^2}}) \\ &= (\frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{t^2}{2\sigma_0^2}}) * (\frac{1}{\sqrt{\sigma\sqrt{\pi}}} e^{-\frac{t^2}{2\sigma^2}}) \\ &= \frac{1}{\sqrt{2\pi^{3/2}\sigma\sigma_0}} \int_{-\infty}^{\infty} e^{-\frac{\tau^2}{2\sigma_0^2}} e^{-\frac{(t-\tau)^2}{2\sigma^2}} d\tau \\ &= \frac{1}{\pi^{1/4}} \sqrt{\frac{\sigma}{\sigma^2+\sigma_0^2}} e^{-\frac{t^2}{2(\sigma^2+\sigma_0^2)}} \end{aligned}$$

which has the peak at $t = 0$. By differentiating $p_t(\sigma, t = 0)$ with respect to σ and setting to zero,

$$\frac{\sigma_0^2 - \sigma^2}{2\pi^{1/4}\sigma^{1/2}(\sigma_0^2 + \sigma^2)^{3/2}} = 0.$$

we get $\sigma_{max} = \sigma_0$ that maximizes $p_t(\sigma, t)$. This example demonstrates the application of the multiscale Gaussian differential operation for measuring the degree of smoothness of a signal.

Another example of estimating smoothness of edges is shown in Figure 4.4. In general, a slope of an edge indicates the degree of smoothness of the edge. Figure 4.4-a is a signal f with ramp edges of slopes -0.0909 and 0.0244 centered at $t = 100$ and 300 respectively. Figure 4.4-b is a degraded version of f corrupted by additive white Gaussian noise $N(0, 0.01)$. The corresponding scale-space image and contour plot of $|p_t(\sigma; t)|$ computed using Equation 4.14 with $o_t = \hat{g}_t(\sigma; t)$ are shown in Figures 4.4-c and 4.4-d respectively. We see that there are 2 dominant peaks at $(t, \sigma) = (101, 4.801)$ and $(300, 14.97)$ corresponding to the two edges where the values of σ at the local minima are related to slopes of the ramp edges.

For higher order multiscale differential operators, normalization factors are needed to be chosen carefully depending upon the integration orders. In general, the n-order multiscale differential mask can be written as

$$\hat{o}_{t^n}(\sigma; t) = \frac{d^n}{dt^n} \hat{o}(\sigma; t). \tag{4.23}$$

The normalization factor is given by the *m^{th}-order integration norm*:

$$\begin{aligned} n(\sigma) &= \| \int \cdots \int \hat{o}_{t^{(n)}}(\sigma; t) dt^m \| & (4.24) \\ &= \| \frac{d^{n-m}}{dt^{n-m}} \hat{o}(\sigma; t) \|, \text{ where } 1 \leq m \leq n. & (4.25) \end{aligned}$$

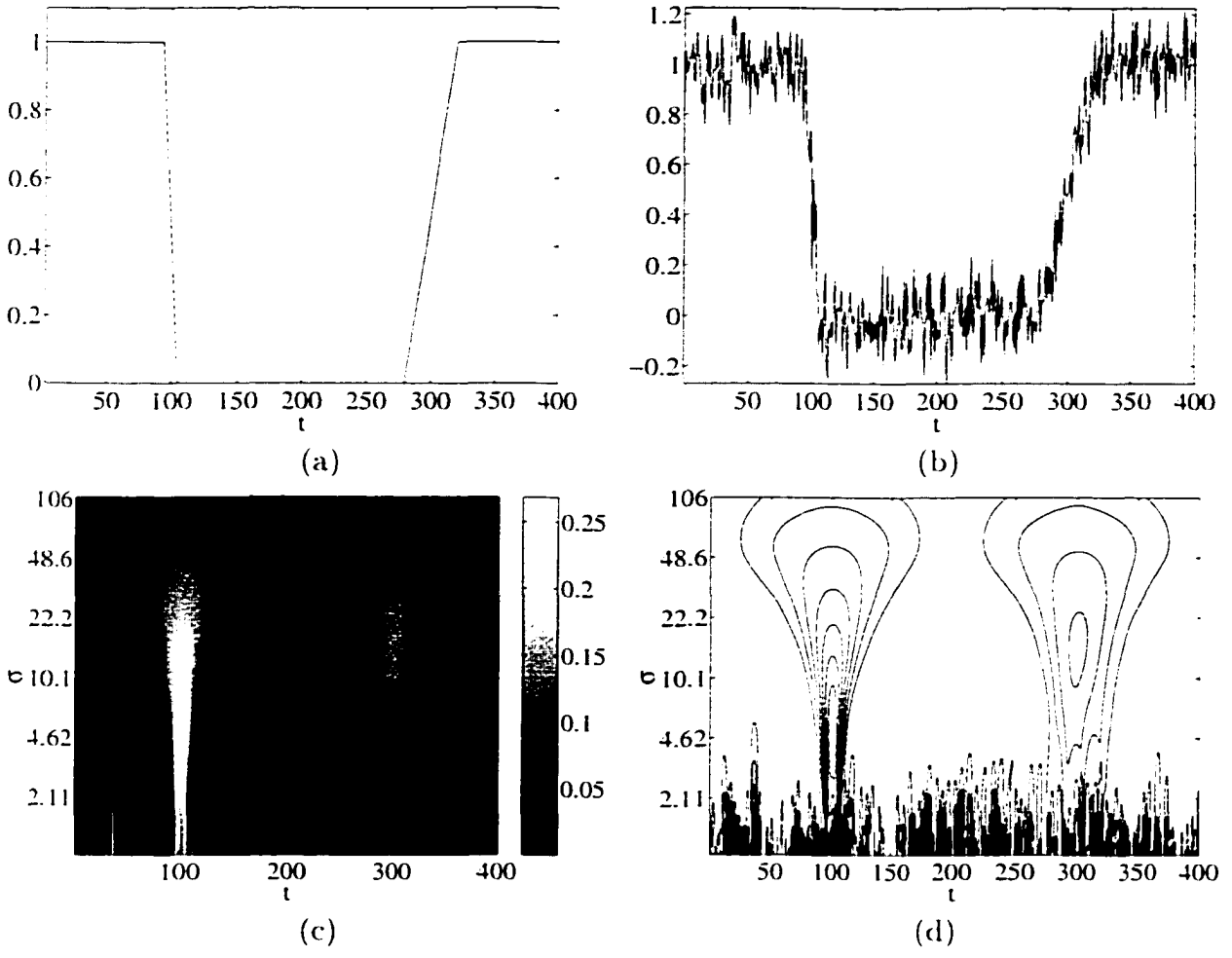


Figure 4.4 (a) Original signal. (b) degraded signal $p(t)$. (c) $|p_t(\sigma; t)|$. (d) contour plot of $|p_t(\sigma; t)|$.

The choice of the integration order m depends on the shape of structures to be matched but in general $m = 1$ works in most cases.

4.2 Extension to N-Dimensional Spaces

To extend the proposed local scale analysis scheme to N-dimensional signals, several issues must be taken into account. The origin and a set of N linearly independent orthonormal basis vectors are adequate to express an N-dimensional coordinate system. In general, the geometrical information representation must be independent of the coordinate system. For example, differential structures [26, 79] are invariant under Cartesian coordinate transformations: rotation and translation. In this section, local scale analysis

of differential structures is investigated.

4.2.1 Differential Structures

In order to express differential invariants of orthogonal transformations in simple forms, it is natural to introduce the local orthonormal coordinate system in which directions of axes depend on local information. Based on the gradient direction, the gauge coordinate (v, w) [26] is introduced such that v-axis and w-axis are, respectively, orthogonal and parallel to the local gradient vector.

$$\begin{pmatrix} v \\ w \end{pmatrix} = \frac{1}{\sqrt{L_x^2 + L_y^2}} \begin{pmatrix} L_y & -L_x \\ L_x & L_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{4.26}$$

$$= \begin{pmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{4.27}$$

$$\tag{4.28}$$

where L_x and L_y represent image derivatives in x and y directions and $\theta = \tan^{-1}\frac{L_y}{L_x}$. Partial derivative operators in v and w directions can be computed by

$$\begin{pmatrix} \partial_v \\ \partial_w \end{pmatrix} = \begin{pmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix}. \tag{4.29}$$

Since the gradient direction is invariant with respect to choice of coordinate system, all polynomial expressions in (v, w) are invariant under orthogonal transformations. In general, the gauge coordinate system in Equation 4.28 is valid only in the areas of none zero gradient but such a condition is adequate for numerical analysis of images. Table 4.1 shows examples of differential geometries which are invariant under Cartesian coordinate transformations of a 2-dimensional image expressed in Cartesian and gauge coordinates along with the corresponding geometrical meanings. In scale-space analysis, differential geometries are related to differential structures.

4.2.2 Scale Invariant Multiscale Differential Operators

The fundamental components of differential structures are partial derivatives of the image. Based on the concept of scale-space analysis, multiscale partial derivative operators must be invariant to scales. In the 1-dimensional case, the integration norm was used as the normalization factor for the multiscale differential mask. However, in a multidimensional space, the choice of suitable integration direction is difficult. Normalization of the mask by the integration norm must be selected according to the integration

Table 4.1 Examples of 2-dimensional differential structures invariant under Cartesian coordinate transformations.

Cartesian	Gauge	Geometrical meaning
L	L	Intensity image
$L_x^2 + L_y^2$	L_w^2	$\|\nabla L\|^2$
$L_{xx} + L_{yy}$	$L_{vv} + L_{ww}$	Laplacian $\nabla^2 L$
$\frac{-L_x^2 L_{yy} + 2L_x L_y L_{xy} - L_y^2 L_{xx}}{(L_x^2 + L_y^2)^{3/2}}$	$-\frac{L_{vv}}{L_w}$	Isophote curvature
$\frac{L_x L_y (L_{yy} - L_{xx}) + L_{xy}(L_x^2 + L_y^2)}{(L_x^2 + L_y^2)^{3/2}}$	$-\frac{L_{vw}}{L_w}$	Flowline curvature
$\frac{L_x^2 L_{xx} + 2L_x L_y L_{xy} + L_y^2 L_{yy}}{L_x^2 + L_y^2}$	L_{ww}	$\nabla(\|\nabla L\|) \cdot \frac{\nabla L}{\|\nabla L\|}$

path. In this research, we propose the scale invariant normalization scheme for the 2-dimensional multiscale differential masks using the directional integration norm of the masks as follows:

For a function $f(x, y)$, the (p, q)-order integration norm in x-direction is defined as

$$\mathcal{N}x_{(p,q)}(f) = \int_{\mathbf{R}} \left(\int_{\mathbf{R}} \left(\int_{\mathbf{R}^{(p+q)}} f(x, y) dx^p dy^q \right)^2 dx \right)^{\frac{1}{2}} dy. \tag{4.30}$$

Similarly, the (p, q)-order integration norm in y-direction is defined as

$$\mathcal{N}y_{(p,q)}(f) = \int_{\mathbf{R}} \left(\int_{\mathbf{R}} \left(\int_{\mathbf{R}^{(p+q)}} f(x, y) dx^p dy^q \right)^2 dy \right)^{\frac{1}{2}} dx. \tag{4.31}$$

The normalization factors for $o_{x^n y^m}(\sigma; x, y)$ are selected from the directional integration norms $\mathcal{N}x_{(p,q)}(o_{x^n y^m})$ and $\mathcal{N}y_{(p,q)}(o_{x^n y^m})$ depending on the order and direction of integration. In this research, the normalization factors are limited to the first order directional integration norms:

$$\mathcal{N}x_{(1,0)}(o_{x^n y^m}(\sigma; x, y)) = \int_{\mathbf{R}} \left(\int_{\mathbf{R}} \left(\int_{\mathbf{R}} o_{x^n y^m}(\sigma; x, y) dx \right)^2 dx \right)^{\frac{1}{2}} dy, \ n \geq 1. \tag{4.32}$$

and

$$\mathcal{N}y_{(0,1)}(o_{x^n y^m}(\sigma; x, y)) = \int_{\mathbf{R}} \left(\int_{\mathbf{R}} \left(\int_{\mathbf{R}} o_{x^n y^m}(\sigma; x, y) dy \right)^2 dy \right)^{\frac{1}{2}} dx, \ m \geq 1. \tag{4.33}$$

In the discrete case, $Nx_{(1,0)}$ and $Ny_{(1,0)}$ are given by

$$Nx_{(1,0)}(o_{x^n y^m}(\sigma; x, y)) = \frac{1}{M}\sum_y (\frac{1}{N}\sum_x |\frac{1}{N}\sum_x o_{x^n y^m}(\sigma; x, y)|^2)^{\frac{1}{2}} \tag{4.34}$$

and

$$Ny_{(0,1)}(o_{x^n y^m}(\sigma; x, y)) = \frac{1}{N}\sum_x (\frac{1}{M}\sum_y |\frac{1}{M}\sum_y o_{x^n y^m}(\sigma; x, y)|^2)^{\frac{1}{2}}. \tag{4.35}$$

where $N \times M$ is the mask size. The choice of directional normalization factors depends on the order of partial derivations that generate the mask $o_{x^n y^m}(\sigma; x, y)$. For instance, the normalization factor of $o_{xxx}(\sigma; x, y)$ is $Nx_{(1,0)}(o_{xxx})$ while the normalization factor of $o_{yyy}(\sigma; x, y)$ is $Ny_{(0,1)}(o_{yyy})$. On the other hand, for $o_{xxy}(\sigma; x, y)$, there are 3 possible partial derivation orders

$$o_{xxy}(\sigma; x, y) = \frac{\partial}{\partial x} o_{xy}(\sigma; x, y) \text{ or } \frac{\partial}{\partial x} o_{yx}(\sigma; x, y) \text{ or } \frac{\partial}{\partial y} o_{xx}(\sigma; x, y).$$

Hence, $Nx_{(1,0)}(o_{xxy})$ is used for the first two cases while $Ny_{(0,1)}(o_{xxy})$ is used for the last case.

In general, the number of partial derivative orders to generate $o_{x^n y^m}$ from $o_{x^{n-1} y^m}$ is $\frac{(n+m-1)!}{(n-1)!m!}$ while the number of partial derivative orders to generate $o_{x^n y^m}$ from $o_{x^n y^{m-1}}$ is $\frac{(n+m-1)!}{n!(m-1)!}$. The total number of partial derivative orders to generate $o_{x^n y^m}$ from $o_{x^{n-1} y^m}$ and $o_{x^n y^{m-1}}$ is $\frac{(n+m)!}{n!m!}$. As a result, the average normalization factor n used in the computation of differential structures for $o_{x^n y^m}$ is given by

$$\frac{1}{n} = \frac{n!m!}{(n+m)!}\left(\frac{(n+m-1)!}{(n-1)!m!}\frac{1}{Nx_{(1,0)}(o_{x^n y^m})} + \frac{(n+m-1)!}{n!(m-1)!}\frac{1}{Ny_{(0,1)}(o_{x^n y^m})}\right). \tag{4.36}$$

For example, the 2-dimensional multiscale Gaussian function and its first order partial derivatives are given by

$$o(\sigma; x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{4.37}$$

$$o_x(\sigma; x, y) = -\frac{x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.38}$$

and

$$o_y(\sigma; x, y) = -\frac{y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{4.39}$$

The normalization factor of $o_y(\sigma; x, y)$ and $o_x(\sigma; x, y)$, are given by

$$Ny_{(0,1)}(o_y(\sigma; x, y)) = Nx_{(1,0)}(o_x(\sigma; x, y)) \tag{4.40}$$

$$= \int_{\mathbf{R}} (\int_{\mathbf{R}} |\int_{\mathbf{R}} o_x(\sigma; x, y) dx|^2 dx)^{\frac{1}{2}} dy \tag{4.41}$$

$$= \int_{\mathbf{R}} (\int_{\mathbf{R}} \phi(\sigma; x, y)^2 dx)^{\frac{1}{2}} dy \tag{4.42}$$

$$= \int_{\mathbf{R}} (\int_{\mathbf{R}} \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{\sigma^2}} dx)^{\frac{1}{2}} dy \tag{4.43}$$

$$= \frac{1}{\sqrt{2\sigma\sqrt{\pi}}}. \tag{4.44}$$

4.2.3 Local Scale Analysis of Edges in 2-Dimensional Images

Image gradient is a natural descriptor of edges. The strength of edges can be determined from the gradient magnitude while edge locations can be obtained from the maximum gradient paths. Local scale analysis of edges determines degrees of smoothness of edges in terms of scale parameters. Lindenberg [52] introduced the differential geometric definition of edges based on image gradient:

$$L_{ww} = \frac{L_x^2 L_{xx} + 2L_x L_y Lxy + L_y^2 L_{yy}}{L_x^2 + L_y^2} = 0 \tag{4.45}$$

$$L_{www} = \frac{L_x^3 L_{xxx} + 3L_x^2 L_y L_{xxy} + 3L_x L_y^2 L_{xyy} + L_y^3 L_{yyy}}{(L_x^2 + L_y^2)^{\frac{3}{2}}} < 0. \tag{4.46}$$

This definition is equivalent to the maximum gradient path. To avoid the direct computation of high order partial derivatives, definition of edges used in this research is given by

$$L_{ww} = \nabla(|\nabla L|) \cdot \frac{\nabla L}{|\nabla L|} = 0 \tag{4.47}$$

and

$$\nabla L_{ww} \cdot \nabla L < 0. \tag{4.48}$$

In practice, edge localization is done by detecting zero crossing paths where L_{ww} changes its sign and Equation 4.48 is satisfied.

For local scale analysis, it is assumed that edges in different parts of an image have different degrees of smoothness corresponding to different σ. To analyze scales of local edges, normalized multiscale Gaussian differential operators, $\hat{o}_x(\sigma; x, y)$ and $\hat{o}_y(\sigma; x, y)$ with normalization factors in Equation 4.44 are employed. For an image L, local scale analysis of edges in L is obtained by collecting the maximum edge map e_{max} along with $\sigma_{max}(x, y)$:

$$e_{max}(x, y) = \max_{\sigma} |\nabla L(\sigma; x, y)| \tag{4.49}$$

$$= \max_{\sigma} \sqrt{L_x^2(\sigma; x, y) + L_y^2(\sigma; x, y)} \tag{4.50}$$

$$= \max_{\sigma} \sqrt{(L * \hat{o}_x(\sigma; x, y))^2 + (L * \hat{o}_y(\sigma; x, y))^2} \tag{4.51}$$

and

$$\sigma_{max}(x, y) = \arg\max_{\sigma} |\nabla L(\sigma)|. \tag{4.52}$$

Using the definition of edges in 4.47 and 4.48 to localize edges, location information of edges is obtained.

Examples of local scale analysis of edges are given below. The first example in Figure 4.5-a shows a simulated image containing a square smoothed by Gaussian masks of different scales. The lower left, lower right, upper right, and upper left quarters of the square are smoothed by the multiscale 2-dimensional Gaussian smoothing functions with $\sigma_0 = 1, 2, 4,$ and 8 respectively. Two local scale analysis methods with and without edge localization are tested. The multiscale Gaussian differential masks with the normalization factors computed using the discrete formulae in Equations 4.34 and 4.35 are used. In the method with edge localization, only edge pixels satisfying conditions in 4.47 and 4.48 are taken into account. The σ-map and e_{max} obtained from Equations 4.52 and 4.51 without edge localization are displayed in Figures 4.5-b and 4.5-c while those with edge localization are displayed in Figures 4.5-d and 4.5-e respectively. Comparing the results with and without edge localization, results without edge localization yields false indication of σ in the areas adjacent to the structures. This attributes to the diffusion of the gradient of the strong structures in the neighborhood. In contrast, in Figures 4.5-d and 4.5-e, only edge pixels are preserved. Figure 4.5-f shows the values of σ_{max} detected along the perimeter of the square which closely match the actual values of σ_0 of the Gaussian smoothing functions used to smooth the square. The next example in Figure 4.6 shows the application of local scale analysis for blur estimation. Figure 4.6-a contain in-focus and out-of-focus objects. The σ-maps in Figures 4.6-c and 4.6-e indicate the small values of σ at edges of the in-focus object and the large values of σ at edges of the out-of-focus object. The results using edge localization yields a sharper e_{max} image shown in Figure 4.6-d compared to the e_{max} image in Figure 4.6-b obtained without edge localization.

Besides sizing the local structures, this procedure can be used for multiscale data fusion where only values associated with local maxima of the scale-space differential structure image are collected. In the next section, several multiscale data fusion methods are investigated.

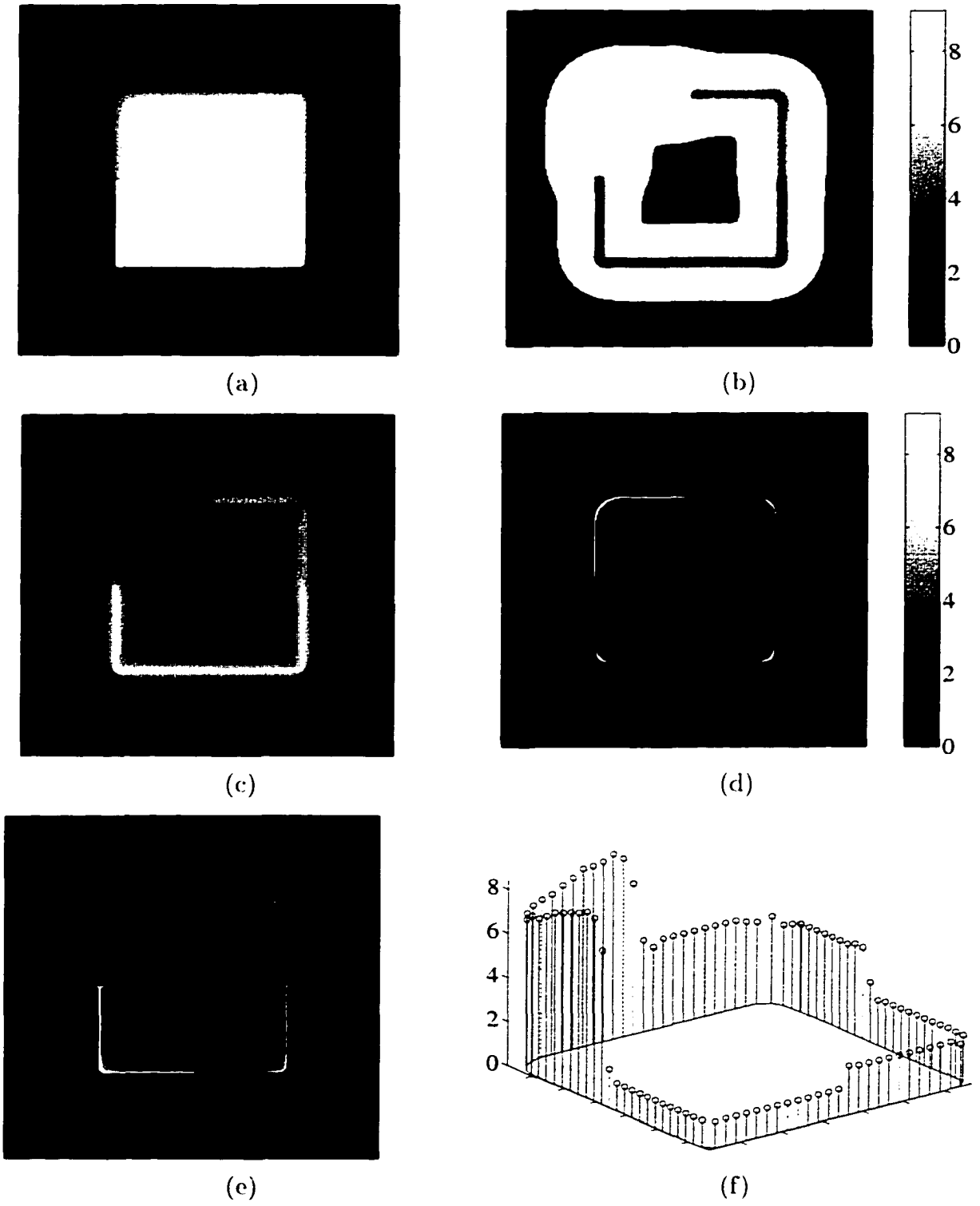


Figure 4.5 (a) Original image of a smoothed square. (b) σ-map and (c) e_{max} without edge localization. (d) σ-map and (e) e_{max} with edge localization. (f) σ_{max} along the boundary of the square.

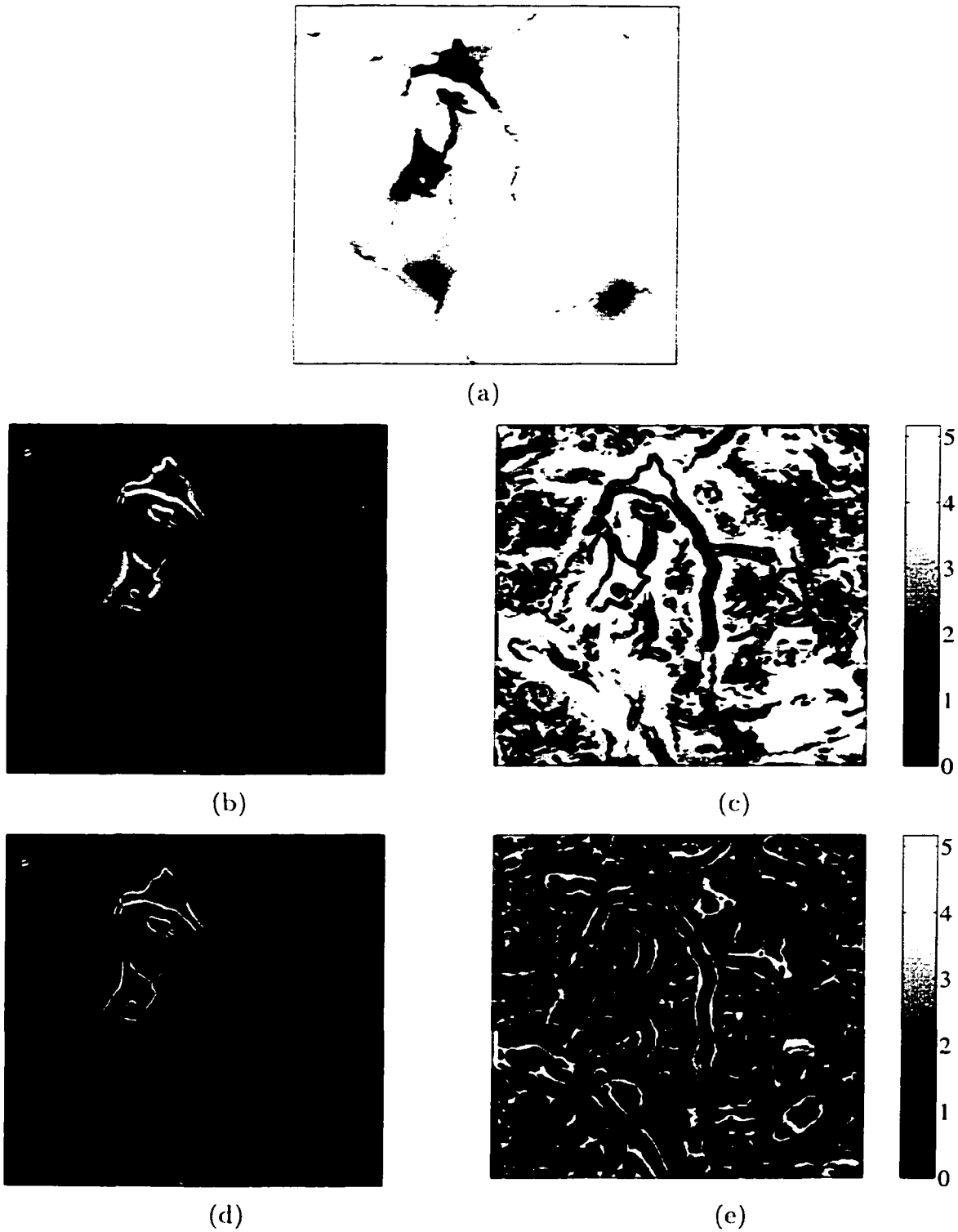


Figure 4.6 (a) Original image (courtesy of John Kesterson, VayTek, Inc.), (b) e_{max} and (c) σ-map without edge localization. (d) e_{max} and (e) σ-map with edge localization.

4.3 Multiscale Data Fusion and Multiple Scale Differential Masks

Multiscale data fusion is the method for utilizing data obtained at different scales. The techniques proposed so far were based on the behavior of image structures such as local extrema of scale-space gradient magnitude images or zero crossings of scale-space Laplacian images [5, 28, 29, 32, 34, 47, 56, 67]. Exhaustive searches as well as tracking schemes that links structures at different scales together using data structures such as trees and techniques for handling structures of different topologies at different scales have been employed. One approach investigated in this research is summation of multiscale data. Based on the scale invariant normalization scheme proposed previously, the details at all scales are treated equally. Summation of data at all scales can therefore be done directly without any modification.

Let $\Psi(\sigma; x, y)$ be a scale-space differential structure image. Multiscale data fusion images are obtained as

$$\sum_{\sigma} \Psi(\sigma; x, y) \tag{4.53}$$

and

$$\int_{Z} \Psi(\sigma = e^{\alpha z}; x, y) dz, \ \alpha > 0, \tag{4.54}$$

for discrete and continuous σ respectively. To obtain a multiscale data fusion image of nonlinear differential structures, computation of Ψ at each scale must be performed individually. Hence the overall computational cost of a multiscale data fusion image in 4.53 is as same as that required for local scale analysis. On the other hand, for linear differential structures, summation of multiscale differential structure images is equivalent to a single convolution between the image and summation of the corresponding multiscale kernels $\hat{\psi}(\sigma; x, y)$:

$$\begin{aligned} \sum_{\sigma} \Psi(\sigma; x, y) &= \sum_{\sigma} L(x, y) * \hat{\psi}(\sigma; x, y) && (4.55) \\ &= L(x, y) * \sum_{\sigma} \hat{\psi}(\sigma; x, y). && (4.56) \end{aligned}$$

As a result, the computational cost is significantly reduced. The mask $\sum_{\sigma} \hat{\psi}(\sigma; x, y)$ is called the *multiple scale mask*. Based on this concept, the multiple scale differential mask is given by

$$\sum_{i=0}^{N_\sigma} \hat{\phi}_{x^n y^m}(\sigma_i; x, y) = \sum_{i=0}^{N_\sigma} \frac{\phi_{x^n y^m}(\sigma_i; x, y)}{N(\phi_{x^n y^m}(\sigma_i; x, y))}. \tag{4.57}$$

where $N(o_{x^n y^m}(\sigma_i; x, y))$ is the normalization factor of $o_{x^n y^m}(\sigma_i; x, y)$ computed using Equation 4.36. N_σ is the total number of scales, and

$$\sigma_i = \sigma_0 \cdot \alpha^i, \ \alpha > 1 \text{ and } \sigma_0 > 0.$$

Figures 4.7 to 4.9 illustrate examples of multiple scale differential masks. Since the multiple scale differential masks are the combinations of the normalized multiscale differential masks at different scales, differential structures computed contain contribution from different scales.

4.3.1 Experimental Results of Multiscale Data Fusion

Let Ψ be a multiscale differential structure image. 4 multiscale data fusion algorithms tested in this section are listed below.

1. Maximization of $|\Psi|$ with respect to σ without feature localization:

   $$\Psi(\sigma_{max}; x, y)$$

   where

   $$\sigma_{max} = \arg\max_{\sigma} |\Psi(\sigma; x, y)|.$$

2. Maximization of $|\Psi|$ with respect to σ with feature localization:

   $$\Psi(\sigma_{max}; x_i, y_i)$$

   where

   $$\sigma_{max} = \arg\max_{\sigma} |\Psi(\sigma; x_i, y_i)|$$

   and $\{(x_i, y_i)\}$ represent locations of structures.

3. Summation of Ψ with respect to σ:

   $$\sum_{\sigma} \Psi(\sigma; x, y)$$

4. Computing Ψ using the multiple scale differential masks:

   $$\Psi(x, y) = f(L, L_x, L_y, L_{xx}, L_{xy}, \ldots, L_{x^n y^m}),$$

   where $L_x, L_y, L_{xx}, L_{xy}, \ldots, L_{x^n y^m}$ are computed using multiple scale differential masks.

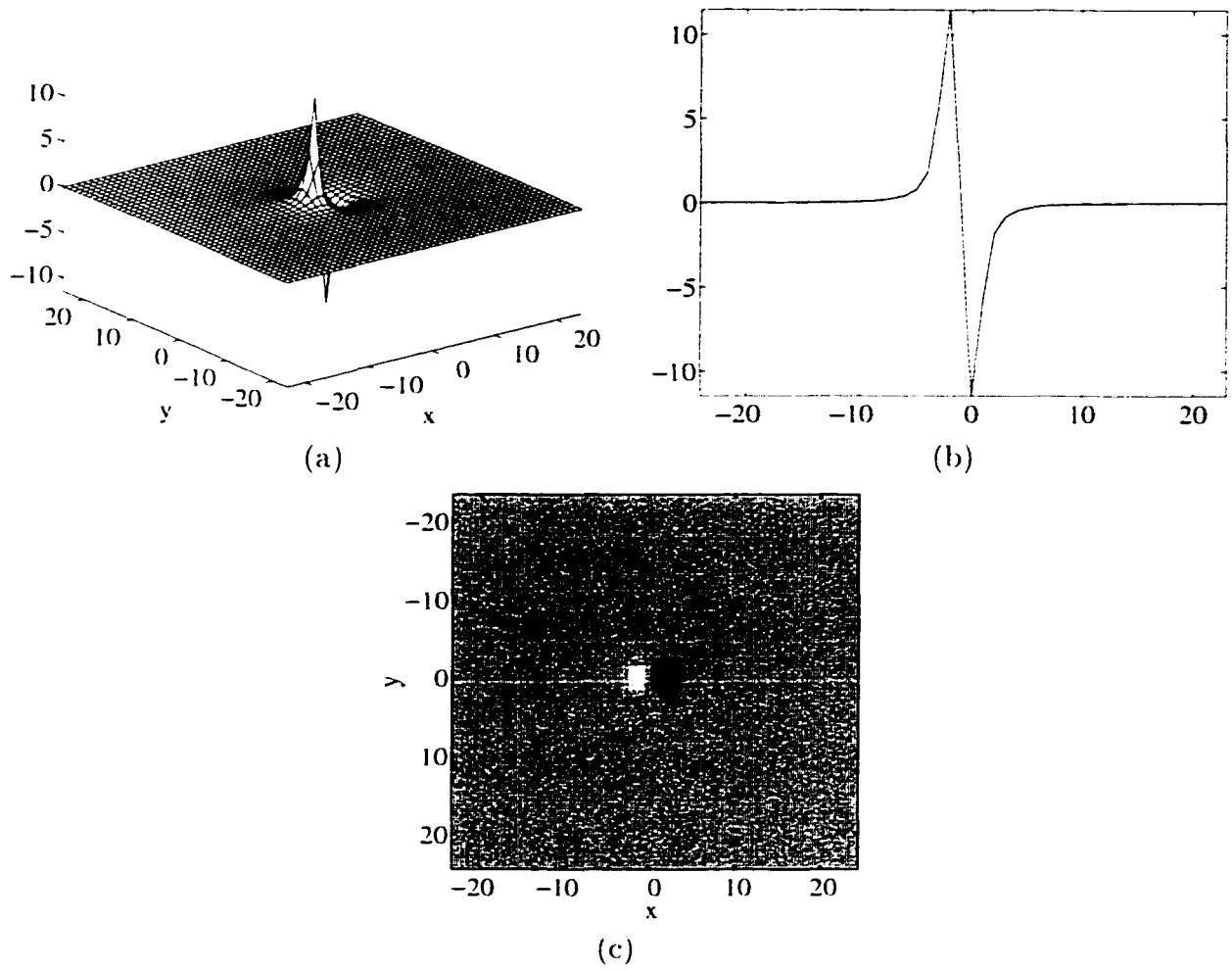


Figure 4.7 The multiple scale differential mask $\sum_{i=0}^{N_\sigma} \hat{o}_x(\sigma_i; x, y)$ with σ varying from 0.5 to 11.88: (a) surface plot, (b) cross section at $y = 0$, (c) intensity image.

A comparison between the different multiscale data fusion algorithms is demonstrated by examples given next. In all the examples, normalized multiscale Gaussian differential masks were used to generate multiscale partial derivatives of the images as well as multiscale differential structure images. Results of multiscale data fusion of gradient magnitude images and Laplacian images are provided.

4.3.1.1 Multiscale data fusion of gradient magnitude images

Figure 4.10 illustrates the gradient magnitude images $|\nabla L(\sigma; x, y)|$ at several scales. The original image shown in Figure 4.10-a contains sharp edges to the right of the

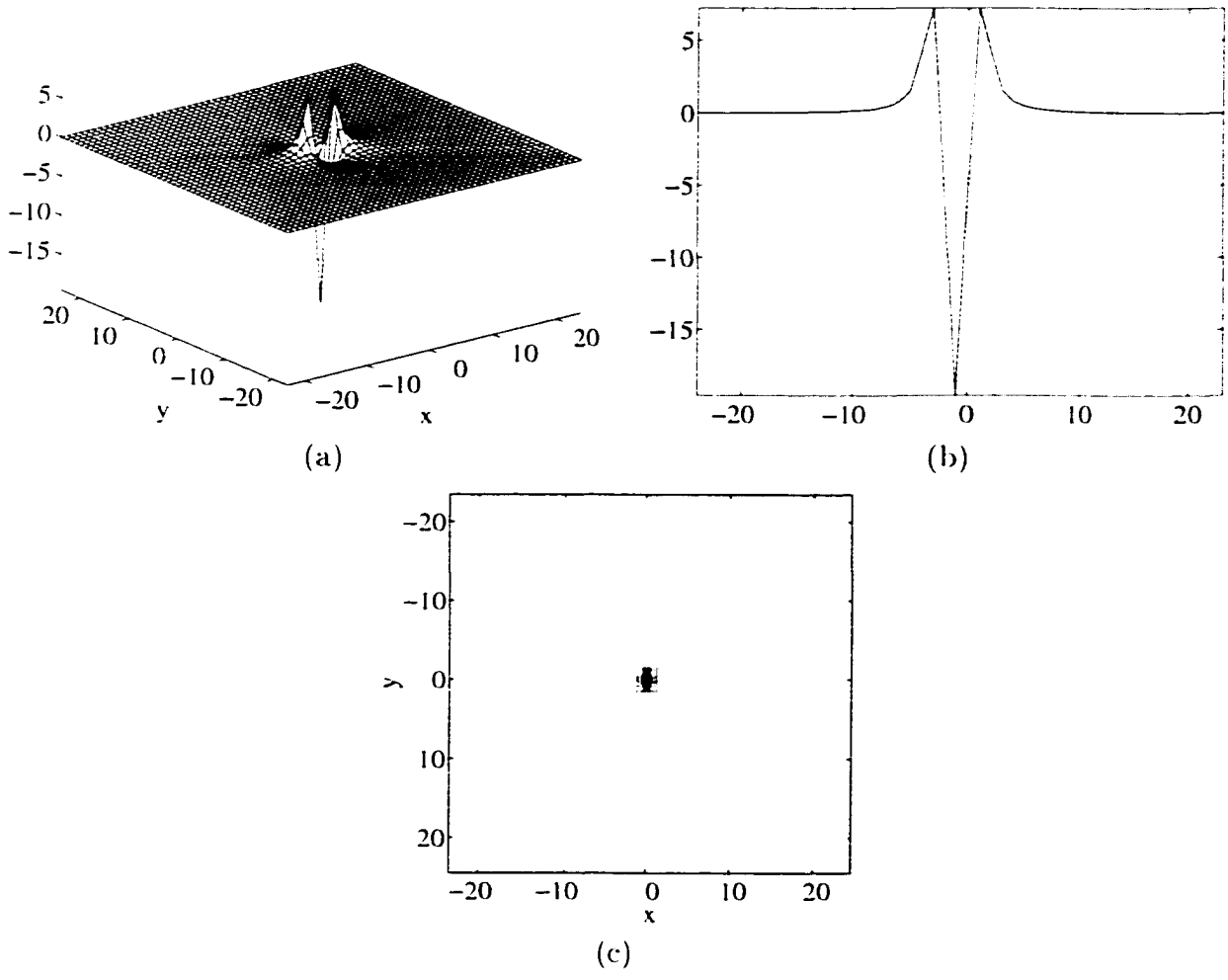


Figure 4.8 The multiple scale differential mask $\sum_{i=0}^{N_\sigma} \hat{o}_{xx}(\sigma_i; x, y)$ with σ varying from 0.5 to 11.88: (a) surface plot, (b) cross section at $y = 0$, (c) intensity image.

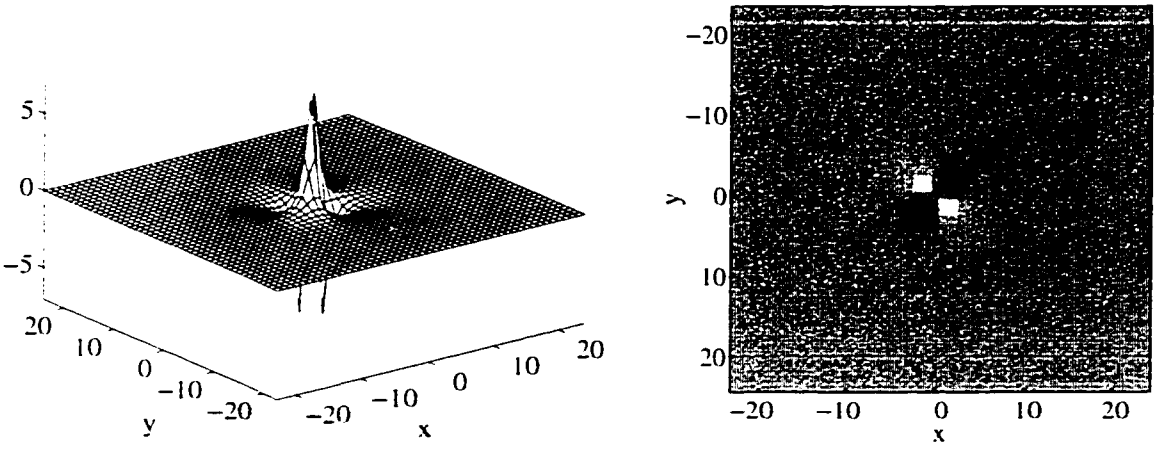


Figure 4.9 The multiple scale differential mask $\sum_{i=0}^{N_\sigma} \hat{o}_{xy}(\sigma_i; x, y)$ with σ varying from 0.5 to 11.88: (a) surface plot. (b) intensity image.

face and blurred edges at the shadow to the left of the face. The images $|\nabla L(\sigma; x, y)|$ shown in Figures 4.10-b to 4.10-f are computed using the normalized multiscale Gaussian differential masks.

$$\hat{o}_x(\sigma; x, y) = -\frac{x}{\sqrt{2\sqrt{\pi}^3 \sigma^7}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

and

$$\hat{o}_y(\sigma; x, y) = -\frac{y}{\sqrt{2\sqrt{\pi}^3 \sigma^7}} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

The 4 multiscale data fusion algorithms for $|\nabla L(\sigma; x, y)|$ are obtained as follows.

1. Maximization of $|\nabla L|$ with respect to σ without edge localization:

$$\max_{\sigma} |\nabla L(\sigma; x, y)|.$$

2. Maximization of $|\nabla L|$ with respect to σ with edge localization:

$$\max_{\sigma} |\nabla L(\sigma; x_i, y_i)|,$$

where (x_i, y_i) satisfies

$$\begin{aligned} \frac{L_x^2 L_{xx} + 2L_x L_y L_{xy} + L_y^2 L_{yy}}{L_x^2 + L_y^2} &= 0 \\ \frac{L_x^3 L_{xxx} + 3L_x^2 L_y L_{xxy} + 3L_x L_y^2 L_{xyy} + L_y^3 L_{yyy}}{(L_x^2 + L_y^2)^{\frac{3}{2}}} &< 0. \end{aligned}$$

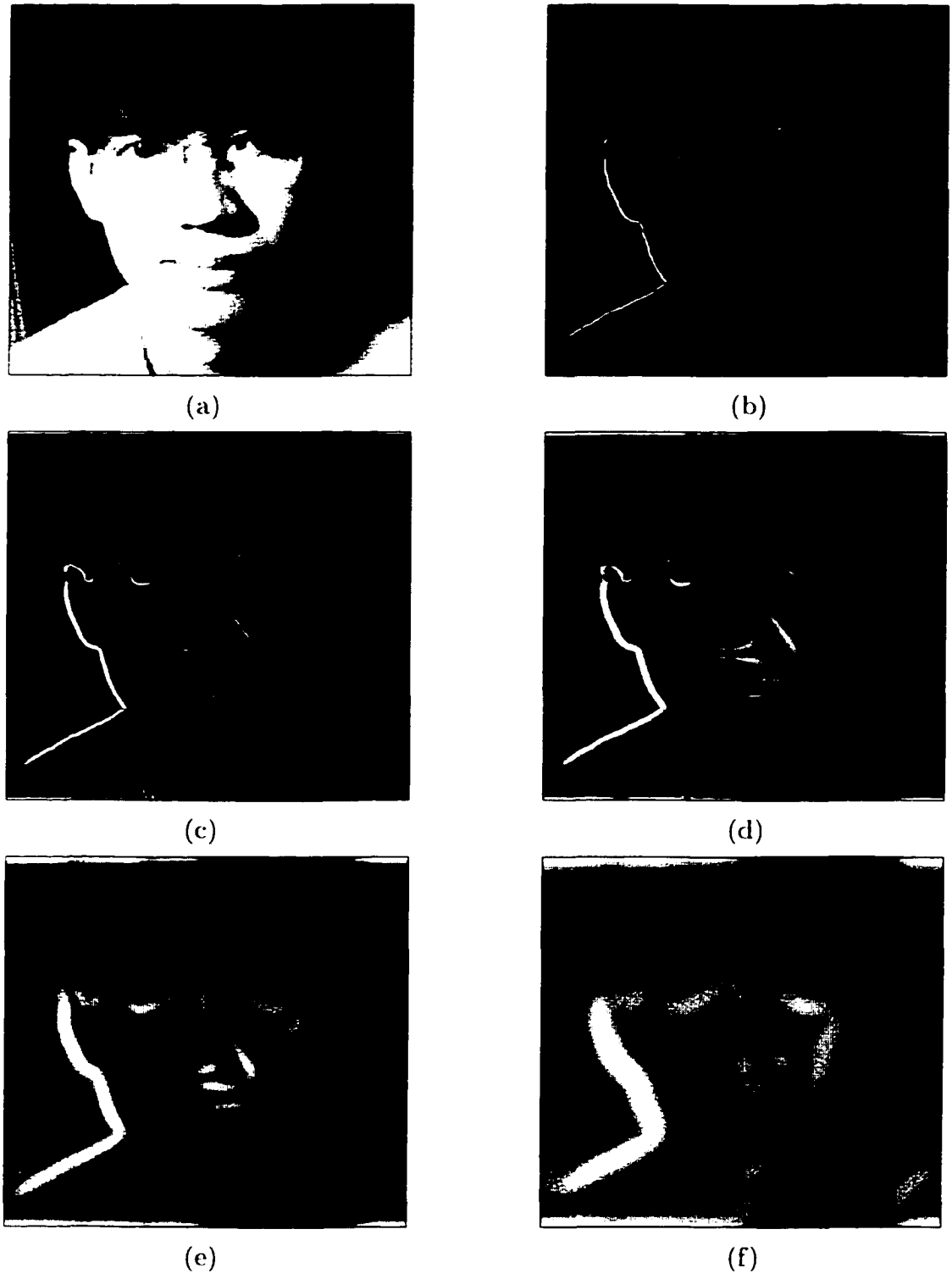


Figure 4.10 (a) Original image L, (b) to (f) gradient magnitude images $|\nabla L(\sigma; x, y)|$ with $\sigma = 0.5, 1, 2, 4$, and 8 respectively.

3. Summation of the multiscale gradient magnitude images:

$$\sum_{\sigma} |\nabla L(\sigma; x, y)|.$$

4. $|\nabla L|$ computed using the multiple scale differential masks:

$$|\nabla L| = \sqrt{L_x^2 + L_y^2},$$

where

$$L_x = L(x, y) * \sum_{\sigma} \hat{o}_x(\sigma; x, y)$$

and

$$L_y = L(x, y) * \sum_{\sigma} \hat{o}_y(\sigma; x, y).$$

The overall computational effort required to perform multiscale data fusion of the first three cases is approximately the same since $|\nabla L|$ at each scale must be computed individually. In the last case, only 2 convolutions are required between the image and multiple scale differential masks.

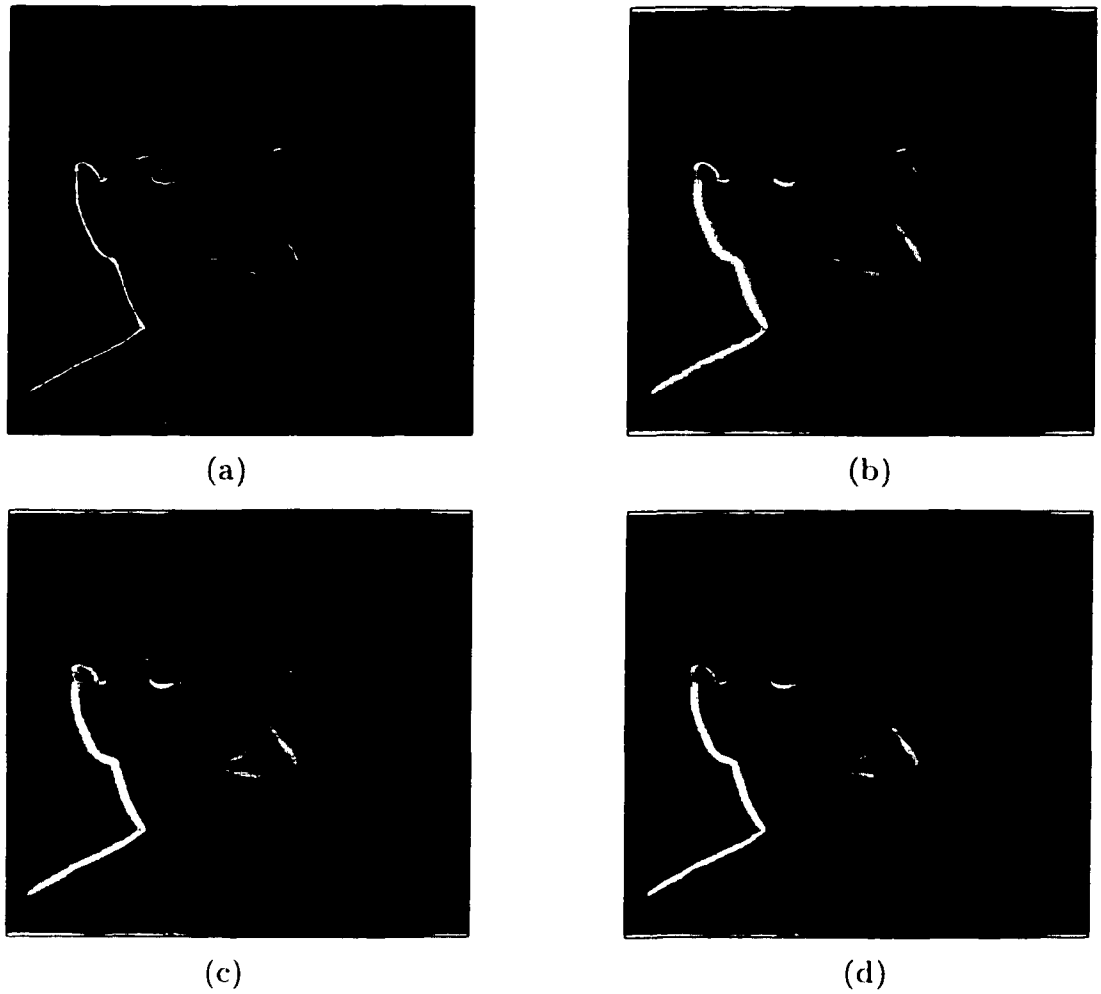
Results obtained from 4 multiscale data fusion algorithms are shown in Figure 4.11. As seen, $\max_{\sigma} |\nabla L(\sigma; x, y)|$ with edge localization in Figure 4.11-a provides the sharpest results while $\max_{\sigma} |\nabla L(\sigma; x, y)|$ without edge localization in Figure 4.11-b provides the most blurred data fusion image. The gradient image in Figure 4.11-d obtained using the multiple scale differential masks is comparable to that obtained using summation of the multiscale gradient images as seen in Figure 4.11-c. The quality of the gradient magnitude images in Figures 4.11-c and 4.11-d are in between those of $\max_{\sigma} |\nabla L(\sigma; x, y)|$ with and without edge localization. In Figures 4.11-c and 4.11-d, the shape preservation property is demonstrated by fine details at blurred edges of the shadow on the left shoulder and on the left side of the face. Unlike the first 2 algorithms that totally discard all weak details, details at all scales are preserved in the summation method and the method using the multiple scale differential masks.

4.3.1.2 Multiscale data fusion of Laplacian images

The Laplacian image $\nabla^2 L$ is defined as

$$\nabla^2 L = L_{xx} + L_{yy}.$$

Since the Laplacian operator is linear, summation of multiscale Laplacian images is equivalent to computing the Laplacian image using the multiple scale differential masks. Multiscale data fusion algorithms for Laplacian images are thus given below:



(a) (b)

(c) (d)

Figure 4.11 Multiscale data fusion images of $|\nabla L(\sigma; x, y)|$ of the image in Figure 4.10-a with σ varying from 0.5 to 7.81: (a) and (b) $\max_\sigma |\nabla L(\sigma; x, y)|$ with and without edge localization. (c) $\sum_\sigma |\nabla L(\sigma; x, y)|$. (d) $|\nabla L|$ computed using the multiple scale differential masks.

1. Maximization of $|\nabla^2 L|$:

$$\nabla^2 L(\sigma_{max}; x, y)$$

where

$$\sigma_{max} = \arg\max_{\sigma} |\nabla^2 L(\sigma; x, y)|.$$

2. Summation of the multiscale Laplacian images:

$$\sum_{\sigma} \nabla^2 L(\sigma; x, y) = L(x, y) * (\sum_{\sigma} \hat{o}_{xx}(\sigma; x, y) + \sum_{\sigma} \hat{o}_{yy}(\sigma; x, y)).$$

The computation of $\sum_{\sigma} \nabla^2 L(\sigma; x, y)$ requires only one convolutions.

Examples of multiscale data fusion of Laplacian images are demonstrated in Figures 4.12 to 4.20 . The original images in Figures 4.14-a, 4.17-a, and 4.20-a contain edge details with different degrees of smoothness. The images of $\nabla^2 L(\sigma; x, y)$ at several scales are shown in Figures 4.12-b to 4.12-f, 4.15-b to 4.15-e, and 4.18-b to 4.18-e while the images of $\text{sign}(\nabla^2 L(\sigma; x, y))$ are shown in Figures 4.13-b to 4.13-f, 4.16-b to 4.16-e, and 4.19-b to 4.19-e where the white areas represent the positive sign. While the results obtained using the small σ are prone to noise, those obtained using the large σ introduce more shape distortion. In these cases, the results obtained at a single scale fail to capture edge details. On the other hand, the results in Figures 4.14-c, 4.14-e, 4.17-c, 4.17-e, 4.20-c, and 4.20-e obtained using the multiscale data fusion methods clearly preserve outlines of the objects and remove nearby spurious details. These examples show how well details at different scales are combined to achieve both structure preservation and spurious noise suppression

In general, in traditional image analysis, diffusion of strong structures to the surrounding areas induced by smoothing process creates undesirable edge effect. However, the proposed multiscale data fusion algorithms exploit this diffusion phenomenon to eliminate spurious details surrounding the structures. According to these preliminary results, advantages and disadvantages of the tested multiscale data fusion methods are summarized in Table 4.2.

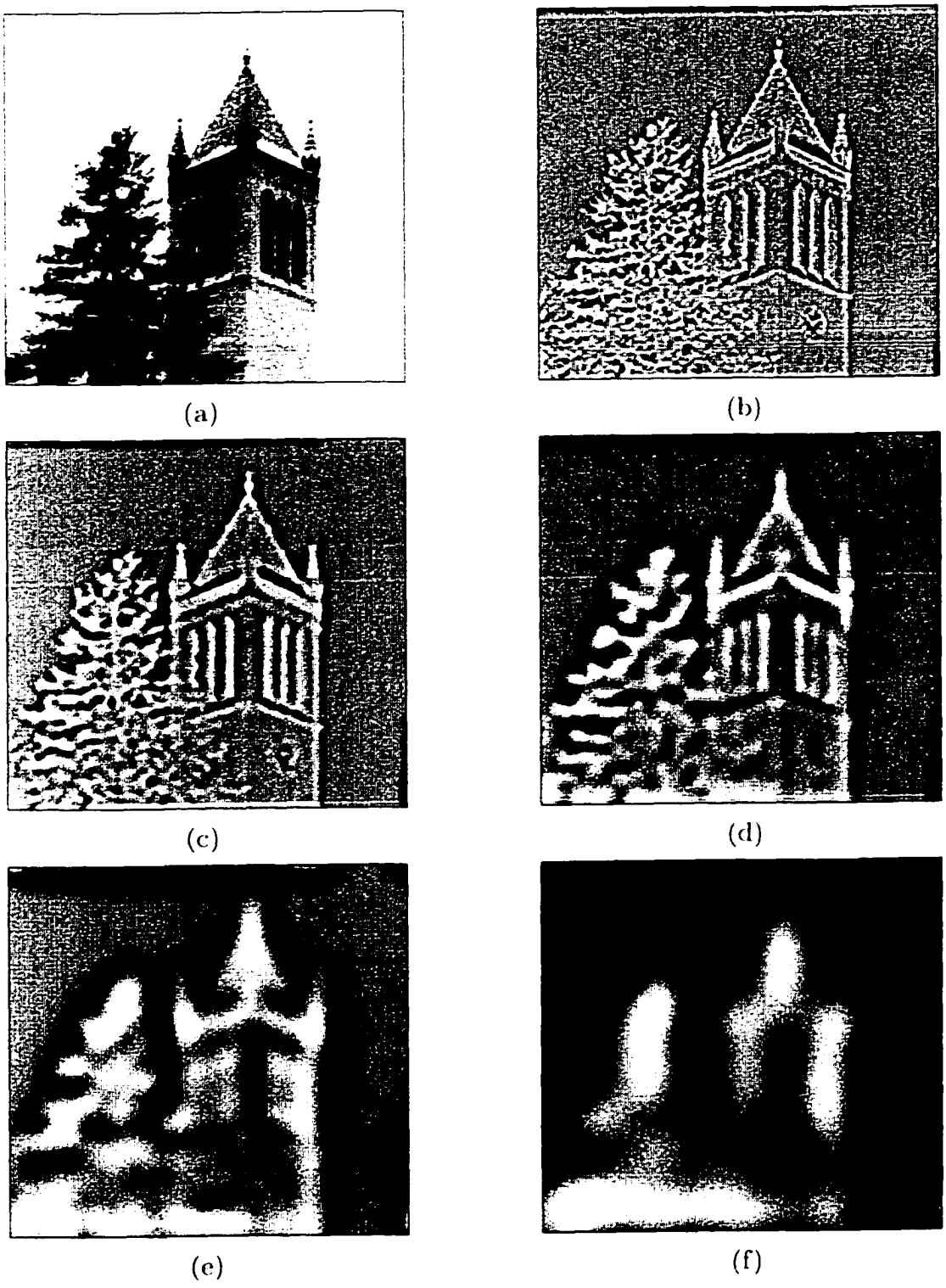


Figure 4.12 (a) Original image L. (b) to (f) Laplacian images $\nabla^2 L$ with σ = 1, 2, 4, 8, and 16 respectively.

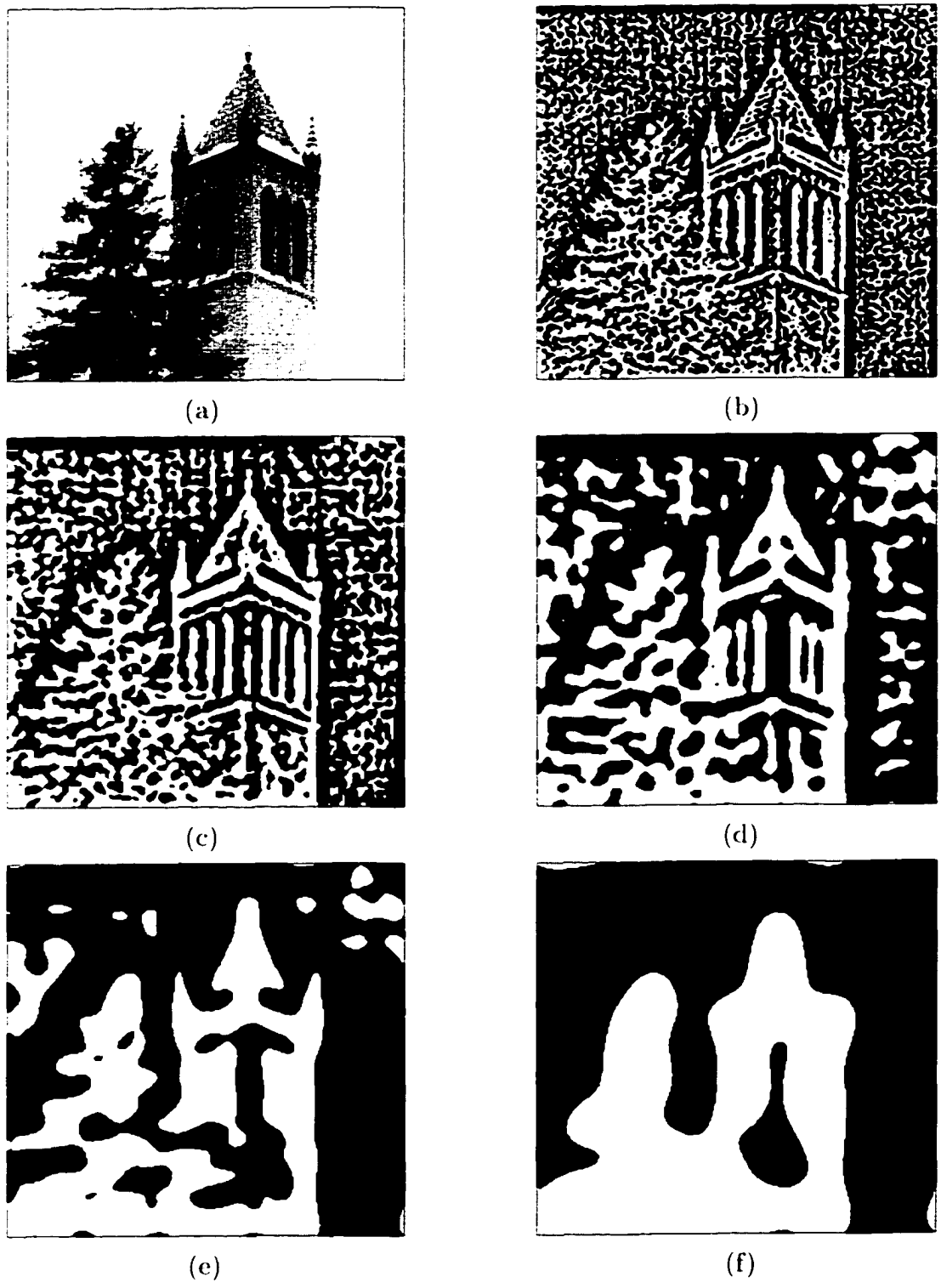
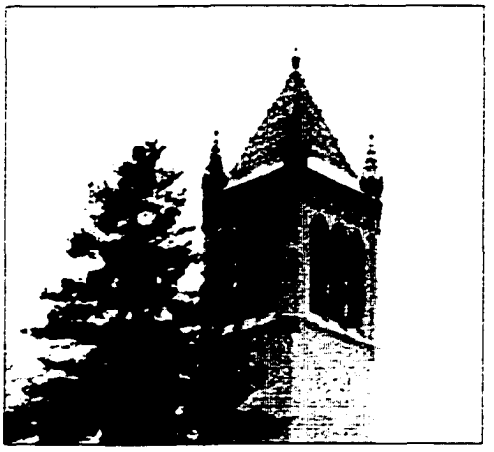


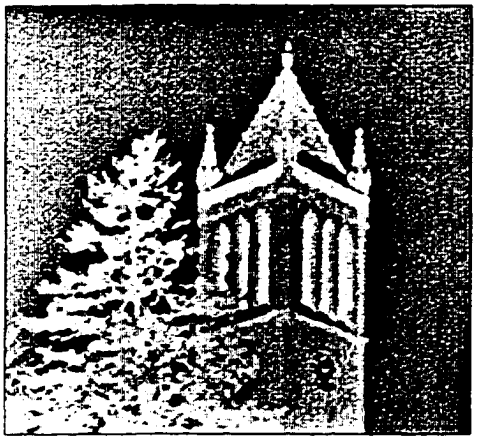
Figure 4.13 (a) Original image L. (b) to (f) images of $\text{sign}(\nabla^2 L(\sigma; x, y))$ with $\sigma = 1, 2, 4, 8,$ and 16 respectively.



(a)

(b)

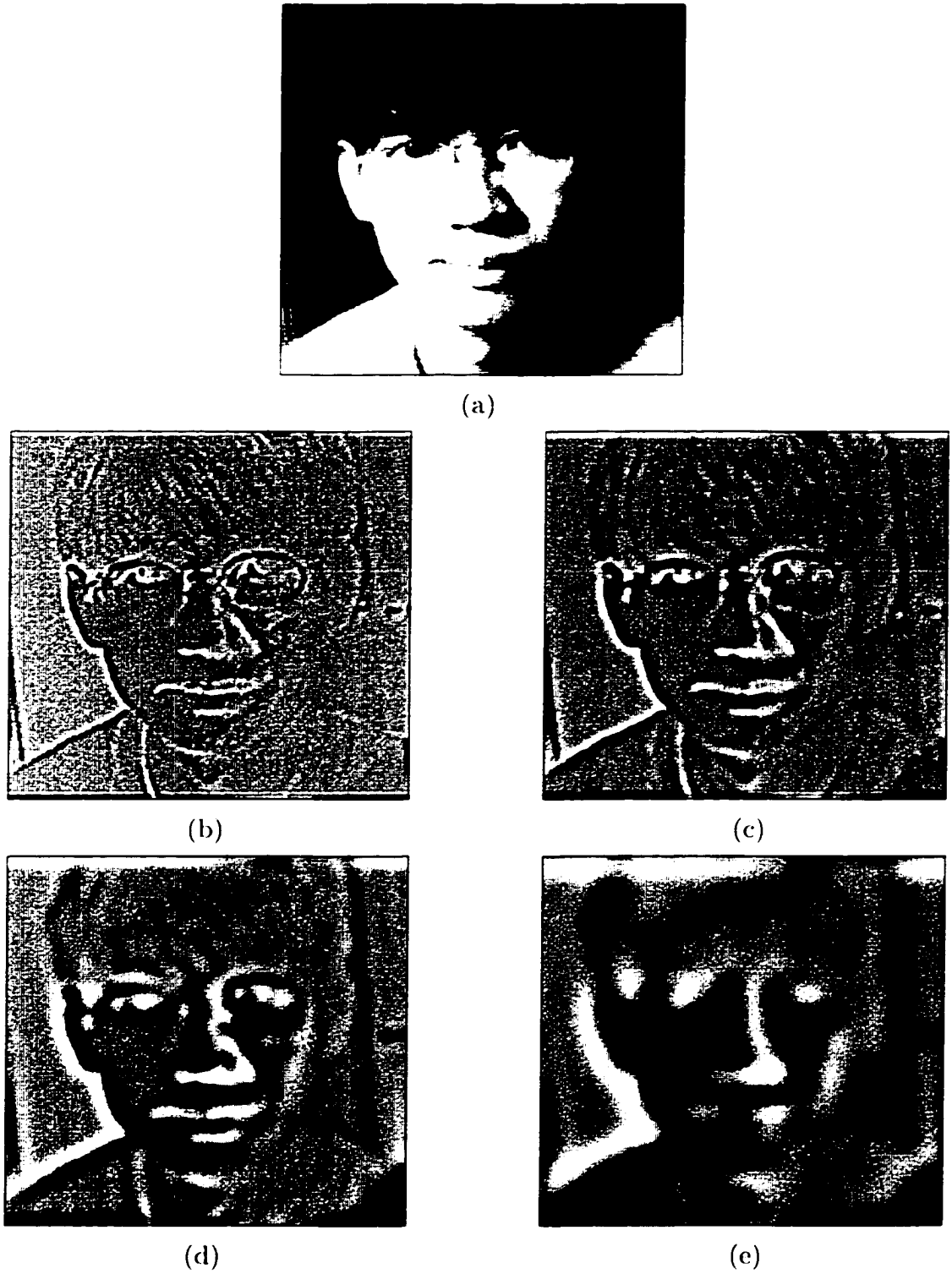
(c)



(d)

(e)

Figure 4.14 Multiscale data fusion images of $\nabla^2 L(\sigma_{max}; x, y)$ with σ varying from 0.5 to 15.77: (a) original image L, (b) $\nabla^2 L(\sigma_{max}; x, y)$, (c) $\text{sign}(\nabla^2 L(\sigma_{max}; x, y))$, (d) $\sum_\sigma \nabla^2 L$, (e) $\text{sign}(\sum_\sigma \nabla^2 L)$.



(a)

(b)

(c)

(d)

(e)

Figure 4.15 (a) Original image L. (b) to (e) Laplacian images $\nabla^2 L$ with σ = 1, 2, 4, and 8 respectively.

(a)

(b)

(c)

(d)

(e)

Figure 4.16 (a) Original image L. (b) to (e) images of $\text{sign}(\nabla^2 L(\sigma; x, y))$ with $\sigma = 1, 2, 4,$ and 8 respectively.

(a)

(b)

(c)

(d)

(e)

Figure 4.17 Multiscale data fusion images of $\nabla^2 L(\sigma_{max}; x, y)$ with σ varying from 0.5 to 8.1: (a) original image L. (b) $\nabla^2 L(\sigma_{max}; x, y)$. (c) $\text{sign}(\nabla^2 L(\sigma_{max}; x, y))$. (d) $\sum_\sigma \nabla^2 L$. (e) $\text{sign}(\sum_\sigma \nabla^2 L)$.

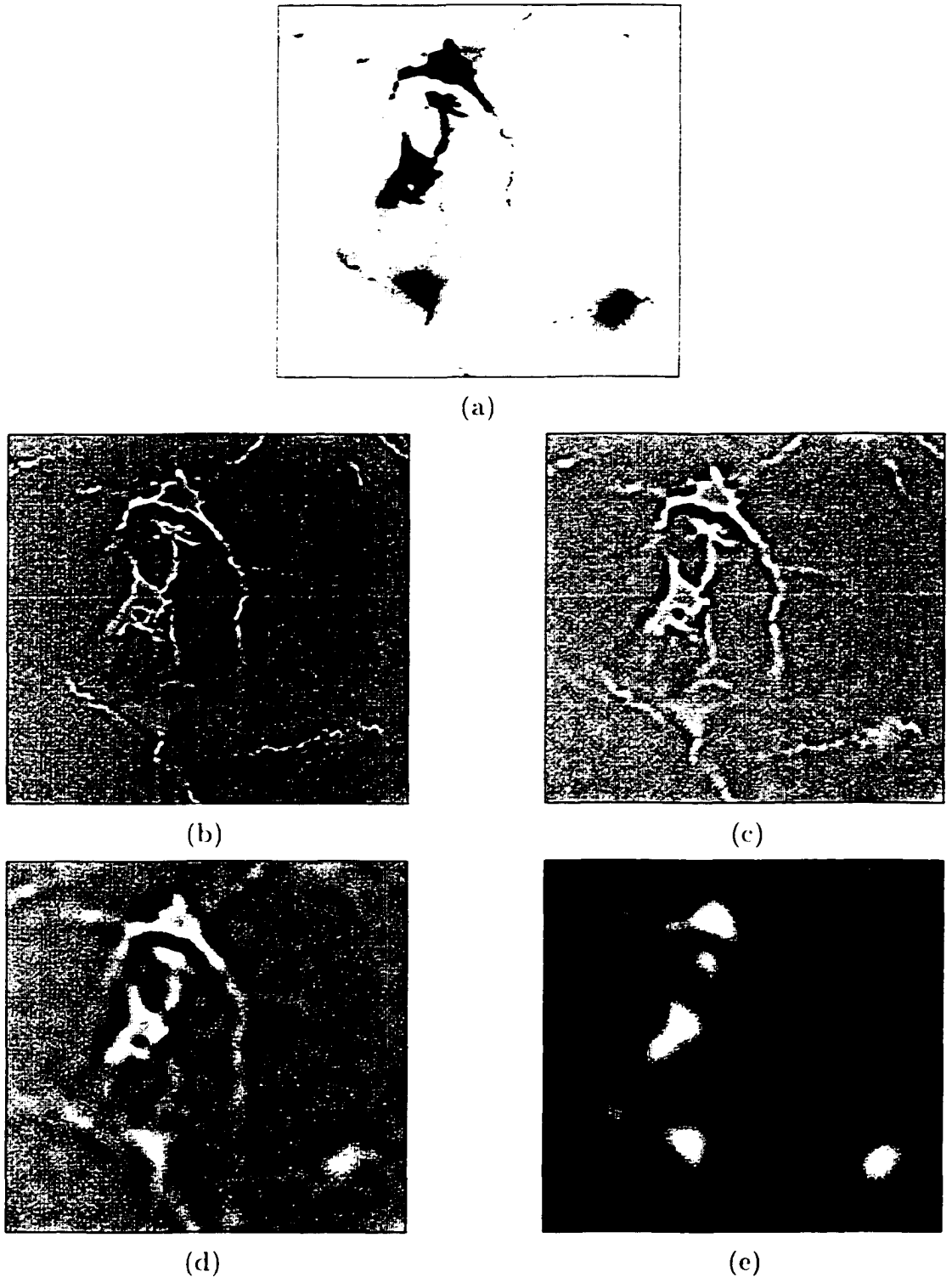
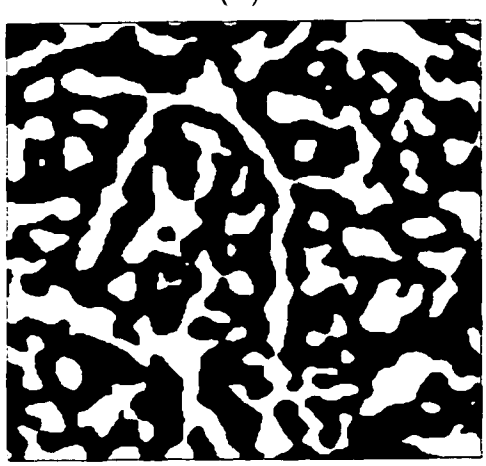


Figure 4.18 (a) Original image L (courtesy of John Kesterson, VayTek, Inc.). (b) to (e) Laplacian images $\nabla^2 L$ with $\sigma = 1, 2, 4,$ and 8 respectively.

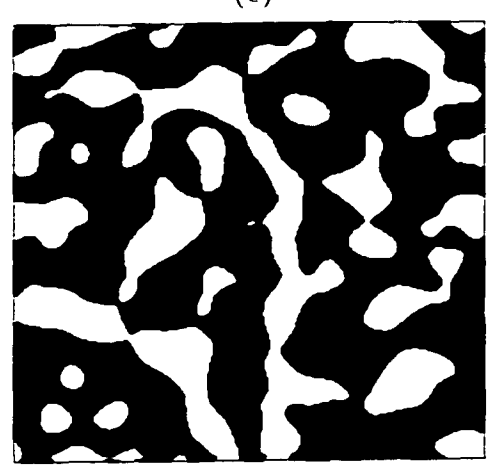
(a)

(b)

(c)



(d)

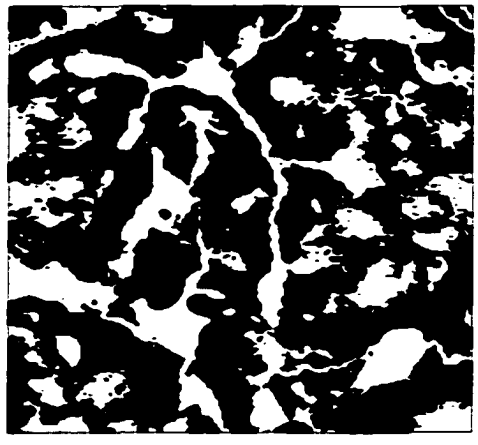


(e)

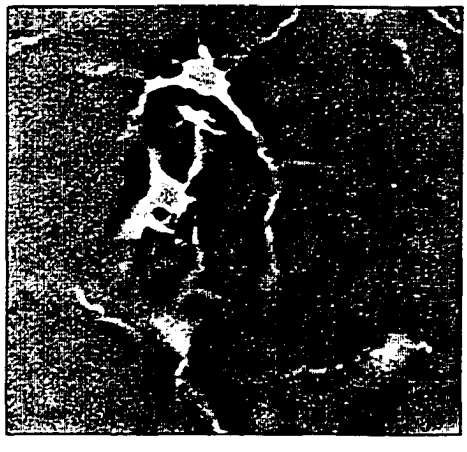
Figure 4.19 (a) Original image L. (b) to (e) images of $\text{sign}(\nabla^2 L(\sigma; x, y))$ with $\sigma = 1, 2, 4$, and 8 respectively.

(a)

(b)



(c)



(d)

(e)

Figure 4.20 Multiscale data fusion images of $\nabla^2 L(\sigma_{max}; x, y)$ with σ varying from 0.5 to 8.1: (a) original image L, (b) $\nabla^2 L(\sigma_{max}; x, y)$, (c) $\text{sign}(\nabla^2 L(\sigma_{max}; x, y))$, (d) $\sum_\sigma \nabla^2 L$, (e) $\text{sign}(\sum_\sigma \nabla^2 L)$.

Table 4.2 Summary of multiscale data fusion algorithms.

	Complexity	Computational cost	Scale analysis
$\Psi(\sigma_{max}; L(x,y))$ without feature localization	Low	High	Yes
$\Psi(\sigma_{max}; L(x,y))$ with feature localization	High	High	Yes
$\sum_{\sigma} \Psi(\sigma; L(x,y))$	Low	High	No
$\Psi(L(x,y))$ obtained using multiple scale differential masks	Low	Low	No

4.4 Conclusions

The local scale analysis method proposed in this chapter is based on multiscale feature matching. The multiscale differential operations are considered as matching between image derivatives and integrations of the masks. The integration norms are used as normalization factors of the multiscale differential masks. Information of scales, locations, and strength of structures are obtained from local maxima of the multiscale differential structure images. In the second part, 4 multiscale data fusion algorithms are proposed where details obtained at all scales are treated equally by using a scale invariant normalization scheme. The most striking features of the method using multiple scale differential masks are the low computational cost needed for the algorithm and the shape preservation property. The method requires only few convolutions to compute the differential structure images in contrast to other methods that compute details at each scale. In addition, the results obtained using multiple scale differential masks preserve all strong details at all scales of interest with little shape distortion. The use of multiple scale differential masks also increases scale selection tolerance since the computation of the mask is performed using a range of scales instead of a single scale value.

CHAPTER 5. BOUNDARY EXTRACTION ALGORITHM

The previous chapters deal with the selection of suitable multiscale differential masks and scale parameters to generate feature images, especially, the differential structures of an image. The next step in image analysis is to extract the information into organized structures using a boundary extraction algorithm that provides complete information of object boundaries essential for image analysis.

5.1 Background

In the previous work [22], a boundary extraction algorithm was developed based on a particle model in two orthogonal velocity fields. In general, a particle P in a system can be described by the 4-tuple

$$P = (\vec{p}, m, \vec{v}, \vec{F}),$$

where $\vec{p}$, m, and $\vec{v}$ are position, mass, and velocity of a particle and $\vec{F}$ is a force acting on the particle as shown in Figure 5.1. When $\vec{F}$ and m are neglected, the particle position in a first order system is given by

$$\vec{p}(t) = \vec{p}(0) + \vec{v}t, \tag{5.1}$$

where $\vec{p}(0)$ is an initial position. In a discrete case, Equation 5.1 is replaced by

$$\vec{p}_{k+1} = \vec{p}_k + \Delta t \vec{v}_k, \tag{5.2}$$

where Δt is the time step and $\vec{p}_k$ and $\vec{v}_k$ denote the particle position and velocity at the k^{th} time step. Equation 5.2 is well suited for boundary extraction where the velocity field is derived from the image. Generally, $\vec{v}$ depends on position and time. In this research, however, since all fields derived from an image are static, $\vec{v}$ is a function of position alone. On the right hand side of Equation 5.2, $\vec{v}_k$ at position $\vec{p}_k$ is needed to be observable. Therefore, for discrete velocity fields derived from an image, a polynomial surface approximation technique is employed for obtaining the field values at arbitrary

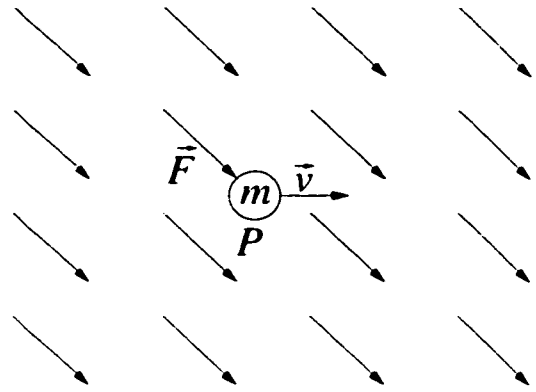


Figure 5.1 Particle P in a force field $\vec{F}$.

positions. Surface approximation techniques used in this research are described in Appendix A.

5.1.1 A Model of Particle Motion in a Combined Orthogonal Velocity Field

Consider the differential representation of a curve by two sets of vectors tangential and normal to the curve. In general, a curve can be uniquely represented using two differential components directed tangential and normal to the curve [54]. In discrete representation, a curve can be expressed by the recursive equation

$$\vec{p}_{k+1} = \vec{p}_k + \alpha_k \vec{t}_k + \beta_k \vec{n}_k. \tag{5.3}$$

as shown in Figure 5.2 where $\vec{p}_k$ is the k^{th} point of the curve, $\vec{t}_k$ and $\vec{n}_k$ are unit vectors tangential and normal to the curve at $\vec{p}_k$ respectively while α_k and β_k are constant coefficients. Based on Equation 5.3, two sets of vectors tangential and normal to the boundary are required for boundary representation.

By combining Equations 5.2 and 5.3, the particle model proposed in the previous work is given by

$$\vec{p}_{k+1} = \vec{p}_k + \alpha \vec{e}_k + \beta \vec{n}_k. \tag{5.4}$$

where α and β are tangential and normal stepping factors respectively and $\vec{e}$ and $\vec{n}$ are the edge vector field and the normal compressive velocity field tangential and normal to object boundaries respectively. The edge vector field $\vec{e}$ is defined as the normalized Hamiltonian gradient vector field:

$$\vec{e} = \frac{1}{c}(L_y \vec{i} - L_x \vec{j}) \tag{5.5}$$

$$= (\nabla L)^{\perp}, \tag{5.6}$$

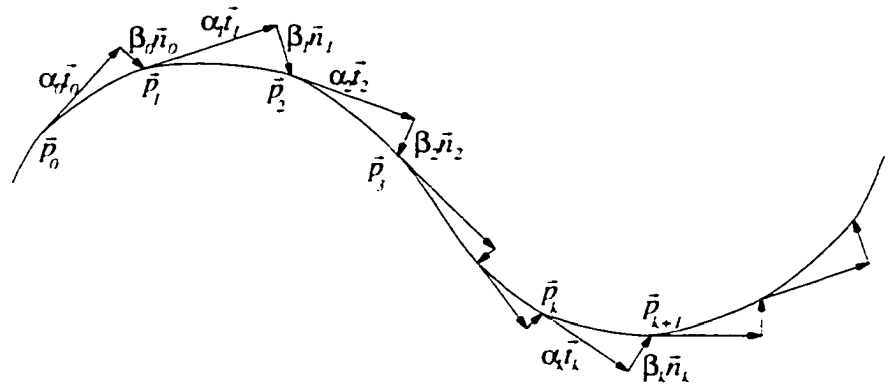


Figure 5.2 Discrete representation of a curve by a set of tangent and normal vectors.

where $()^{\perp}$ denotes the 90-degree rotation transformation, $c = \max|\nabla L|$, and $\vec{i}$ and $\vec{j}$ are unit vectors in x and y directions respectively. An example of the edge vector field obtained using the Gaussian differential masks with $\sigma = 1$ is shown in Figure 5.3-a where edge vectors tangential to the object boundaries form the "edge currents" flowing around the object.

For the normal compressive velocity field $\vec{n}$, 2 vector fields, namely the normalized gradient of the gradient magnitude image and the normalized Laplacian-gradient vector fields, were used in the previous work. The normalized gradient of the gradient magnitude image is computed as

$$\begin{aligned}\frac{1}{c}\nabla(|\nabla L|) &= \frac{1}{c}\nabla(\sqrt{L_x^2 + L_y^2}) \\ &= \frac{1}{c\sqrt{L_x^2 + L_y^2}}((L_x L_{xx} + L_y L_{xy})\vec{i} + (L_x L_{xy} + L_y L_{yy})\vec{j})\end{aligned}$$

while the normalized Laplacian-gradient vector field is defined as

$$\frac{1}{c}\nabla L \cdot \nabla^2 L = \frac{1}{c}(L_{xx} + L_{yy})(L_x\vec{i} + L_y\vec{j})$$

where c is the normalization factor, $c = \max|\nabla(|\nabla L|)|$ in the former case and $c = |\max \nabla L \cdot \nabla^2 L|$ in the latter case. Figures 5.3-c and 5.3-e show examples of $\frac{1}{c}\nabla|\nabla L|$ and $\frac{1}{c}\nabla^2 L \cdot \nabla L$ respectively computed using the Gaussian differential masks with $\sigma = 1$. In the case of $\frac{1}{c}\nabla|\nabla L|$, since $|\nabla L|$ has its maximum gradient path along the object boundary, $\frac{1}{c}\nabla|\nabla L|$ points toward the nearest object boundaries. Similarly, $\nabla^2 L$ has opposite signs in the areas inside and outside the object. As a result, $\frac{1}{c}\nabla^2 L \cdot \nabla L$ points towards the nearest object boundaries. This is clearly demonstrated in both Figures 5.3-c and 5.3-e.

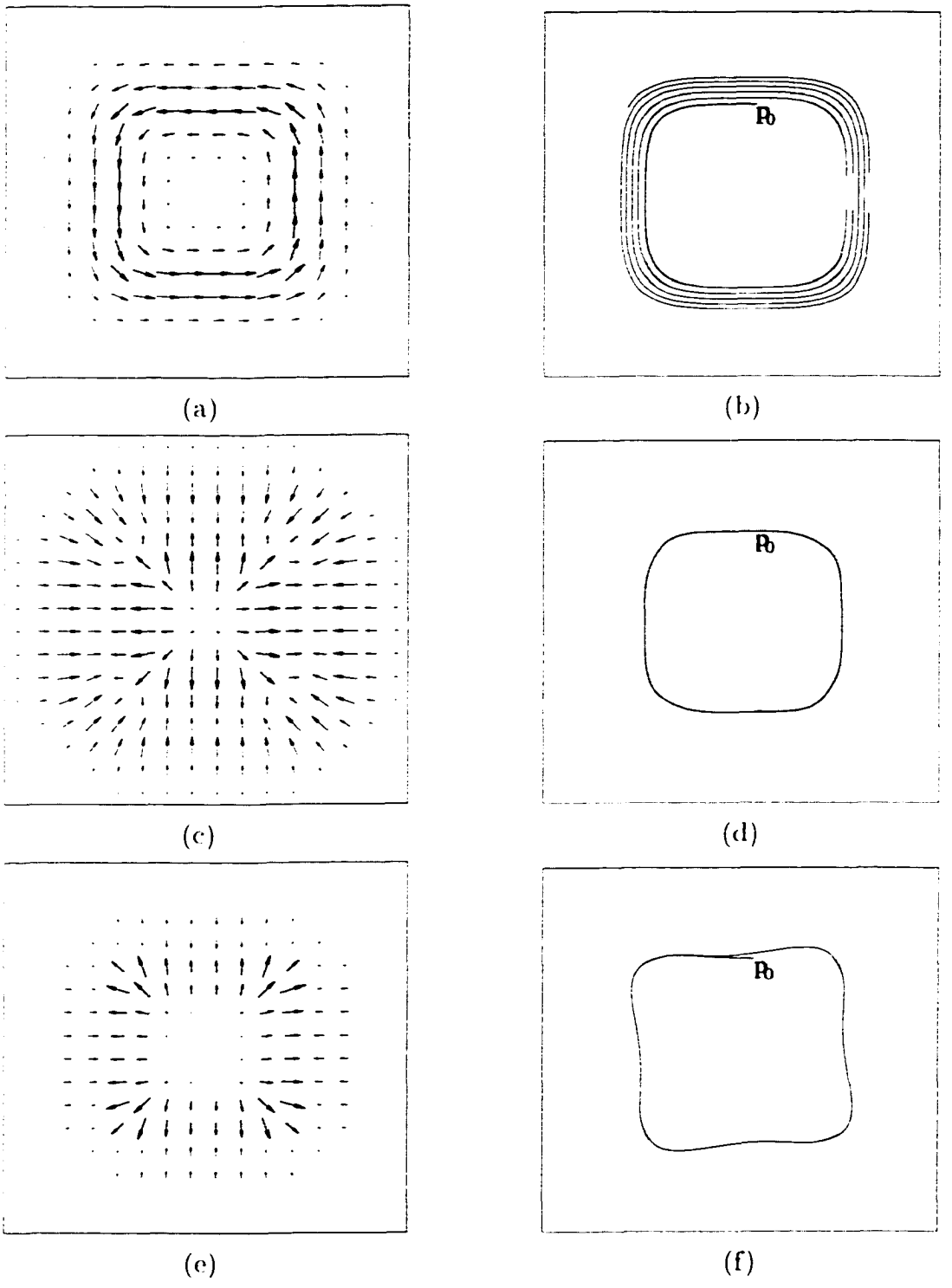


Figure 5.3 Particle trajectories based on Equation 5.4: (a) edge vector field. (b) trajectory with $\alpha = 0.1$ and $\beta = 0$. (c) $\frac{1}{c}\nabla|\nabla L|$. (d) trajectory with $\alpha = 0.1$, $\beta = 0.1$, and $\vec{n} = \frac{1}{c}\nabla|\nabla L|$. (e) $\frac{1}{c}\nabla^2 L \cdot \nabla L$. (f) trajectory with $\alpha = 0.1$, $\beta = 0.1$, and $\vec{n} = \frac{1}{c}\nabla^2 L \cdot \nabla L$.

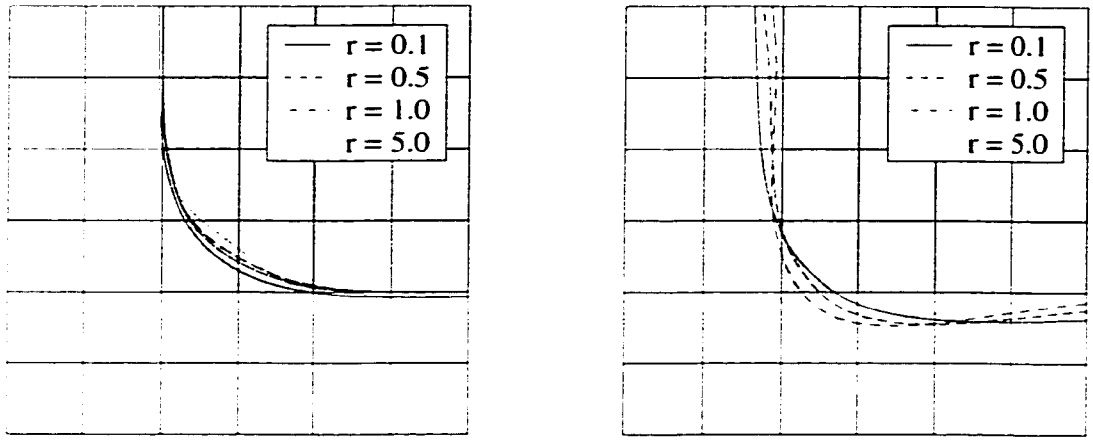
In Equation 5.4, while the edge vector field $\vec{e}$ drives a particle along the object boundary, the normal velocity field $\vec{n}$ forces the particle to stay within the confines of the boundary. In terms of edge localization, the edge vector field is used to locate an edge in the direction tangential to the edges while the normal compressive velocity field localizes the edge in the direction normal to the edge. The particle trajectory obtained using Equation 5.4 with $\alpha = 0.1$ and $\beta = 0$ is displayed in Figure 5.3-b where $\vec{p}_0$ is a starting point. Clearly the absence of the normal compressive velocity field results in a spiral trajectory pattern. The particle trajectories obtained using Equation 5.4 with $\alpha = 0.1$, $\beta = 0.1$, and $\vec{n} = \frac{1}{c}\nabla|\nabla L|$ and $\vec{n} = \frac{1}{c}\nabla^2 L \cdot \nabla L$ are shown in Figures 5.3-d and 5.3-f respectively. In all examples in Figure 5.3, bilinear interpolation was used to estimate the field values. Obviously, the combination of the edge vector field and the normal compressive velocity field yields closed trajectories confined to object boundaries.

5.1.2 Particle Trajectories in a Combined Orthogonal Velocity Field

From Equation 5.4, we see that the particle trajectory depends on the starting point, scale of the masks σ, the tangential and normal compressive velocity fields, and parameters α and β. Figure 5.4 illustrates how the particle follows the object boundary at the object corner using various ratios $\frac{\beta}{\alpha}$. In the case when $\vec{n} = \frac{1}{c}\nabla|\nabla L|$, the ratio $\frac{\beta}{\alpha}$ has little effect on the resulting trajectories. In contrast, when $\vec{n} = \frac{1}{c}\nabla^2 L \cdot \nabla L$, the significance of $\frac{\beta}{\alpha}$ can be observed. As seen in Figure 5.4-b, the larger the ratio $\frac{\beta}{\alpha}$, the closer the trajectory to the object edge. The asymmetrical property of the particle trajectory can also be seen in Figure 5.4-b.

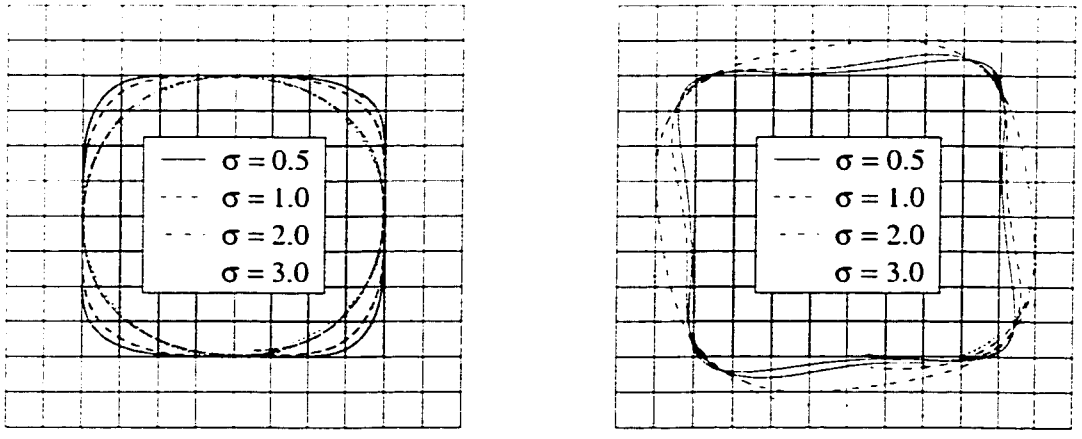
Another important parameter is the scale σ of the masks used to generate the fields. Figure 5.5 shows the effect of σ on the particle trajectories where multiscale Gaussian differential masks with different values of σ were used to generate the fields. Both Figures 5.5-a and 5.5-b demonstrate that the larger the scale, the smoother the trajectory resulting in poor edge localization. One obvious difference between the trajectory obtained using $\vec{n} = \frac{1}{c}\nabla|\nabla L|$ and that obtained using $\vec{n} = \frac{1}{c}\nabla^2 L \cdot \nabla L$ is that in the first case, the obtained trajectory tends to be smaller than the actual object boundary particularly at corners while in the latter case, the particle deviates away from the object boundary at corners resulting in a trajectory larger than the actual object boundary.

The object boundaries can be retrieved by following the convergent paths of the particle trajectories. In order to allow the particle to converge to the true boundaries rapidly, local maxima of the gradient magnitude image are used as initial points. By using both positive and negative values of the tangential stepping factor, the algorithm



(a) (b)

Figure 5.4 Particle trajectories at the object corner obtained using Equation 5.4 with $\sigma = 1.0$: (a) $\vec{n} = \frac{1}{c}\nabla|\nabla L|$ and (b) $\vec{n} = \frac{1}{c}\nabla^2 L \cdot \nabla L$.



(a) (b)

Figure 5.5 Particle trajectories obtained using Equation 5.4 with $\alpha = 0.1$ and $\beta = 0.1$: (a) $\vec{n} = \frac{1}{c}\nabla|\nabla L|$ and (b) $\vec{n} = \frac{1}{c}\nabla^2 L \cdot \nabla L$.

can be extended to boundary extraction of multiple objects.

5.1.3 Disadvantages of the Previous Particle Model

The boundary extraction algorithm based on particle motion in a combined orthogonal velocity field in the previous work has several advantages. Since the positions of a particle in Equation 5.4 are real and can be computed using an arbitrary step size, this algorithm provides results with subpixel resolution. Due to its sequential edge linking scheme, the method is topologically unrestricted and requires simple initialization. In addition, the boundaries obtained by the method are guaranteed to be connected. The method is also fast and simple. Nevertheless, the algorithm also suffers from several sources of edge localization errors.

1. **Edge localization error due to the particle motion equation.** Equation 5.4 does not exploit the optimization algorithm for edge localization. In Equation 5.4, the particle position is updated in both tangential and normal direction in each time step. Hence, before the particle trajectory converges to the object boundary the particle has already traveled some distance in the tangential direction. This causes significant distortion at corners. In other words, the particle model based on Equation 5.4 has an "inertial force" preventing the particle to follow the object boundaries in the areas of high curvature accurately.

2. **Edge localization error due to the normal compressive velocity field.** Most edge localization errors occur at corners and junctions. The trajectory obtained using the normalized gradient of the gradient magnitude image tends to cut through the corners while that obtained using the normalized Laplacian-gradient vector field tends to deviate away from the corners.

3. **Edge localization error due to the size of the differential masks.** The effect of the mask size on the detected object edges is well known in image processing. Using small size differential masks, accurate boundaries can be obtained but the results are prone to noise. On the other hand, large size masks yield results with better noise suppression but poorer edge localization.

In addition to edge localization error, the representation of the particle trajectory is less efficient since all computed positions of the particle are recorded as boundary points.

5.2 New Model for Boundary Extraction

At the heart of boundary extraction is edge localization. The performance of boundary extraction algorithms is usually determined in terms of the accuracy of edge localization. In general, for sequential boundary extraction, two edge localization steps, one in tangential and another in normal direction of object boundaries, are necessary. In the previous work, the edge vector field orthogonal to image gradient is used to direct the particle along object boundaries in the tangential direction while the normal compressive velocity field forces the particle to rest on the boundary. The overall performance of the algorithm depends significantly on the choice of the fields and edge localization methods. A generalized framework for deriving the tangential and normal compressive velocity fields and a new edge localization scheme for overcoming the earlier disadvantages are proposed next.

5.2.1 Fields for Boundary Localization

The fields used for edge localization are refered to as *tangential* and *normal compressive velocity fields*, $\vec{t}$ and $\vec{n}$, respectively. The tangential velocity field must have all flux lines parallel or directed tangential to the nearest boundary edge while all flux lines in the normal compressive velocity field must point toward the nearest boundary edge. Only the fields that satisfy these conditions are useful for boundary extraction.

The normal compressive velocity field has the unique compressive property where all flux lines are directed to the proximate object boundary. An example of the field having this property is the gradient of gradient magnitude image, $\nabla(|\nabla L|)$. Since the maximum gradient path always occurs at an object boundary and $\nabla(|\nabla L|)$ always points towards the areas of the higher gradient magnitude, $\nabla(|\nabla L|)$ always points to the boundary. In the case of the Laplacian-gradient vector field $\nabla^2 L \cdot \nabla L$, the compressive property is due to the multiplication

$$\vec{n} = \lambda \cdot \vec{\gamma} \tag{5.7}$$

where λ is the *scalar boundary localization field* with its sign indicating the side with respect to the boundary and $\vec{\gamma}$ is the *normal velocity field* with its flux lines oblique or orthogonal to the object boundaries in one direction, similar to the gradient field. Principal choices of λ are $L_{ww} = \frac{L_x^2 L_{xx} + 2L_x L_y L_{xy} + L_y^2 L_{yy}}{L_x^2 + L_y^2}$ and $\nabla^2 L$. For example, in the case of the Laplacian-gradient vector field, $\lambda = \nabla^2 L$ and $\vec{\gamma} = \nabla L$. Using the boundaries as dividing lines, sign(λ) determines if a point is inside or outside the objects. The change of sign(λ) is used to locate the boundaries. Figures 5.6-c and 5.7-c show examples of L_{ww}

and $\nabla^2 L$ computed using the Gaussian differential masks with $\sigma = 1.0$. Assuming that the objects are darker than background, in both Figures 5.6-c and 5.7-c, the positive signs indicate the pixels inside the object while the negative signs indicate the pixels outside the object.

For the normal velocity field $\vec{\gamma}$, the principal choice is the gradient field ∇L as shown in Figures 5.6-b and 5.7-b. Although, the gradient flux lines are orthogonal to the boundaries, the gradient field does not have the compressive property since all local gradient vectors at pixels inside and outside the object point in the same direction of the steepest inclination. On the other hand, even though, the boundaries can be detected by observing the changes of sign(λ), no direction information can be obtained directly from λ. However, when $\vec{\gamma}$ is multiplied by λ, the resultant vector field $\vec{n}$ becomes the normal compressive velocity field and all vectors in the field point to the nearest boundary edge. Figure 5.6-d shows the results for $\vec{n} = L_{ww} \cdot \nabla L$ and Figure 5.7-d presents the results for $\vec{n} = \nabla^2 L \cdot \nabla L$. Therefore, the complete information for edge localization in both normal direction and location can be obtained from the normal compressive velocity field. Once the normal velocity field $\vec{\gamma}$ is given, $\vec{t}$ can be obtained by rotating $\vec{\gamma}$ by 90 degrees, that is,

$$\vec{t} = (\vec{\gamma})^{\perp} \tag{5.8}$$

$$= \gamma_y \vec{i} - \gamma_x \vec{j}. \tag{5.9}$$

where γ_x and γ_y are x and y components of $\vec{\gamma}$.

In addition to computing $\vec{t}$ and $\vec{n}$, λ and $\vec{\gamma}$ can also be used for determining the boundary supports. Let μ be defined as

$$\mu = \frac{\nabla L}{|\nabla L|} \cdot \frac{\nabla \lambda}{|\nabla \lambda|}. \tag{5.10}$$

The boundary support regions are determined by

$$\mu \leq 0. \tag{5.11}$$

The examples of $\nabla\lambda$ where $\lambda = L_{ww}$ and $\lambda = \nabla^2 L$, computed using the Gaussian differential masks with $\sigma = 1.0$ are shown in Figures 5.6-e and 5.7-e while images of μ are shown in Figures 5.6-f and 5.7-f where the dark areas represent the boundary support regions.

5.2.2 New Edge Localization Method: 2-Step Approach

The purpose of using 2 velocity fields, one tangential and another orthogonal to the boundaries, is that the first field provides information in the tangential direction

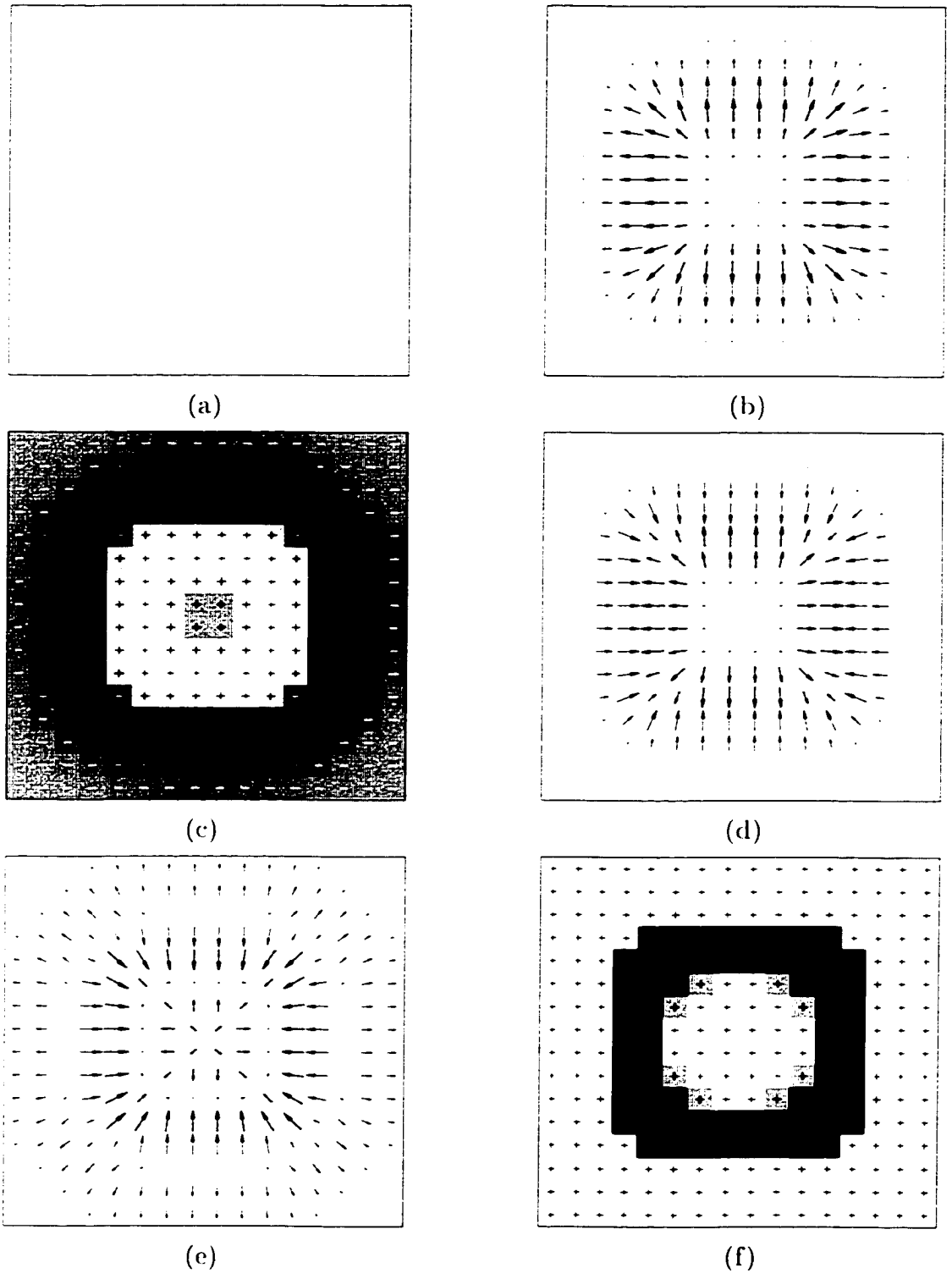


Figure 5.6 (a) Original image L. (b) ∇L. (c) L_{ww}. (d) $L_{ww}\nabla L$. (e) ∇L_{ww}. (f) $\mu = \frac{\nabla L}{|\nabla L|} \cdot \frac{\nabla L_{ww}}{|\nabla L_{ww}|}$.

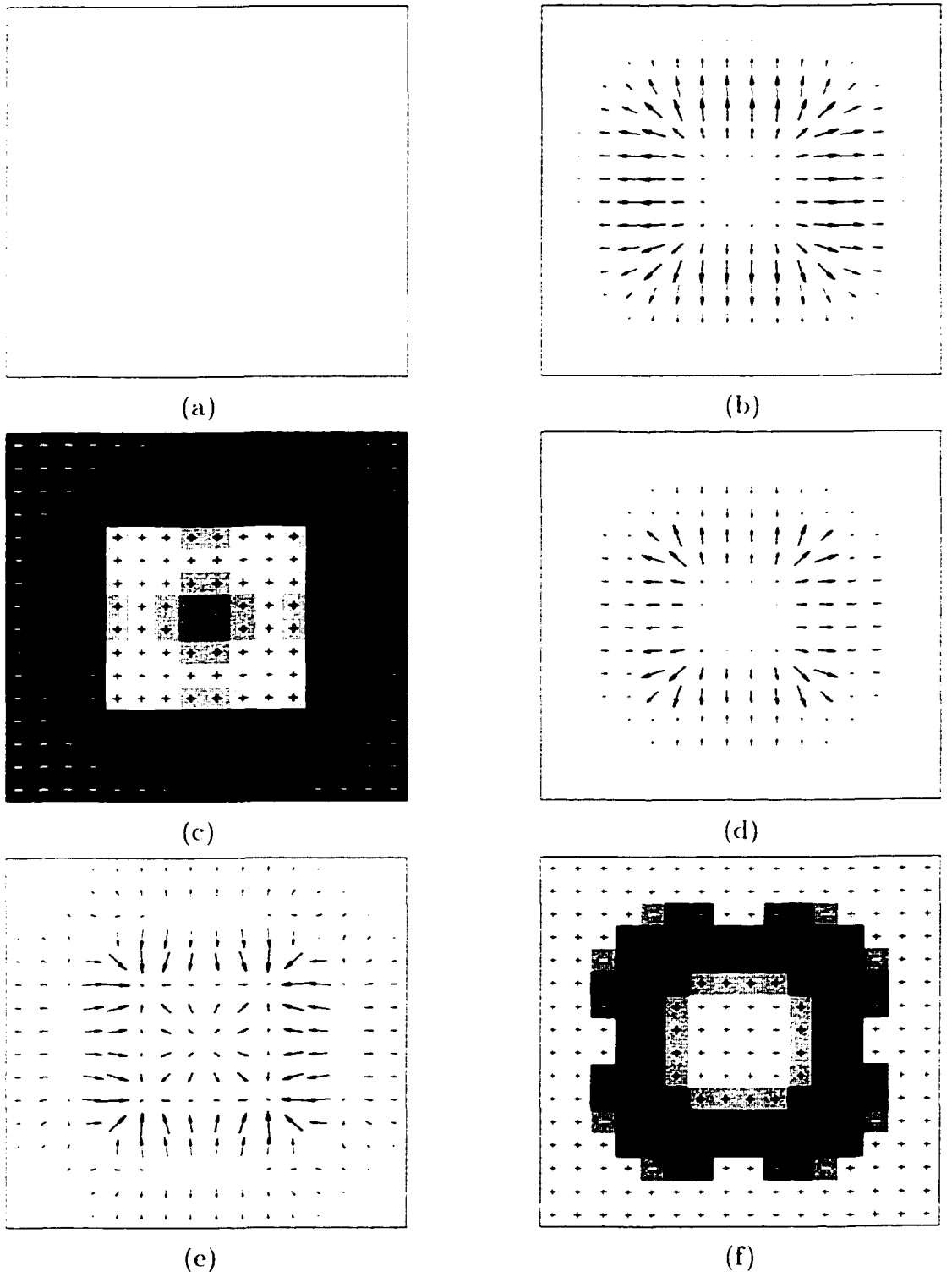


Figure 5.7 (a) Original image L, (b) ∇L, (c) $\nabla^2 L$, (d) $\nabla^2 L \nabla L$, (e) $\nabla(\nabla^2 L)$, (f) $\mu = \frac{\nabla L}{|\nabla L|} \cdot \frac{\nabla(\nabla^2 L)}{|\nabla(\nabla^2 L)|}$.

while the second one is used to locate the boundaries in the normal direction. In this research, a 2-step approach is proposed to improve edge localization performance. Edge localization in the direction tangential to the boundary is used to extract the boundary while edge localization in the normal direction provides an adjustment to improve edge localization.

Step 1

For the given tangential and normal compressive velocity fields, $\vec{t}$ and $\vec{n}$, the updating equation for the particle position in the tangential direction is given by

$$\vec{p}_k^{(0)} = \vec{p}_{k-1} + \alpha \frac{\vec{t}_{k-1}}{|\vec{t}_{k-1}|}. \tag{5.12}$$

where $\vec{t}_{k-1}$ is the tangential velocity at the previous position $\vec{p}_{k-1}$ and α is a tangential step size. After updating the particle position in the tangential direction, $\vec{p}_k^{(0)}$ is not guaranteed to lie on the boundary. The minor position adjustment in the direction normal to the boundary is required.

Step 2

Once, the position of the particle is updated in the direction tangential to the boundary, the particle will be forced to move only in the direction normal to the boundary. The normal direction $\vec{u}_0$ at the initial point $\vec{p}_k^{(0)}$ is approximated by the unit vector:

$$\vec{u}_0 = \frac{(-\vec{t}_k^{(0)})^{\perp}}{|\vec{t}_k^{(0)}|} \tag{5.13}$$

$$= \frac{\vec{n}_k^{(0)}}{|\vec{n}_k^{(0)}|}. \tag{5.14}$$

The particle position is then allowed to be updated only along the line l:

$$l(z) = \vec{p}_k^{(0)} + \vec{u}_0 z, \; z \in \mathbf{R}.$$

Edge localization in the normal direction is performed by minimizing $|\lambda|$ along the line l by the recursive equation:

$$\vec{p}_k^{(i)} = \vec{p}_k^{(i-1)} + \beta(\vec{u}_0 \cdot \vec{n}_k^{(i-1)})\vec{u}_0. \tag{5.15}$$

where $\vec{n}_k^{(i-1)}$ is the normal compressive velocity at the position $\vec{p}_k^{(i-1)}$ and β is a normal stepping factor, $\beta > 0$. The position $\vec{p}_k^{(i)}$ is updated until $|\lambda|$ approaches zero or the

number of iterations i exceeds the maximum number of iterations allowed. $\vec{p}_k^{(i_{max})}$ will be used as the initial position for the next edge localization step in the tangential direction. Unlike the previous particle model, the shapes of the boundaries obtained using this method will not be affected by α and β.

For a given boundary point p_0, the tangential step size α, and the normal step factor β, the overall pseudo code for determining the next boundary point p_1 is given below:

1. $p_1 := p_0 + \alpha \frac{\vec{t}(p_0)}{|\vec{t}(p_0)|}$.
2. $\vec{u}_0 := \frac{(-\vec{t}(p_1))^{\perp}}{|\vec{t}(p_1)|}$.
3. **for** i := 1 to N_{max}
4. $\quad p_1 := p_1 + \beta(\vec{u}_0 \cdot \vec{n}(p_1))\vec{u}_0$.
5. $\quad$ **if** $|\vec{u}_0 \cdot \vec{n}(p_1)| < \epsilon$ **then**
   $\quad\quad$ **break**.
6. **return** p_1.

In practice, the value of β can be set to a small fixed positive number.

Figure 5.8 shows the results obtained using the 2-step edge localization method with α and $\beta = 0.1$ compared with the results obtained using the 1-step edge localization method based on Equation 5.4. The Gaussian differential masks with $\sigma = 1.0$ were used to generate the fields while bilinear interpolation was used to estimate the field values. In Figures 5.8-a and 5.8-c where $\vec{n} = \nabla(|\nabla L|)$ and $\vec{n} = L_{ww}\nabla L$, the differences between the results obtained using the 2 methods are not significant. However, Figure 5.8-b shows the effect of the inertial force presented in the 1-step method. Figure 5.9 illustrates boundaries extracted using the 2-step edge localization scheme at various σ with bilinear interpolation used to estimate the field values. Obviously, larger σ generates the smoother boundaries. In addition, near the corners, the extracted boundaries obtained using $\lambda = \nabla^2 L$ are slightly larger than the actual boundary while those obtained using $\lambda = L_{ww}$ are smaller than the actual boundary. These patterns are consistent with the shapes of boundary support areas shown in Figures 5.6-f and 5.7-f.

5.3 Practical Implementation

This section investigates the problems encountered in the implementation of boundary extraction algorithm including initialization, boundary extraction of connected re-

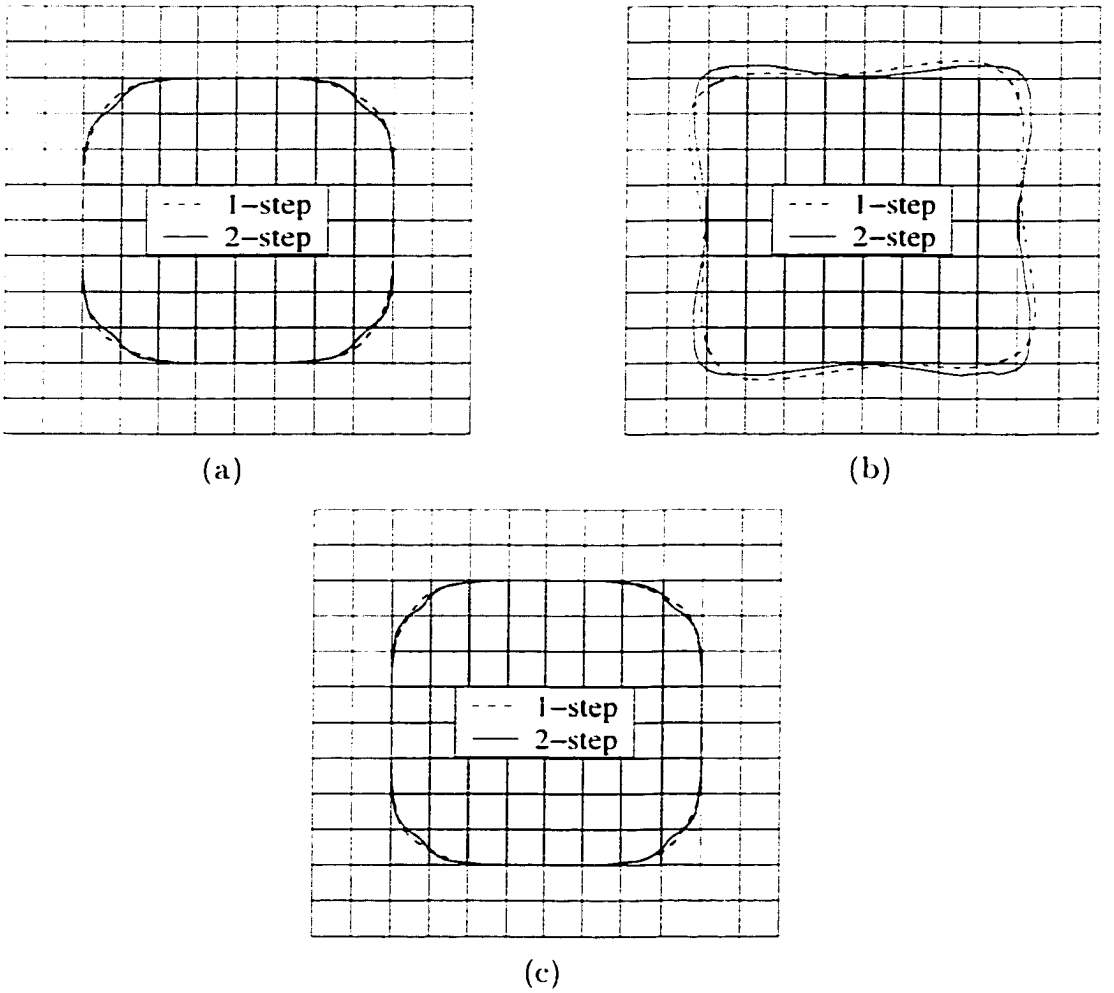
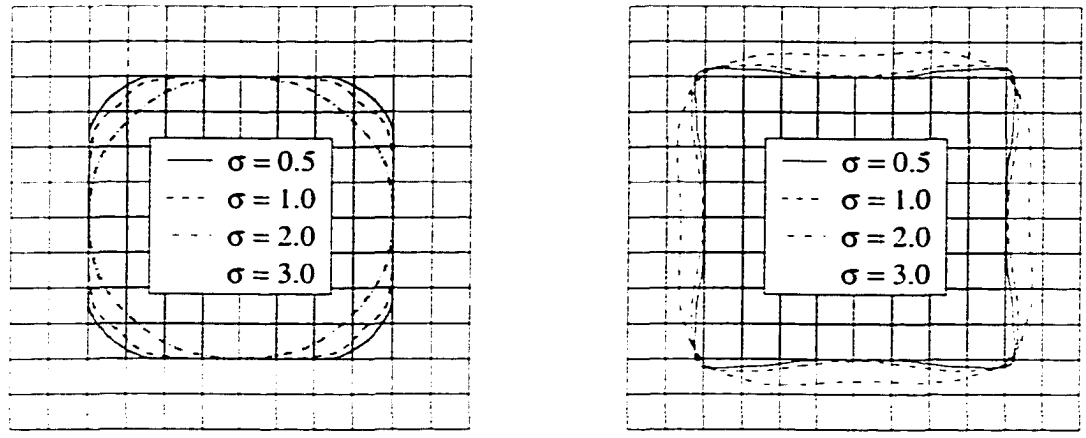


Figure 5.8 Extracted boundaries using 1-step and 2-step edge localization schemes with $\vec{t} = (\nabla L)^{\perp}$, $\alpha = 0.1$, $\beta = 0.1$ and (a) $\vec{n} = \nabla(|\nabla L|)$, (b) $\vec{n} = \nabla^2 L \nabla L$, (c) $\vec{n} = L_{ww} \nabla L$.



(a) (b)

Figure 5.9 Extracted boundaries using the 2-step edge localization scheme with $\vec{t} = (\nabla L)^{\perp}$, $\alpha = 0.1$, $\beta = 0.1$, and (a) $\vec{n} = L_{ww}\nabla L$, (b) $\vec{n} = \nabla^2 L \nabla L$.

gions, estimation of boundary points, and termination criteria. In 2 dimensions, boundaries are in general represented in the form of planar graphs. Data structures and procedures used in the construction of boundaries are discussed in Appendix B.

5.3.1 Initialization of Boundary Extraction Process

In order to allow the particle trajectory to converge to the true object boundary rapidly, the particle trajectory must start at the point close to the boundary. Since the boundaries are associated with regions of large gradient magnitude, potential candidates for starting points of the particle trajectories are local maxima of the gradient magnitude image $|\nabla L|$. However, since the local maximum points of the gradient magnitude image may not lie exactly on the boundaries, edge localization techniques must be used to optimize the starting points. Next, the different local maxima must be evaluated in terms of their significance.

In human vision, the most dominant edges are provided by high contrast objects. Therefore, in this research, a set of starting points for the boundary extraction process is obtained from local maxima of $|\nabla L|$ sorted in the descending order. For each starting point, the boundary extraction process is performed until the termination criterion is met. The process is then restarted at the next starting point until all starting points are used. Using local maxima of $|\nabla L|$ as starting points, the process can be automated. Figure 5.10 shows extracted boundaries of isolated objects. All fields were computed

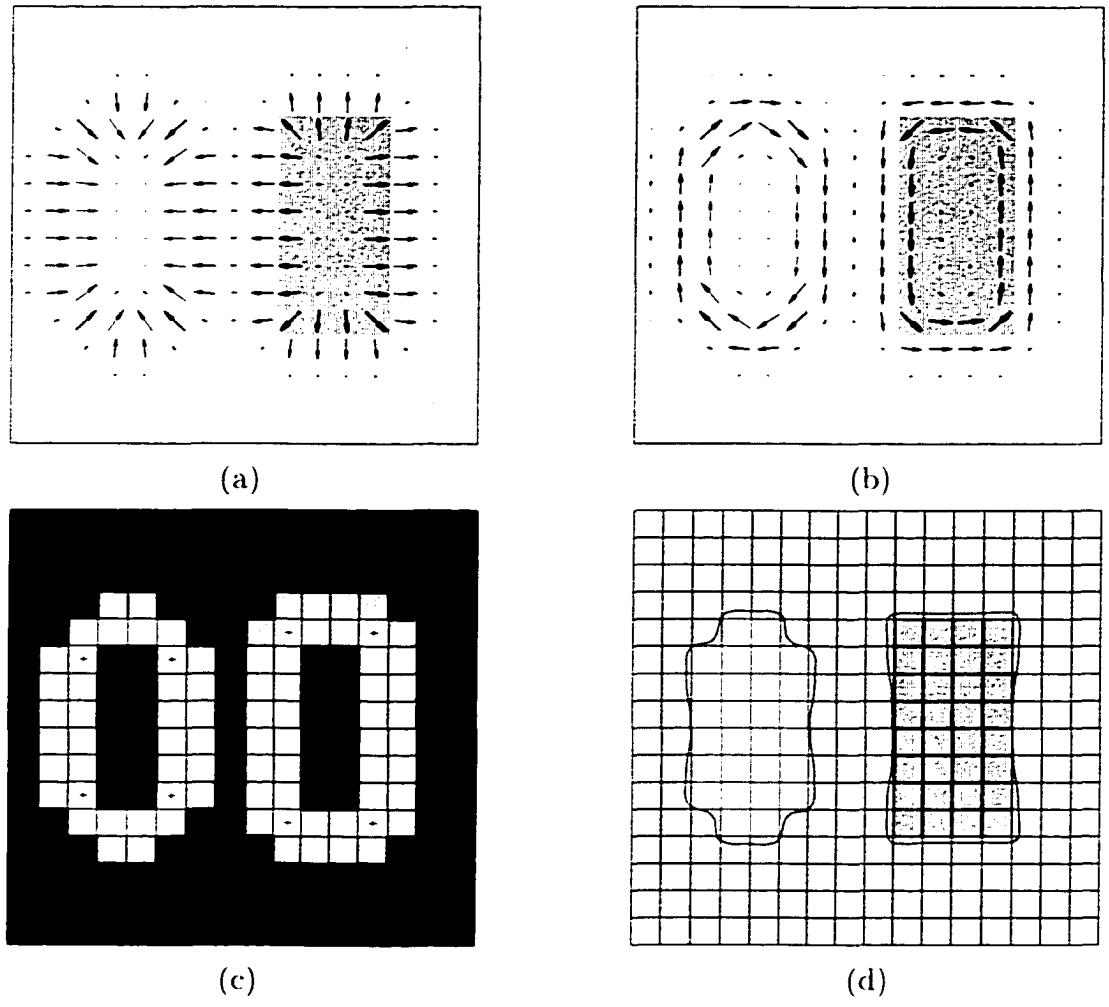


Figure 5.10 Example of an image containing isolated objects: (a) ∇L. (b) $(\nabla L)^{\perp}$. (c) $|\nabla L|$. (d) extracted boundaries.

using the Gaussian differential masks with $\sigma = 0.5$. As seen, the corresponding edge vector field forms complete loops encircling individual objects as shown in Figure 5.10-b. The clockwise direction of the loop indicates that the object intensity is higher than the background intensity while the counterclockwise direction of the loop indicates that the object intensity is lower than the background intensity. Figure 5.10-c shows $|\nabla L|$ with local maxima indicated by the "+" signs. The resultant boundaries obtained using the 2-step edge localization with $\alpha = 0.1$, $\beta = 0.05$, $\vec{t} = (\nabla L)^{\perp}$, and $\lambda = L_{ww}$ are shown in Figure 5.10-d.

5.3.2 Boundary Extraction of Connected Regions

In the case of connected regions as demonstrated in Figure 5.11-a where 3 different regions are labeled by the numbers 1, 2, and 3, the corresponding edge vector field does not form closed loops but rather a complicated "circuit" of edge vector currents as illustrated in Figure 5.11-b. Therefore, boundaries of connected regions can be modeled by planar graphs whose branches and nodes correspond to boundaries between pairs of connected regions and junctions respectively.

The proposed edge localization scheme is based on minimization of $|\lambda|$ along with the boundary where different signs of λ represent different regions. However, using sign of λ to locate boundaries is not valid when dealing with the junctions where 3 or more regions meet. This is because description of 3 or more connected regions needs the descriptor having number of states to be equal to the number of regions while the sign of λ has only 2 possible states, positive and negative. As seen in Figure 5.11-d, boundaries of Region 1 can be located using zero crossings of λ. However, boundaries between Region 2 and Region 3 at junctions A and B cannot be located using zero crossings of λ. Moreover, not all zero crossings of λ correspond to boundaries. For example, in Figure 5.11-d, there exists no boundary at point C where zero crossing of λ occurs. These false indications are filtered out using boundary support areas which will be described in Section 5.4. To handle the problem of extracting boundaries at junctions where sign(λ) is not adequate, edge localization by minimizing $|\lambda|$ must be relaxed by reducing the number of maximum iterations used in the second step of edge localization. This allows the particle to follow the tangential velocity field and rapidly merge with another boundary. Since edge localization is relaxed, the values of λ associated with the boundary points near the junctions are not expected to be close to zero.

Another problem of extracting boundaries of connected objects is illustrated in Figure 5.11-e. A particle that follows the tangential velocity field only in one direction does not completely extract the boundary of Region 2. This is because when the particle starts at Point D and travels along the boundary of Region 2 in the counterclockwise direction, at B, the trajectory shifts to the stronger edge vector currents in Region 1 and does not return back to complete the boundary of Region 2. In other words, the particle cannot escape to the weak edge vector currents once it is caught in the strong edge vector currents. This problem can be solved by allowing the particle to follow the tangential velocity field in forward direction by using positive α and backward direction by using negative α. This 2-way boundary extraction tracks weak edges completely in both forward and backward directions. The result of 2-way boundary extraction is

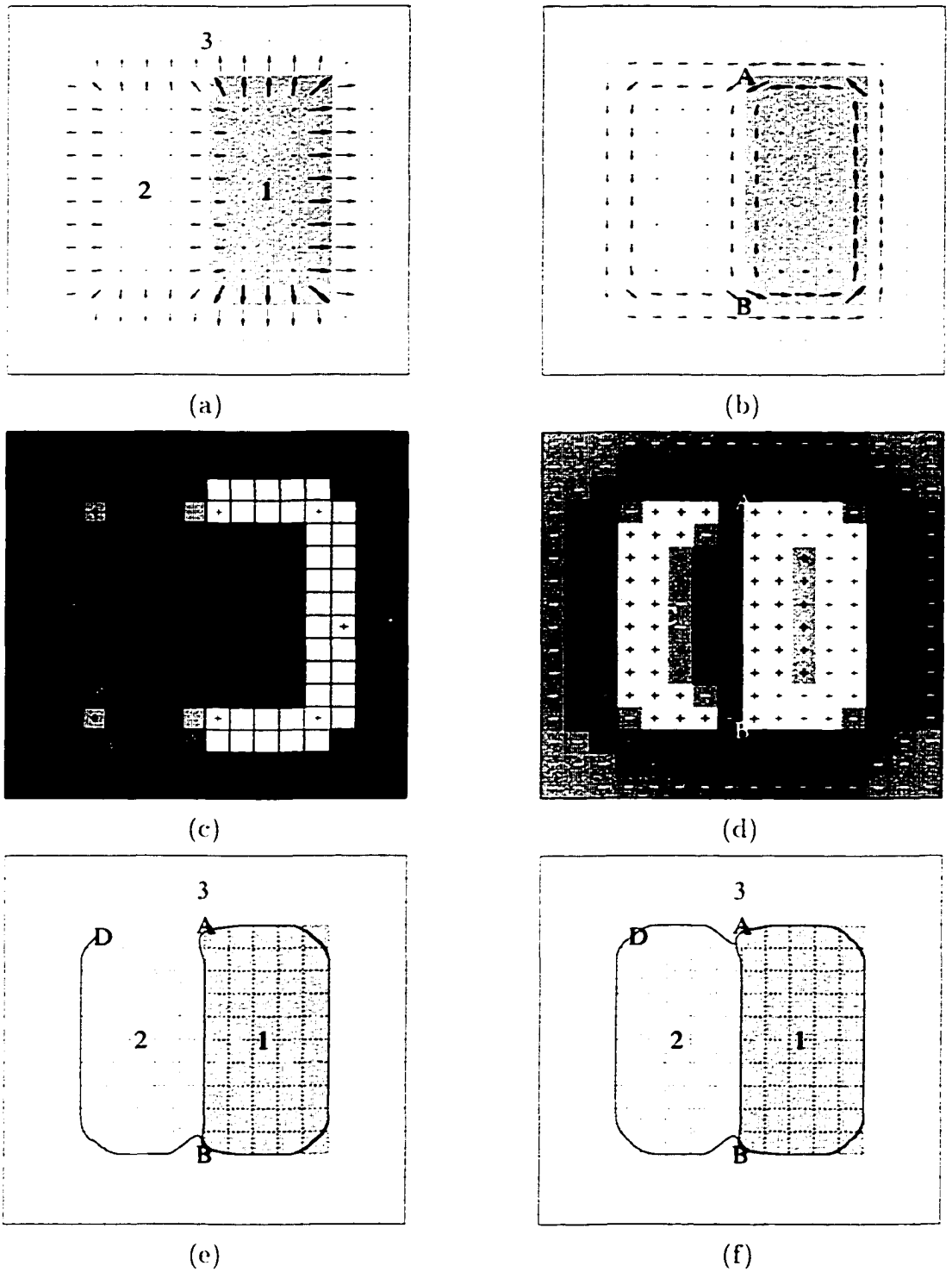


Figure 5.11 Boundary extraction of connected objects: (a) ∇L. (b) edge vector field $(\nabla L)^{\perp}$. (c) $|\nabla L|$. (d) L_{ww}. All fields are computed using Gaussian differential masks with $\sigma = 0.5$. Boundaries obtained using $\vec{t} = (\nabla L)^{\perp}$ and $\lambda = L_{ww}$: (e) 1-way. (f) 2-way.

shown in Figure 5.11-f.

5.3.3 Estimation of Boundary Points

Another problem encountered in boundary extraction is efficient representation of information using minimum memory storage while preserving all significant details. In the particle model, since all positions of the particle are recorded as boundary points, the method does not take into account the optimal locations of points along the boundary line. Consequently, if the step size used is too fine, extracted boundaries will contain redundant collinear boundary points. One way of reducing the number of points for boundary representation is to store only boundary points separated by a specified threshold distance:

$$|\vec{p}_{k+1} - \vec{p}_k| \geq \Delta T \tag{5.16}$$

where ΔT is the minimum allowed distance between the recorded boundary points. Nevertheless, this method does not truly utilize geometrical information of the image. Another approach is to store corners and junctions and decimate redundant collinear points.

Let $\vec{s}(u)$ be a parameterized curve with the arc length parameter u, the tangent vector $\vec{t}$ of s is defined as

$$\vec{t} = \frac{d\vec{s}(u)}{du}. \tag{5.17}$$

The curvature $\vec{k}$ of s is defined as

$$\vec{k} = \frac{d\vec{t}}{du} = \frac{d^2\vec{s}(u)}{du^2}. \tag{5.18}$$

Geometrically, local extrema of $\vec{k}$ are corners of the curve. However, in the boundary extraction problem, an explicit parametric equation of the boundary curve is unavailable. For corner detection, the curvature in Equation 5.18 can be replaced by

$$L_{vv} = \frac{L_x^2 L_{yy} - 2L_x L_y L_{xy} + L_y^2 L_{xx}}{L_x^2 + L_y^2}.$$

Local maxima of $|L_{vv}|$ along the curve and other features such as λ can detect corners and other suitable boundary points. In practice, in high curvature areas, the values of λ along extracted boundaries hardly approach zero. Therefore, corners can also be reflected by local maxima of $|\lambda|$ along the boundaries. Simplification of boundary representation can be done by maintaining only those boundary points that coincide with local extrema of the selected feature image.

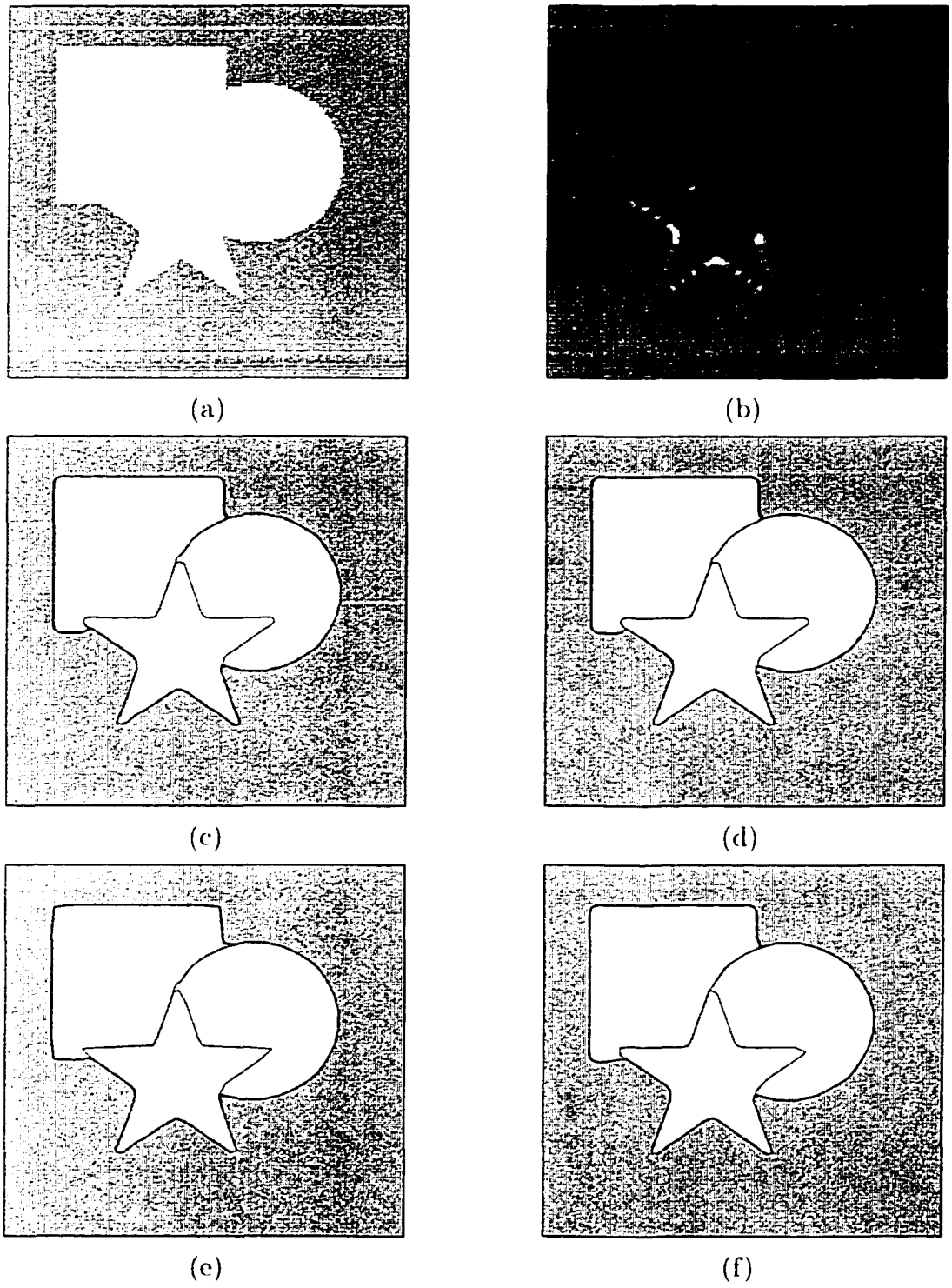


Figure 5.12 Results obtained using various boundary point estimation methods: (a) original image, (b) L_{vv}, (c) and (d) results obtained using $\Delta T = 0.5$ and 1.0 pixels, (e) and (f) results obtained using local extrema of L_{vv} and L_{ww} respectively.

Table 5.1 The numbers of boundary points of the results in Figure 5.12.

Figure	The number of boundary points
5.12-c	1406
5.12-d	971
5.12-e	668
5.12-d	993

Figure 5.12 demonstrates results obtained using various boundary point estimation methods. All fields were computed using the Gaussian differential masks with $\sigma = 1$. Figure 5.12-b displays the image of L_{vv} where dark and white regions reflect object corners. In Figures 5.12-c to 5.12-f, the 2-step edge localization method with $\vec{t} = (\nabla L)^{\perp}$, $\lambda = L_{ww}$, $\alpha = 0.1$, and $\beta = 0.1$ and bicubic B-spline polynomial interpolation were used to compute particle trajectories. Figures 5.12-c and 5.12-d show results obtained using $\Delta T = 0.5$ and 1.0 pixels. The numbers of points recorded are 1406 and 971. In contrast, only 668 and 993 boundary points are recorded in the results obtained using local extrema of L_{vv} and L_{ww} respectively as shown in Figures 5.12-e and 5.12-f. These results are summarized in Table 5.1.

5.3.4 Termination of Boundary Extraction

In boundary extraction, a suitable choice of termination criterion is essential in order to prevent the process from repeatedly tracking the same parts of boundaries or straying away from the object. In this research, 4 termination criteria are used:

1. When the particle reaches the starting point of the trajectory, boundary extraction must be terminated by closing the boundary loop. This condition can be detected by measuring the distance between the current position of the particle and the starting point.

2. When the particle reaches some previously extracted boundaries, the process must be terminated by creating a junction at the terminating point.

3. When the particle travels beyond borders of the image, boundary extraction must be terminated. This condition can be detected by comparing the coordinate of the particle with borders of the image.

4. In some rare cases, namely *local traps*, the particle can be trapped in a small loop formed in the tangential velocity field or locked in the areas where both $\vec{t}$ and $\vec{n}$ approach zero. Therefore, if the particle has not traveled beyond some small distance ϵ with respect to the previously recorded boundary point, after N iterations, the process must be terminated to avoid the local trap. The values of ϵ and N depend on the step size α. N can be set to be multiples of $\frac{\epsilon}{\alpha}$.

5.4 Choices of Fields

5.4.1 Boundary Localization Field (λ)

Since the 2-step edge localization method minimizes $|\lambda|$ along the extracted boundaries, zeros crossings of λ associated with boundaries roughly determine the shapes of objects. Candidate choices of λ are

$$L_{ww} = \frac{L_x^2 L_{xx} + 2L_x L_y L_{xy} + L_y^2 L_{yy}}{L_x^2 + L_y^2}$$

and

$$\nabla^2 L = L_{xx} + L_{yy}.$$

Figure 5.13 shows the results obtained using the 2-step edge localization method with $\vec{t} = (\nabla L)^{\perp}$, $\alpha = 0.1$, and different choices of λ. Bilinear interpolation was used to approximate the field value. All fields were computed using Gaussian differential masks with $\sigma = 1$. The boundary in Figure 5.13-a obtained using $\lambda = L_{ww}$ is smaller than the actual boundary while the boundary in Figure 5.13-b obtained using $\lambda = \nabla^2 L$ is larger than the actual boundary. Consequently, L_{ww} was combined with $\nabla^2 L$ to get

$$\lambda = \nabla^2 L + L_{ww}. \tag{5.19}$$

to obtain an improved result. Figure 5.13-c shows the corresponding result. Another example demonstrating the effect of λ is shown in Figure 5.14. All fields are computed using the Gaussian differential masks with $\sigma = 5$ and using $\vec{t} = (\nabla L)^{\perp}$ and $\alpha = 0.5$ with bilinear interpolation for surface approximation. As in the previous example, the best result shown in Figure 5.14-c was obtained using $\lambda = \nabla^2 L + L_{ww}$.

5.4.2 Normal Velocity Field ($\vec{\gamma}$)

From Section 5.2, two critical components, the tangential and normal compressive velocity fields, $\vec{t}$ and $\vec{n}$, are derived from

$$\vec{t} = (\vec{\gamma})^{\perp}$$

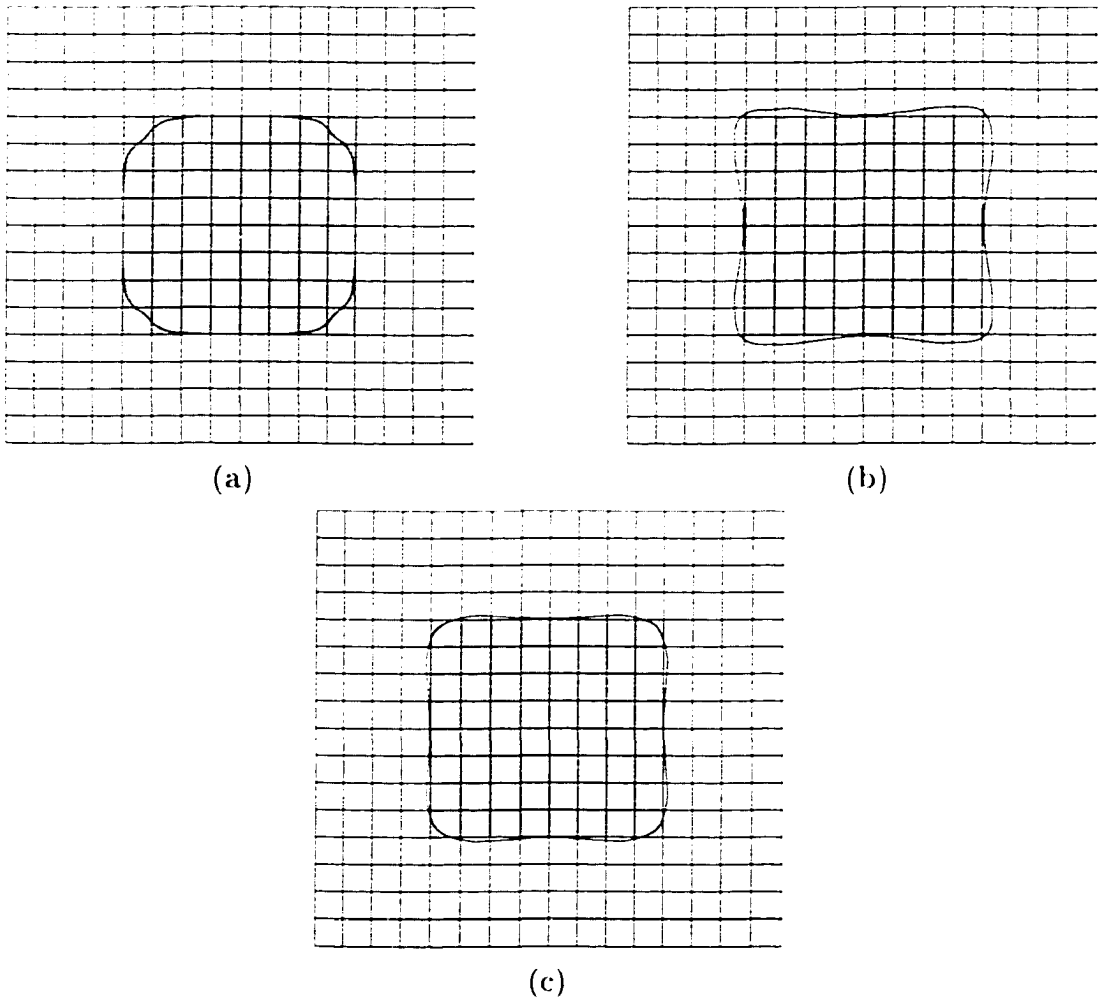


Figure 5.13 Results obtained using $\vec{t} = (\nabla L)^{\perp}$, $\alpha = 0.1$, and (a) $\lambda = L_{ww}$, (b) $\lambda = \nabla^2 L$, (c) $\lambda = \nabla^2 L + L_{ww}$.

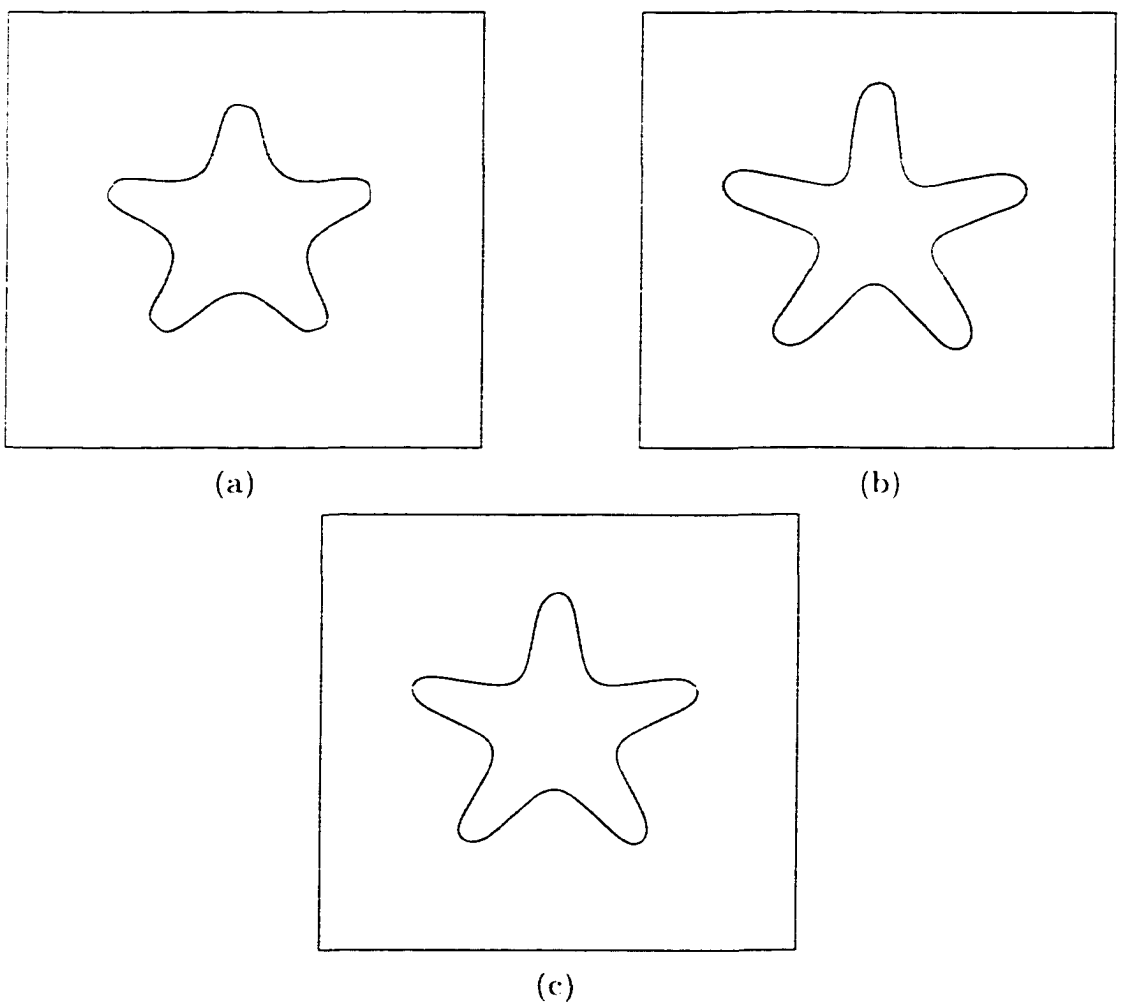


Figure 5.14 Results obtained using $\vec{t} = (\nabla L)^{\perp}$, $\alpha = 0.5$, and (a) $\lambda = L_{ww}$, (b) $\lambda = \nabla^2 L$, (c) $\lambda = \nabla^2 L + L_{ww}$.

and

$$\vec{n} = \lambda \cdot \vec{\gamma}.$$

In general, the principal choice of $\vec{\gamma}$ is ∇L. However, as demonstrated by Figure 5.15-a, $\vec{\gamma} = \nabla L$ is not effective when the fields are generated from the multiple scale differential masks. Besides ∇L, $-\nabla\lambda$ can be used as an alternative choice for $\vec{\gamma}$. Figure 5.15-c shows the improved result obtained using $\vec{\gamma} = -\nabla\lambda$. Compared with the method using $\lambda = \nabla L$, the method using $\vec{\gamma} = -\nabla\lambda$ yields better edge localization performance in the areas of high curvature.

Nevertheless, in the case of connected objects, $\vec{\gamma} = -\nabla\lambda$ also results in some false boundaries as demonstrated in Figure 5.15-d. Combining the advantages $\vec{\gamma} = \nabla L$ to track junctions and $\vec{\gamma} = -\nabla\lambda$ to track high curvature regions, the generalized expression of the normal velocity field γ is proposed as

$$\vec{\gamma} = o_1(\mu)\nabla L - o_2(\mu)\nabla\lambda \tag{5.20}$$

where $\mu = \frac{\nabla L}{|\nabla L|} \cdot \frac{\nabla\lambda}{|\nabla\lambda|}$, o_1 is an arbitrary function of μ, and o_2 is a decreasing function of μ. One choice of $\vec{\gamma}$ is given by

$$\vec{\gamma} = \nabla L - \frac{1-\mu}{2}\nabla\lambda. \tag{5.21}$$

Figure 5.16 displays the results obtained using $\vec{\gamma}$ in Equation 5.21. Boundary support areas are shown by the dark pixels in Figures 5.16-c and 5.16-d while extracted boundaries are shown in Figures 5.16-e and 5.16-f. Obviously, results obtained using $\vec{\gamma} = \nabla L - \frac{1-\mu}{2}\nabla\lambda$ are superior to those obtained using $\vec{\gamma} = \nabla L$ and $\vec{\gamma} = \nabla\lambda$ in handling object junctions and tracking boundaries in the high curvature areas.

5.5 Experimental Results

From the previous section, the choice of differential masks, surface approximation, fields, edge localization method, and boundary representation used in this experiment are summarized in Table 5.2. The values of scale parameters depend on sizes of structures of interest. Selection of a scale parameter σ is applied in the generation of fields.

5.5.1 Boundary Extraction of Multiscale Objects

Multiscale structures contain structures at different scales that can be observed by means of using multiscale filtering techniques. Boundary extraction of structures at each

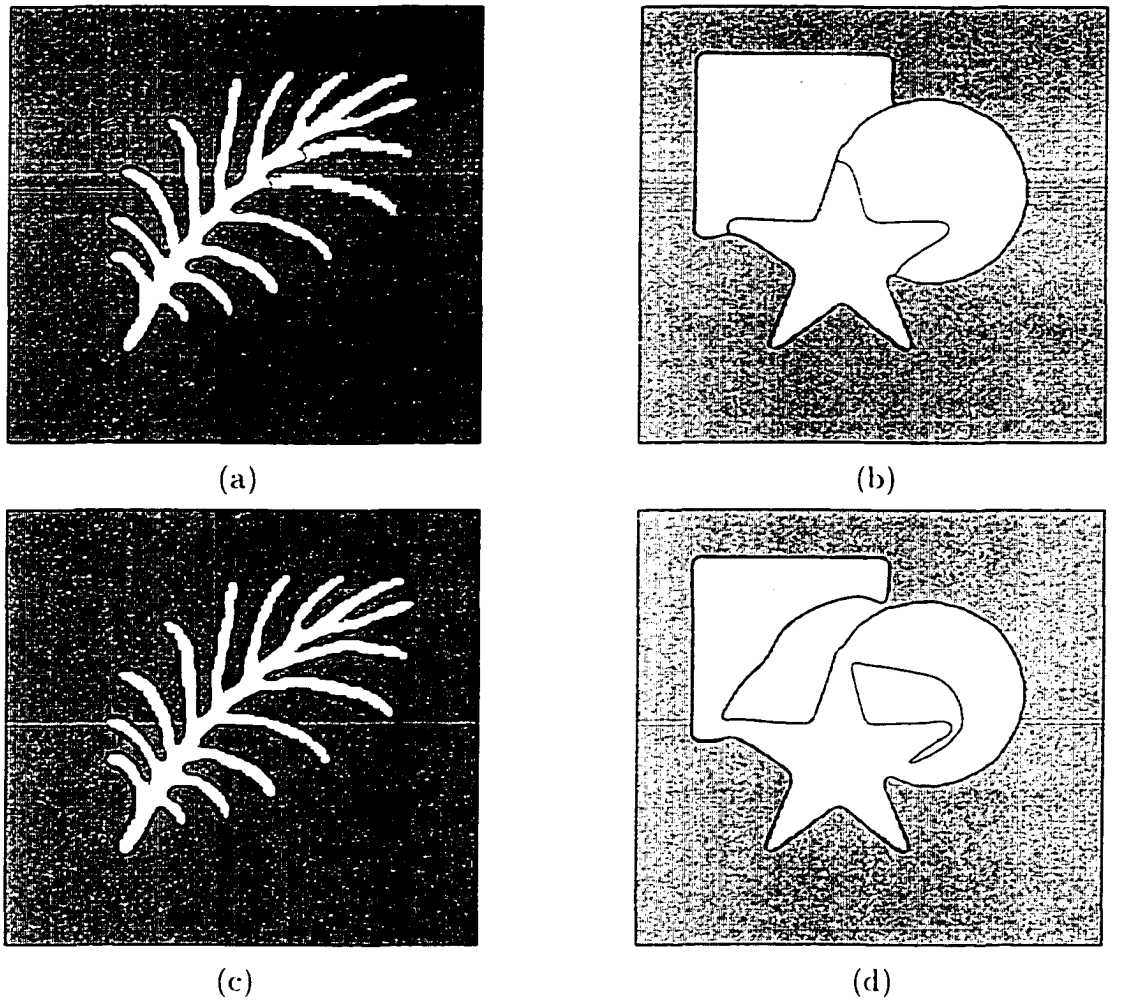


Figure 5.15 Boundary extraction results obtained using $\alpha = 0.1$. $\lambda = L_{ww} + \nabla^2 L$: (a) and (b) $\vec{\gamma} = \nabla L$. (c) and (d) $\vec{\gamma} = -\nabla\lambda$. All fields are computed using multiple scale Gaussian differential masks with σ varying from 1 to 8.

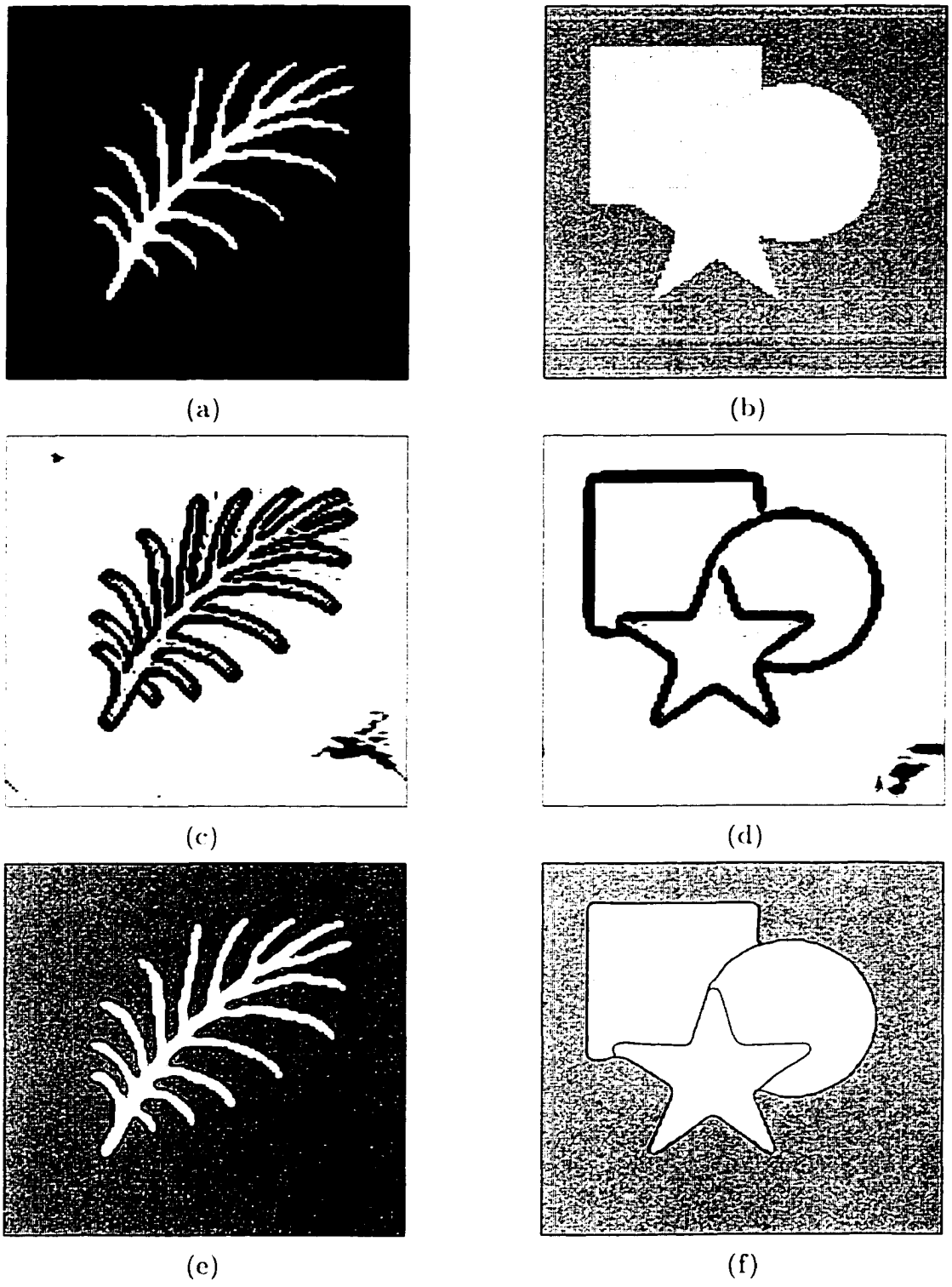


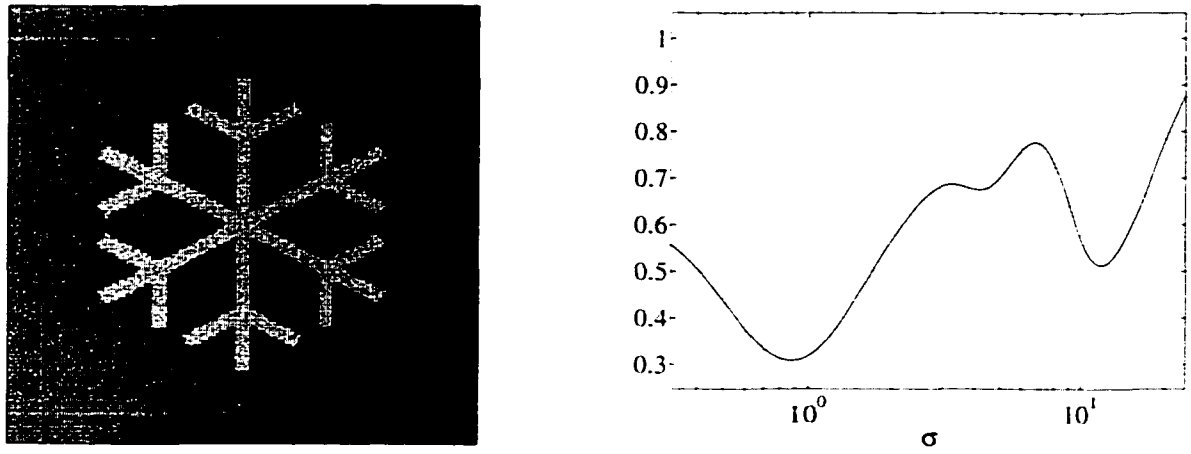
Figure 5.16 (a)-(b) Original images. (c)-(d) $\mu = \frac{\nabla L}{|\nabla L|} \cdot \frac{\nabla \lambda}{|\nabla \lambda|}$. (e)-(f) extracted boundaries obtained using $\vec{\gamma} = \nabla L - \frac{1-\mu}{2} \nabla \lambda$.

Table 5.2 Choices of differential masks, surface approximation method, fields, edge localization method, and boundary representation used in the experiment.

Differential masks	Multiscale Gaussian differential masks and multiple scale Gaussian differential masks
Surface approximation method	Bilinear interpolation
Boundary localization field	$\lambda = L_{ww} + \nabla^2 L$
Normal velocity field	$\vec{\gamma} = \nabla L - \frac{1-\mu}{2}\nabla\lambda$ where $\mu = \frac{\nabla L}{\lvert\nabla L\rvert} \cdot \frac{\nabla\lambda}{\lvert\nabla\lambda\rvert}$
Normal compressive velocity field	$\vec{n} = \lambda \cdot \vec{\gamma}$
Tangential velocity field	$\vec{t} = (\vec{\gamma})^{\perp}$
Edge localization	2-Step approach with $\alpha = 0.1$
Boundary representation	Doubly-connected edge lists
Minimum allowed distance ΔT for marking boundary points	1 pixel

scale can be performed using multiscale differential masks of the corresponding size. The suitable choice of scale parameters depends on the size of structures of interest. When such information is unavailable, selection of scales can be done using global scale analysis or local scale analysis described in previous chapters. Global scale analysis also provides information of existing scales that can be used to separate structures in an image at different scales. Local minima of the DPSNVR graph correspond to scales existing in the underlying image. Multiscale boundary extraction are performed by extracting boundaries of objects at each individual scale separately. Next, several examples of multiscale boundary extraction utilizing scale information obtained from the DPSNVR graphs are presented.

The first example in Figure 5.17 shows a simulated image of a multiscale snowflake. The DPSNVR graph in Figure 5.17-b shows 3 local minima at $\sigma = 0.88$, 4.07, and 11.75 corresponding to 3 existing scales. Multiscale boundary extraction of structures at these 3 scales are shown in Figure 5.18 where details at each scale are clearly seen. The examples in Figures 5.19 and 5.20 demonstrate multiscale boundary extraction of multiscale cluster-like objects, respectively, in the absence and presence of noise. The image in Figure 5.20-a is the degraded version of the image in Figure 5.19-a corrupted by additive white Gaussian noise with signal to noise ratio 2. All 3 existing scales of these two images are detected by local minima of the corresponding DPSNVR graphs in Figures 5.19-b and 5.20-b. In the absence of noise, the results displayed in Figure



(a) (b)

Figure 5.17 Image of a multiscale object: (a) original image, (b) DPSNVR graph.

5.21 clearly show the boundaries of the objects at individual scales. Furthermore, the extracted boundaries at large scales shown in Figure 5.22 are unaffected by noise.

The next example in Figure 5.23 shows a real image of multiscale object. The image consists of 3 scale structures as reflected by 3 local minima in the DPSNVR graph in Figure 5.23-b. Boundary extraction results obtained using the scale parameters selected from the local minima of the DPSNVR graph are shown in Figure 5.24. As in the three previous examples, details at each scale are precisely obtained.

5.5.2 Boundary Extraction of Real Images

In general, natural images do not always contain multiscale structures. Selection of a suitable scale can be difficult. Moreover, natural images usually contain fuzzy objects with different degrees of blurring rendering the use of single-scale differential masks ineffective. In these cases, multiple scale differential masks can be employed to handle edges with different degrees of blurring and to simplify the scale selection process where a rough range of σ can be specified instead of using a single value of σ as in the case of a single scale differential mask. Several examples of real images and the results of the proposed boundary extraction are given next. The analysis of aerial, facial, and medical images are presented. In Figures 5.25 and 5.27, the multiple scale Gaussian differential masks with σ varying from 0.5 to 2.0 were used. For the images containing edges with different degrees of blurring shown in Figures 5.29 and 5.31, the multiple scale Gaussian differential masks with σ varying from 0.5 to 4, were used. The boundary

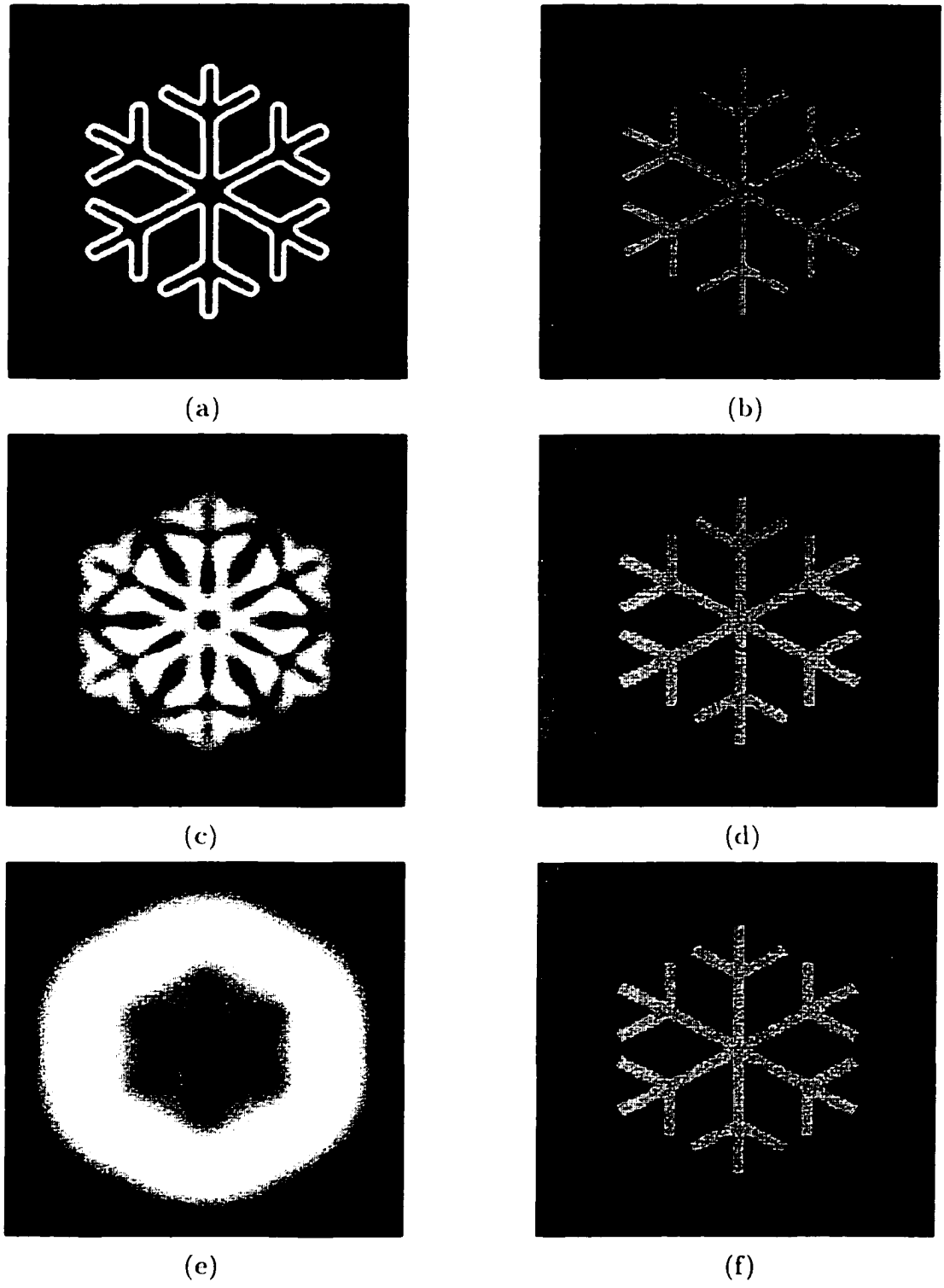
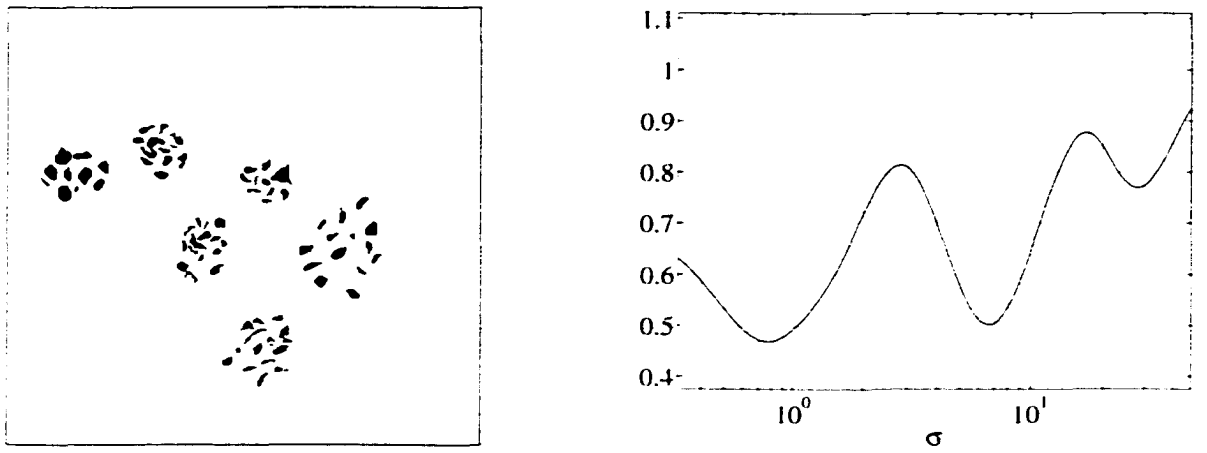
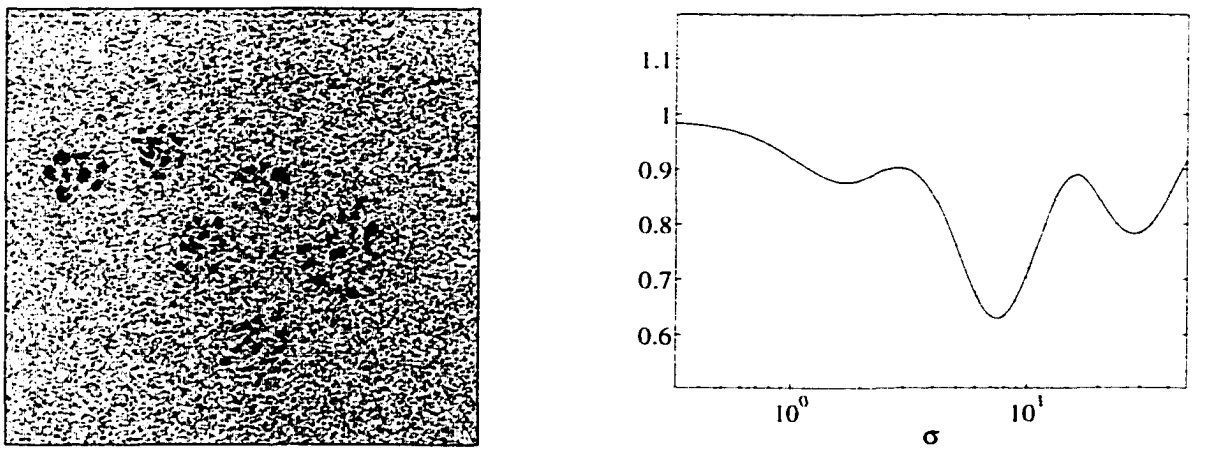


Figure 5.18 Gradient images and extracted boundaries of the image in Figure 5.17-a obtained using: (a) and (b) $\sigma = 0.88$, (c) and (d) $\sigma = 4.07$, (e) and (f) $\sigma = 11.75$.



(a) (b)

Figure 5.19 Image of multiscale objects: (a) original image. (b) DPSNVR graph.



(a) (b)

Figure 5.20 Degraded image with SNR = 2 of the image in Figure 5.19-a: (a) original image. (b) DPSNVR graph.

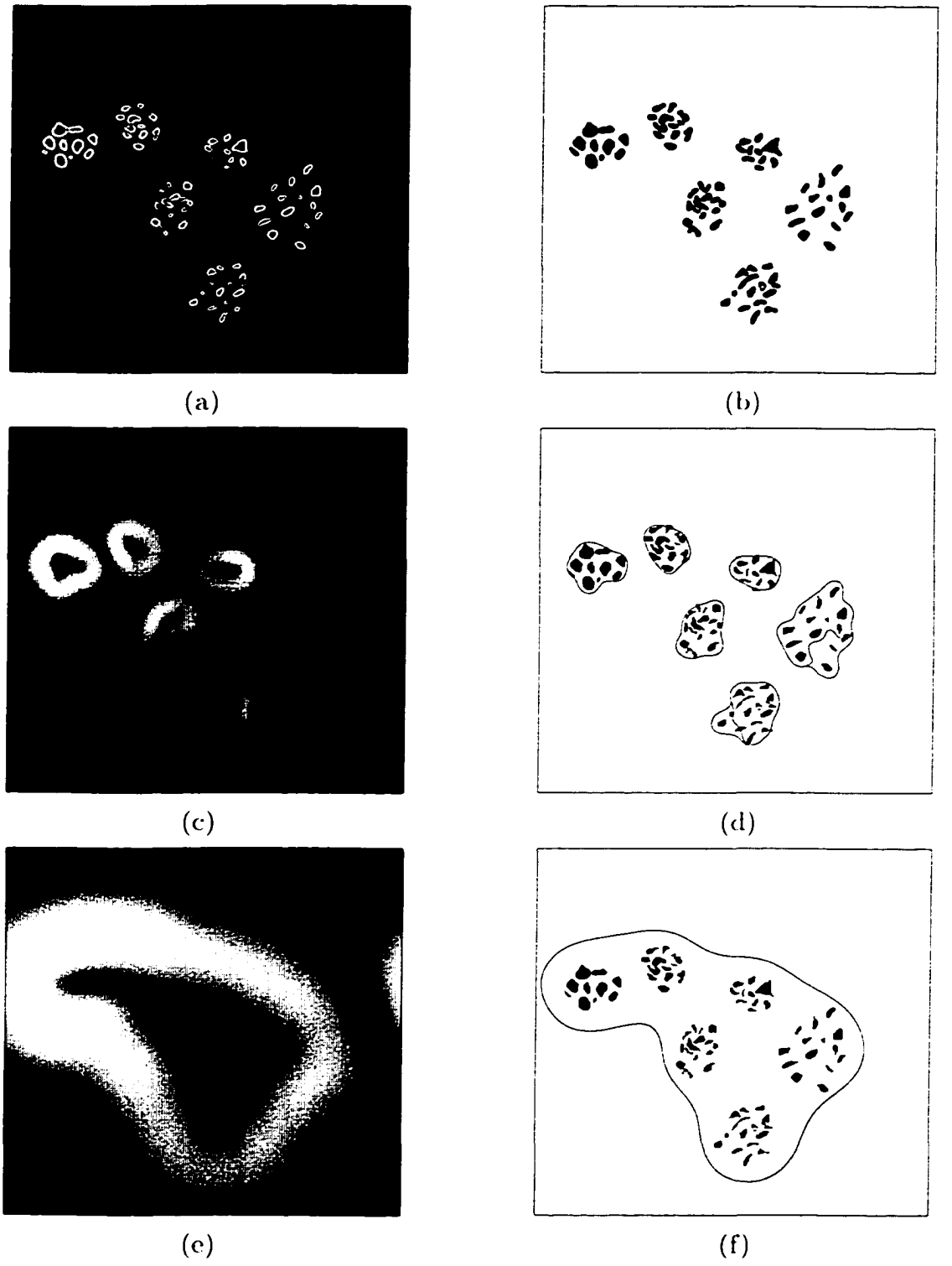


Figure 5.21 Gradient images and extracted boundaries of the image in Figure 5.19-a obtained using: (a) and (b) $\sigma = 0.78$. (c) and (d) $\sigma = 6.52$. (e) and (f) $\sigma = 27.84$.

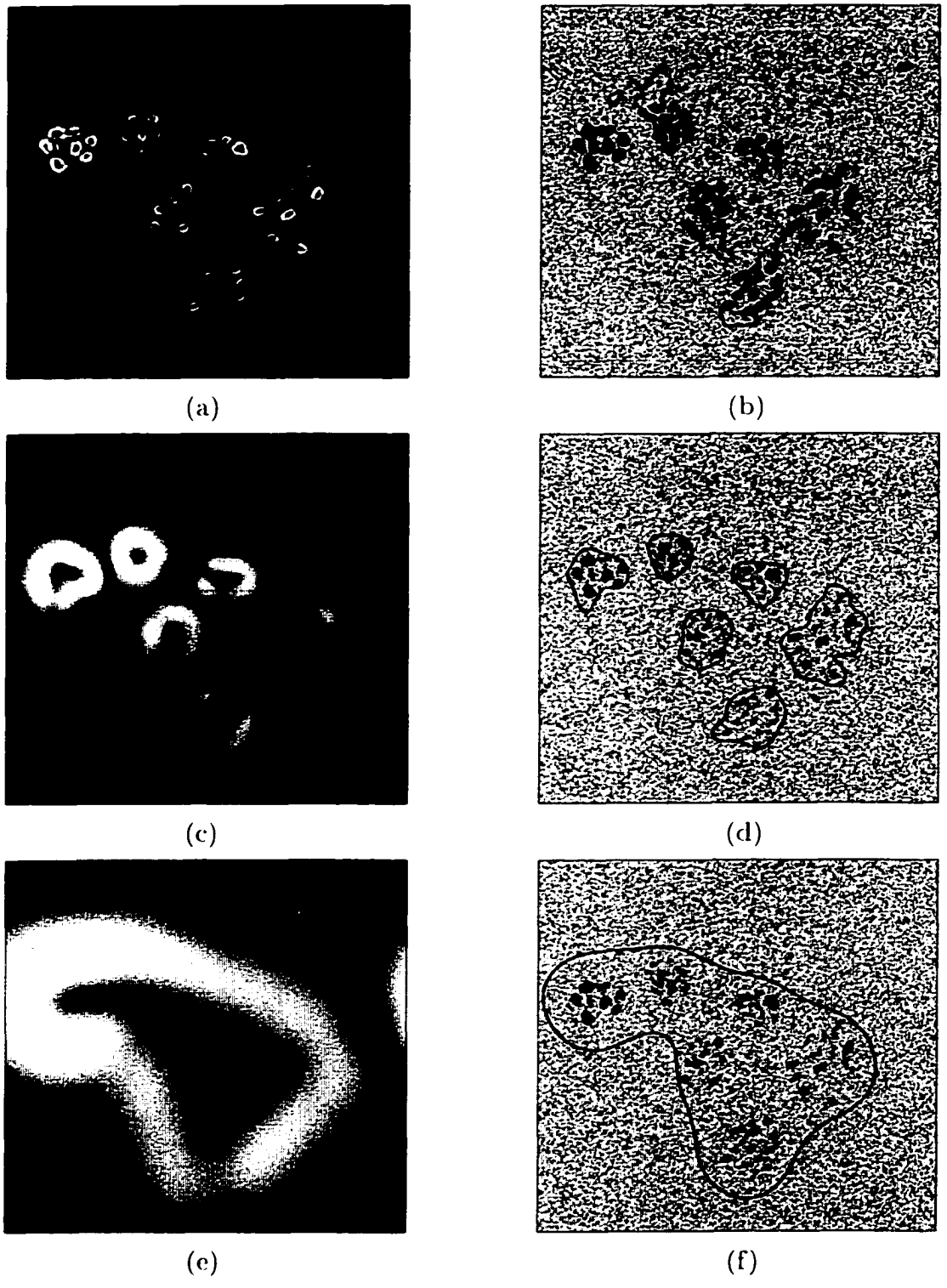
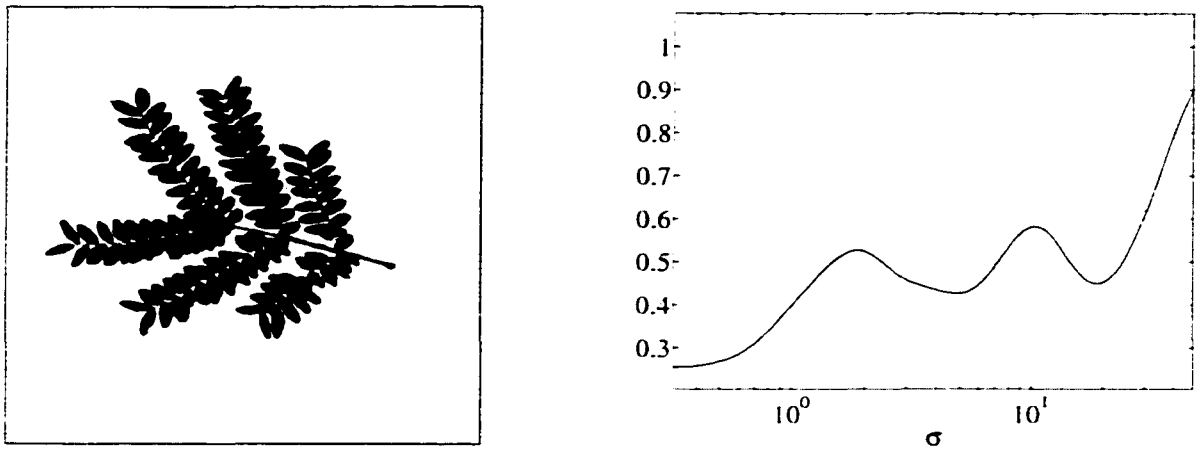


Figure 5.22 Gradient images and extracted boundaries of the image in Figure 5.20-a obtained using: (a) and (b) $\sigma = 1.72$, (c) and (d) $\sigma = 7.34$, (e) and (f) $\sigma = 27.84$.



(a) (b)

Figure 5.23 Multiscale objects: (a) original image. (b) DPSNVR graph.

point estimation method based on the minimum allowed distance was used for storing the results in Figures 5.30 and 5.32 while the method based on local extrema of $|L_{vv}|$ was used in Figures 5.26 and 5.28. In these examples, most of boundaries were successfully extracted while very few incomplete boundaries exist. In Figure 5.30, sharp edges and blurred edges are captured with subpixel accuracy. The algorithm has no difficulty in handling junctions and different topologies of boundaries of multiple connected objects.

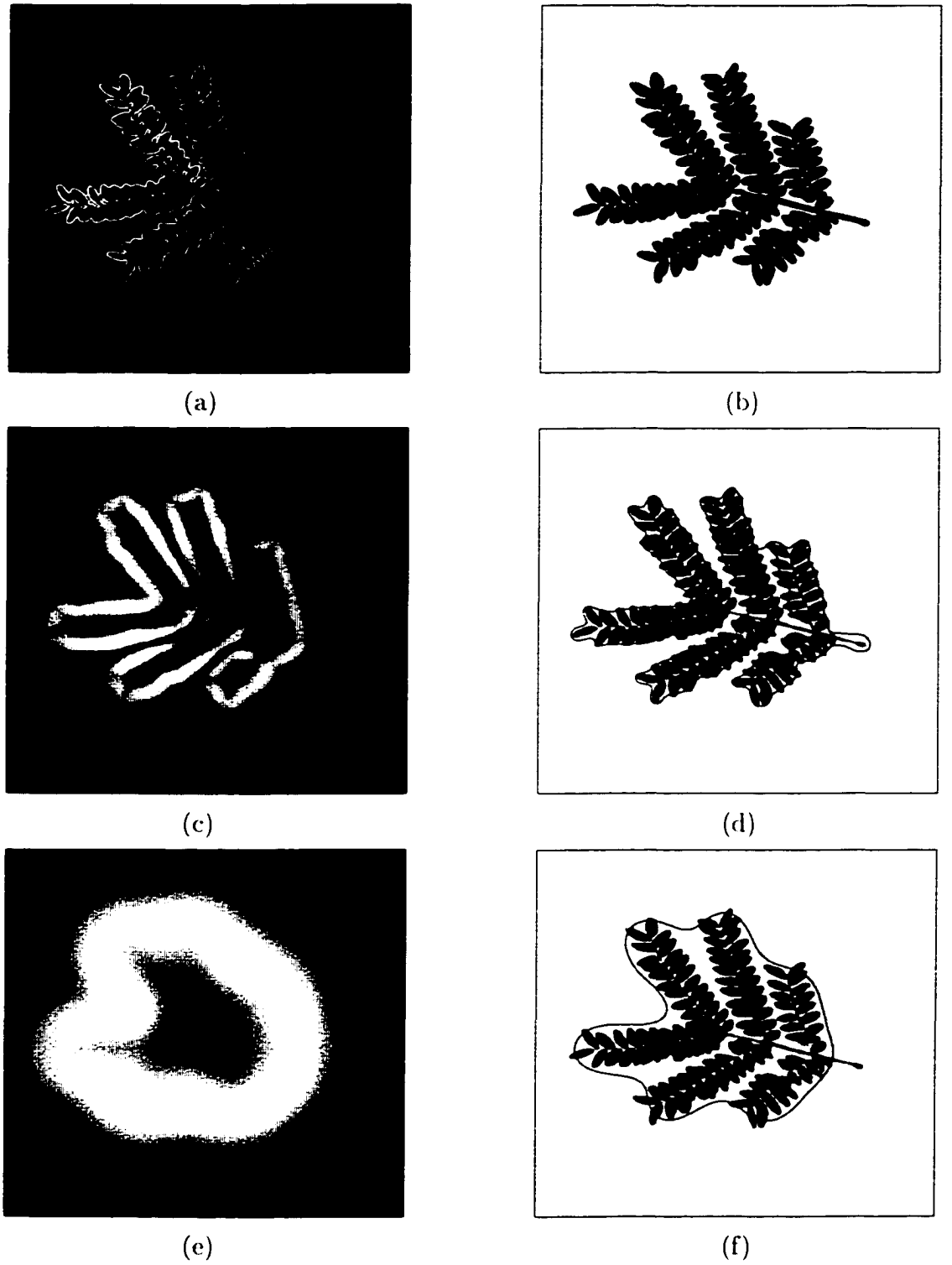


Figure 5.24 Gradient images and extracted boundaries of the image in Figure 5.23-a obtained using: (a) and (b) $\sigma = 0.5$, (c) and (d) $\sigma = 4.77$. (e) and (f) $\sigma = 18.81$.

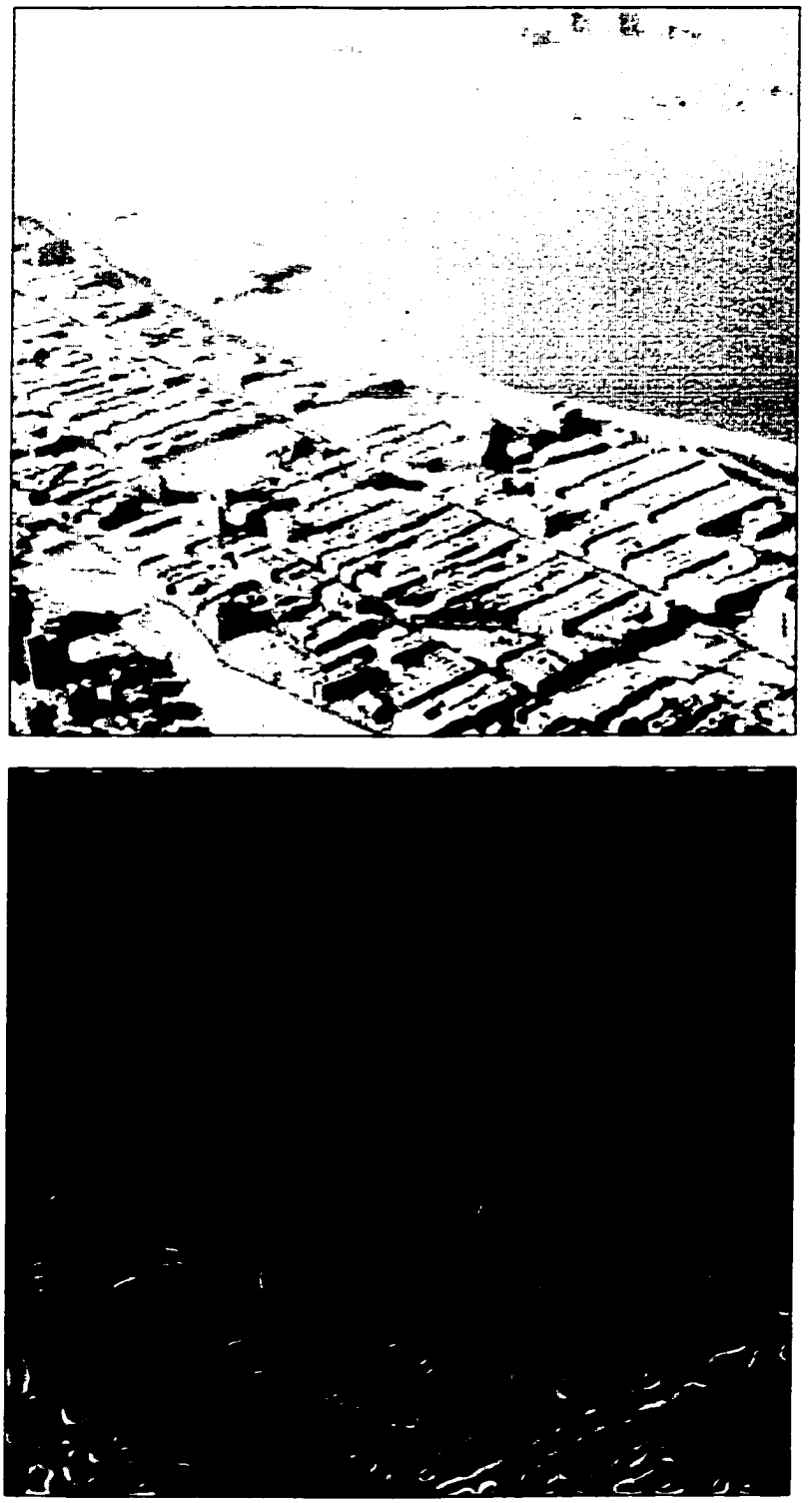


Figure 5.25 Original image (above) and gradient image (below).

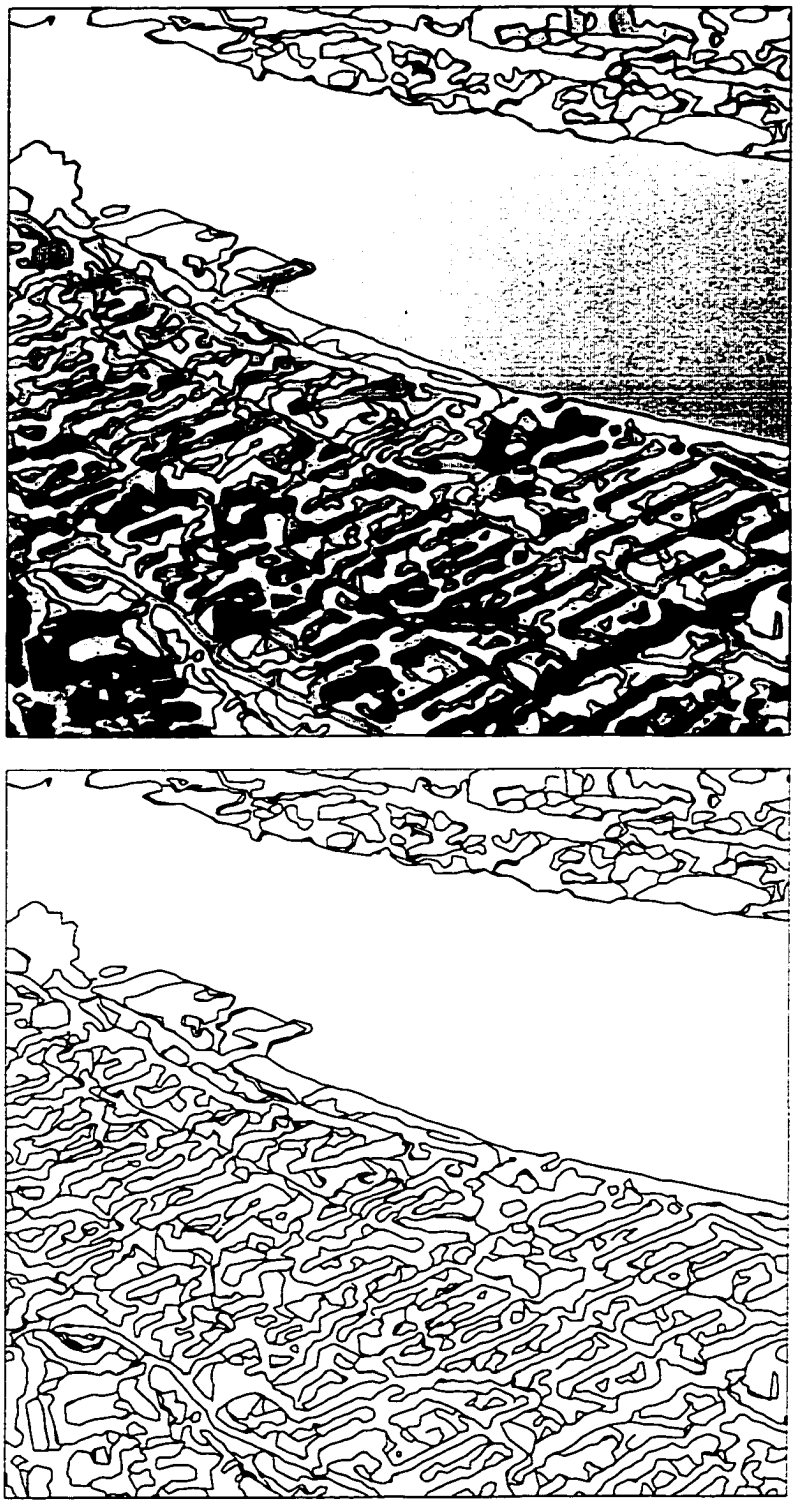


Figure 5.26 Extracted boundaries.

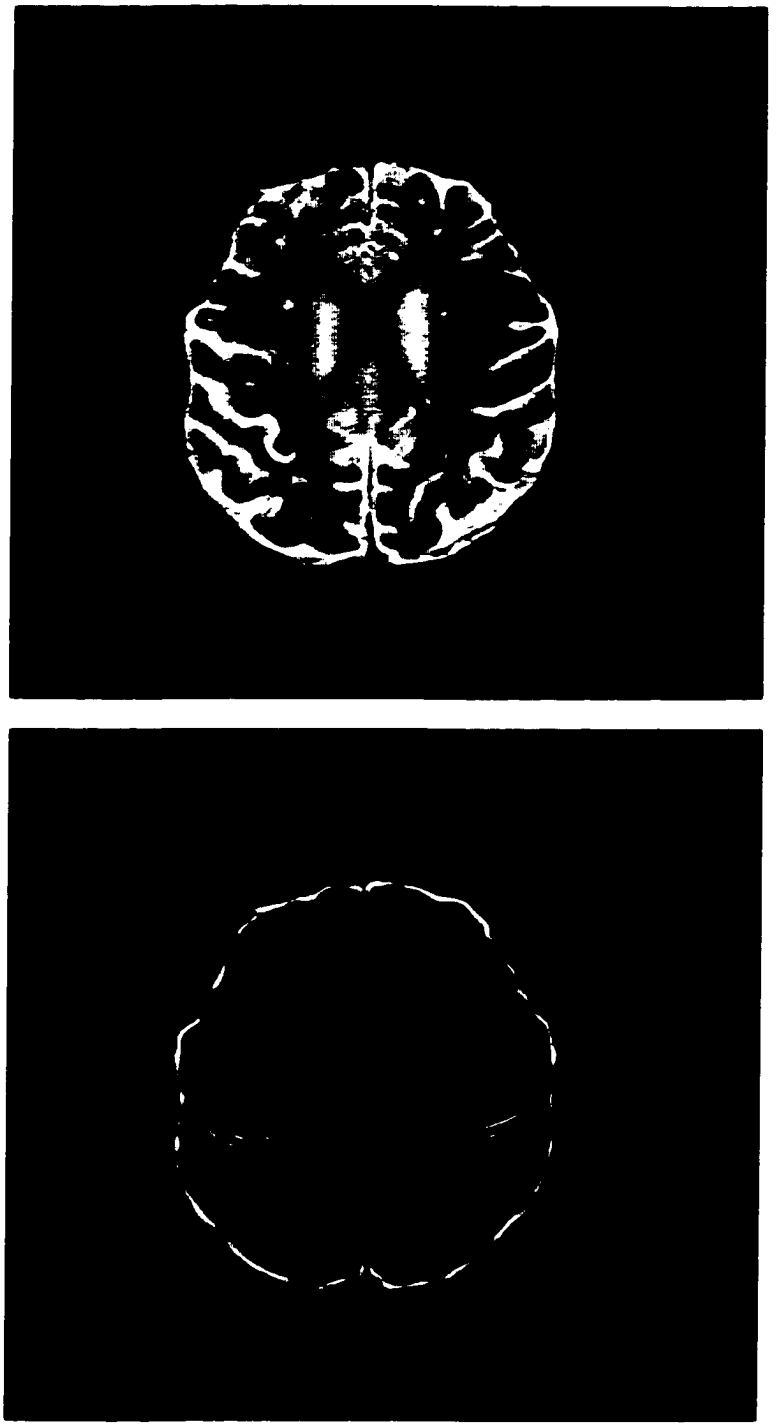


Figure 5.27 MRI image (above) and gradient image (below). (From the Whole Brain Atlas, http://www.med.harvard.edu/AANLIB /home.html. Courtesy of Keith A. Johnson, M.D., and J. Alex Becker.)

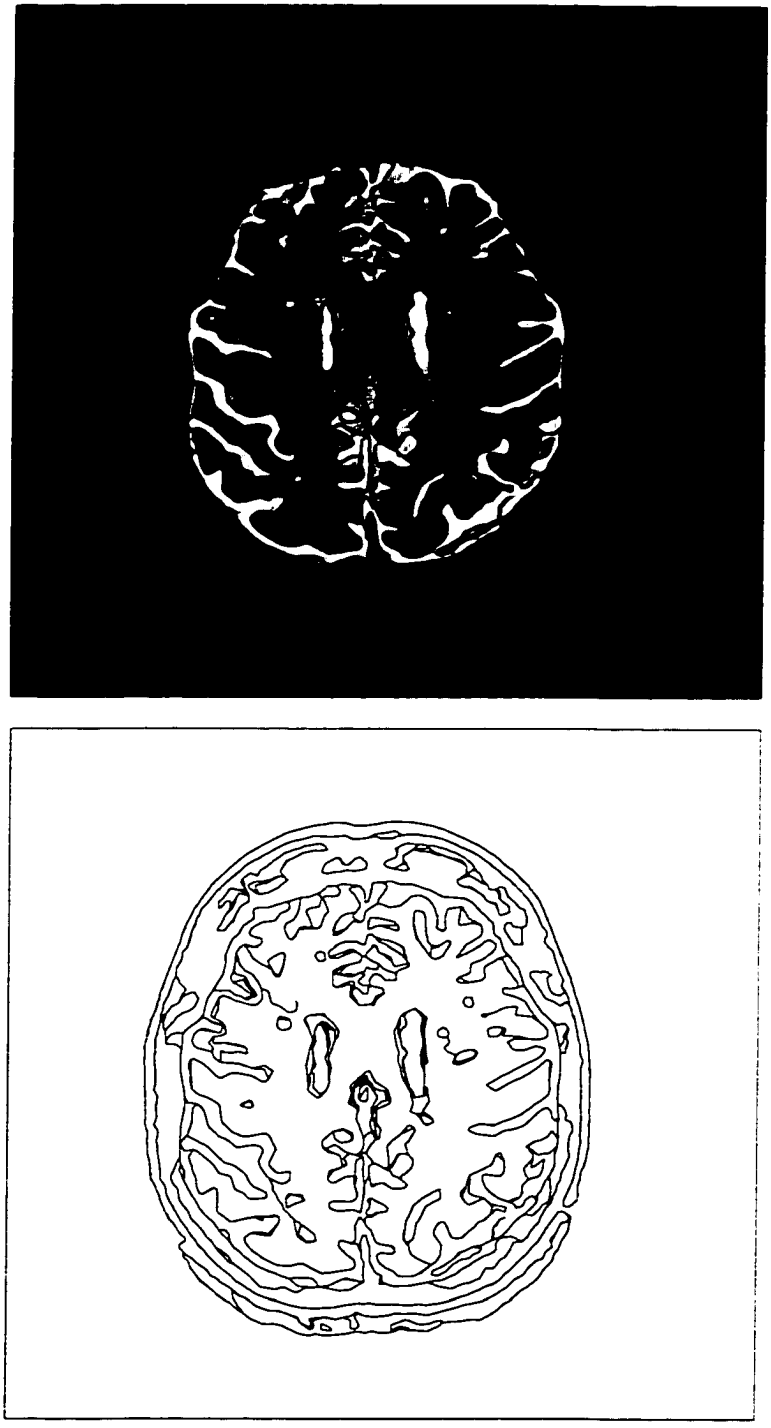


Figure 5.28 Extracted boundaries.

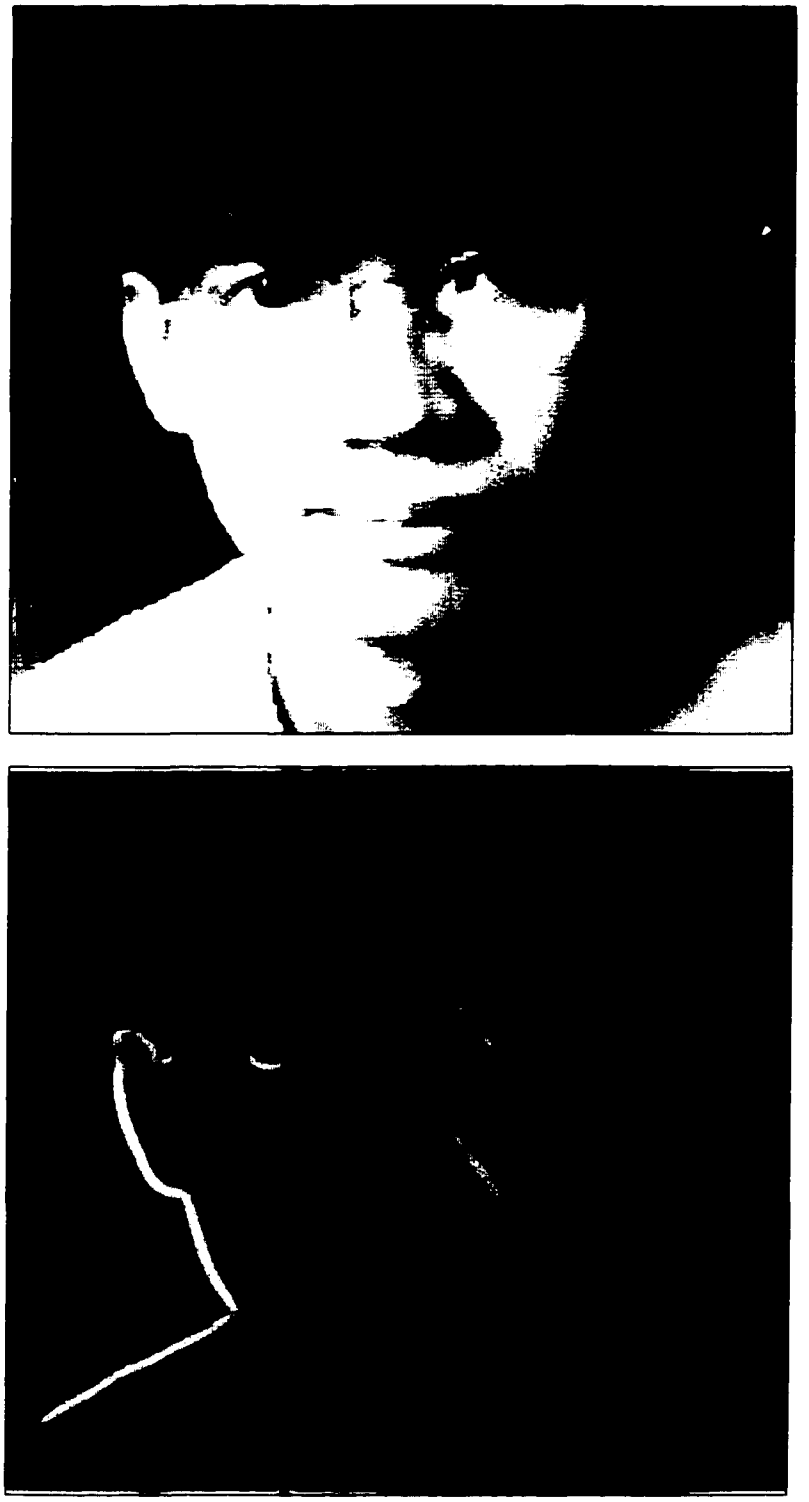


Figure 5.29 Original image (above) and gradient image (below).

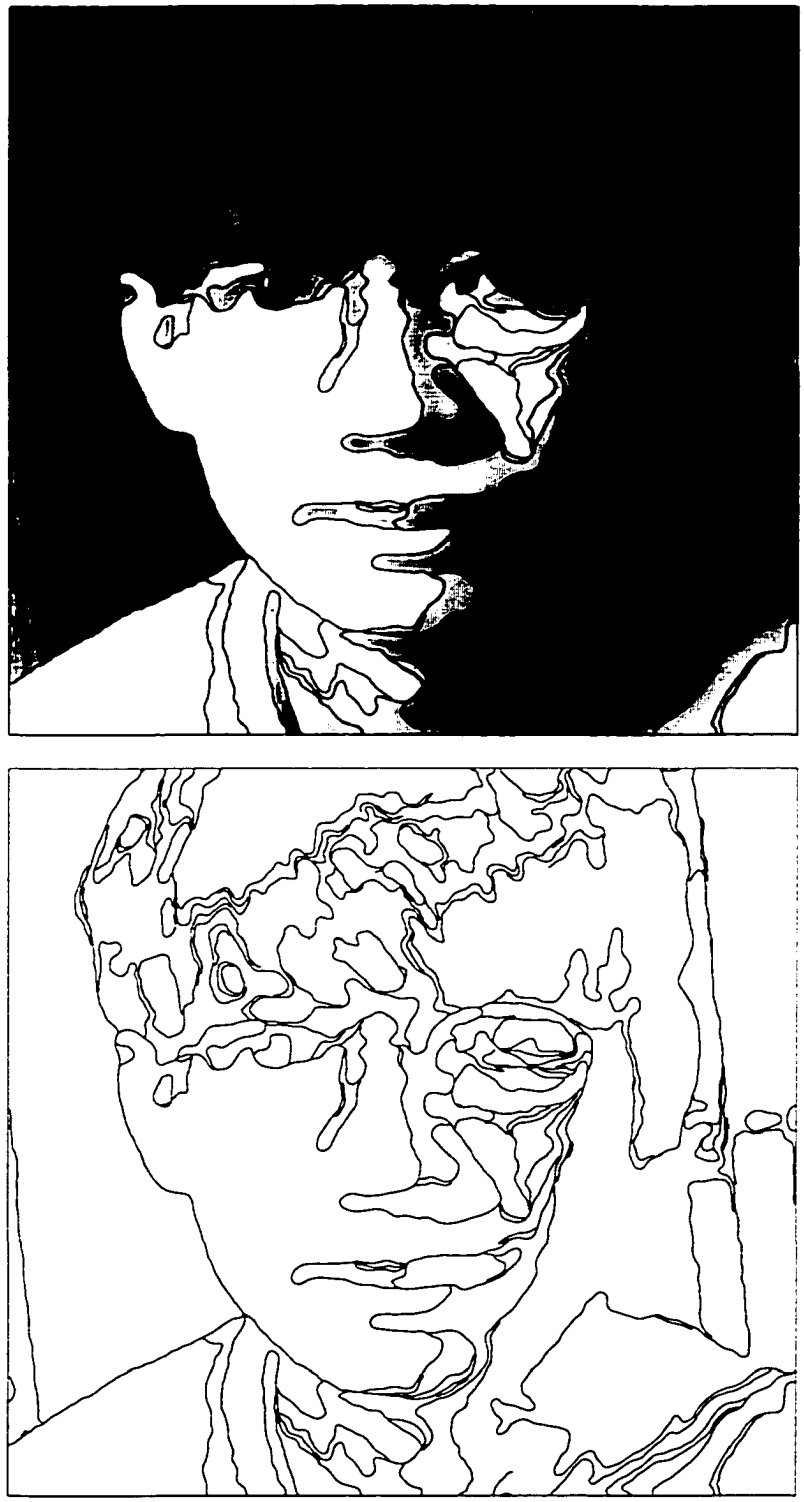


Figure 5.30 Extracted boundaries.

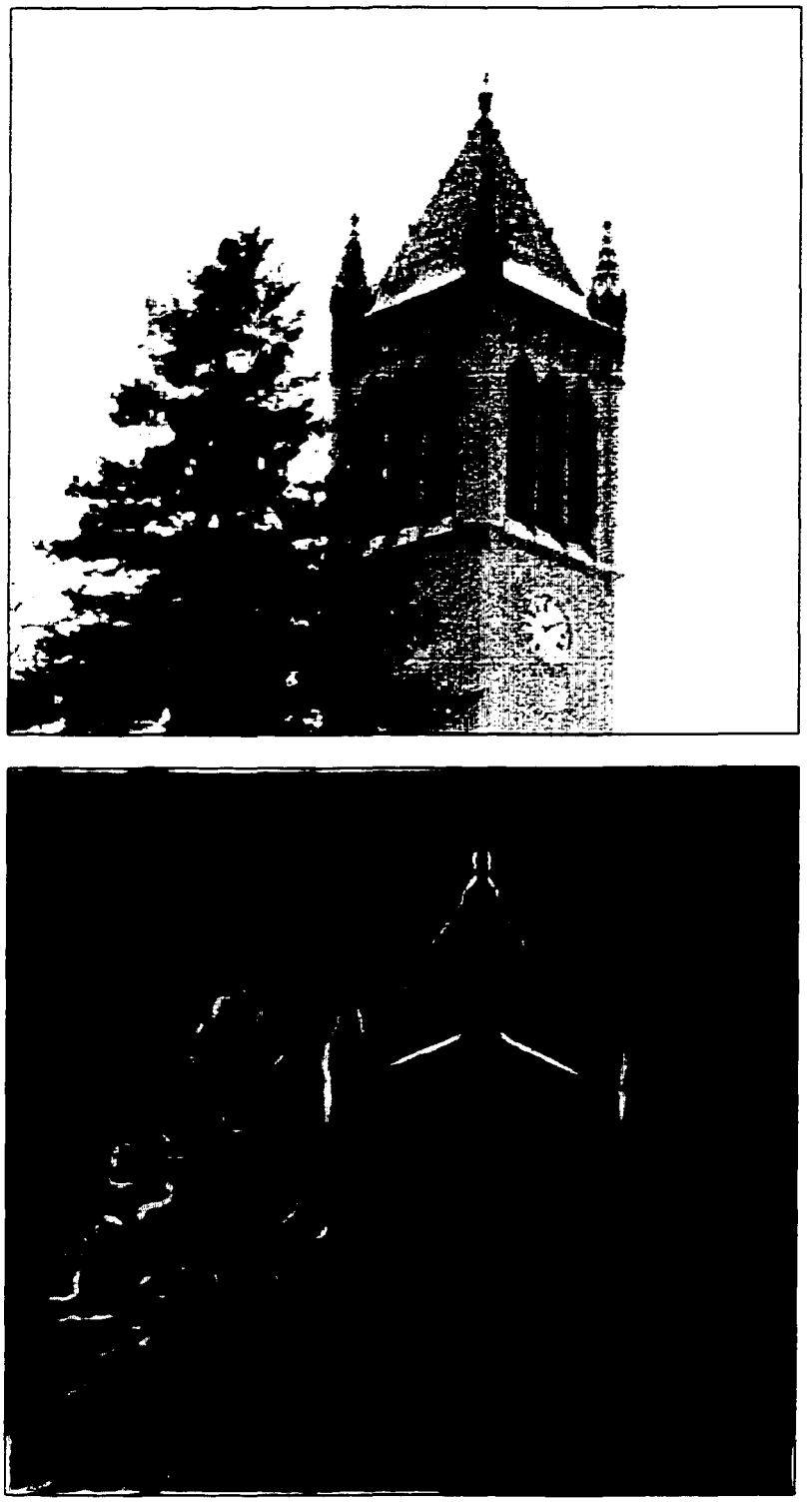


Figure 5.31 Original image (above) and gradient image (below).

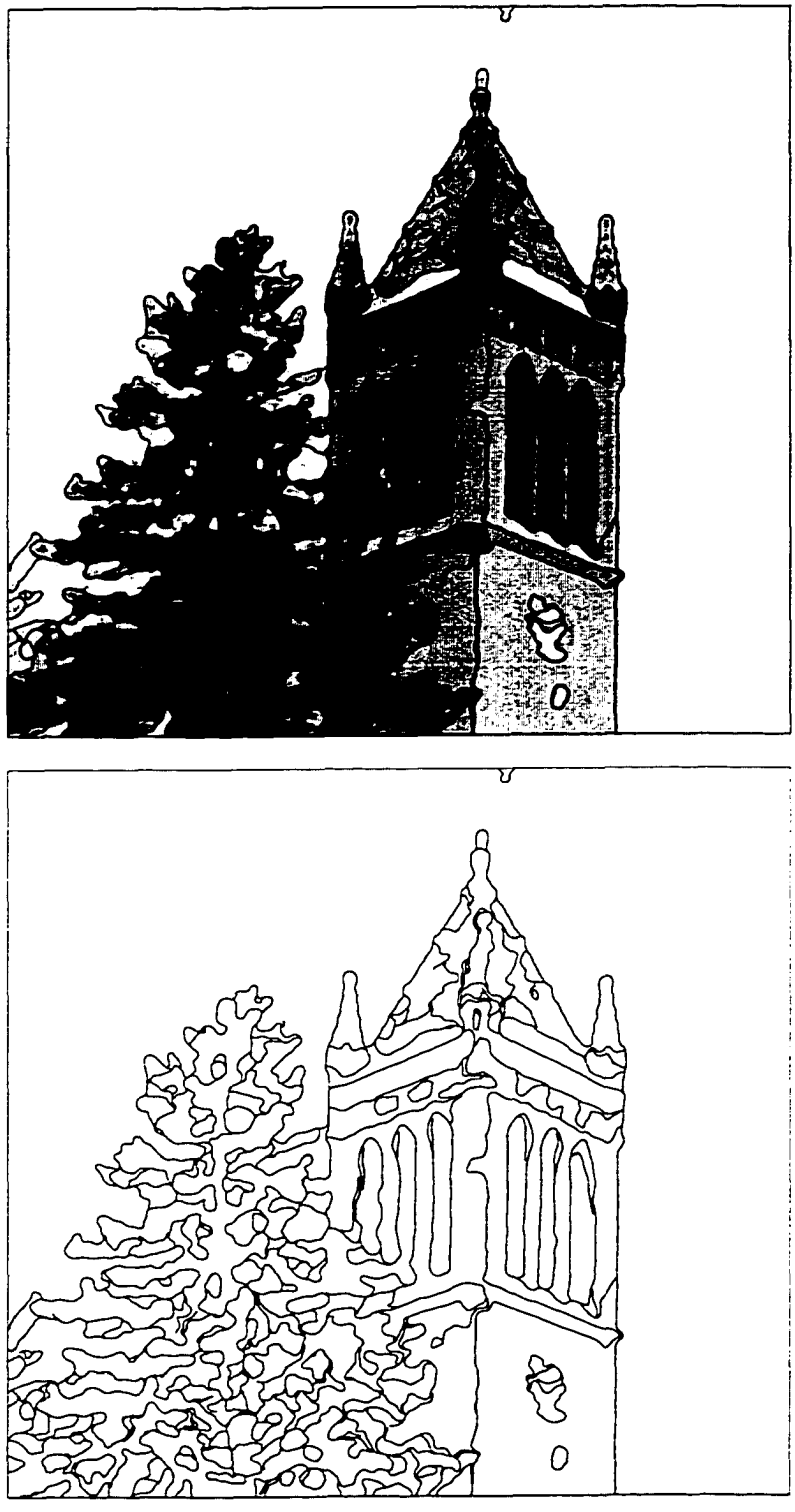


Figure 5.32 Extracted boundaries.

CHAPTER 6. SUMMARY

From scale analysis to boundary extraction, a framework for multiscale boundary extraction has been established. Scale analysis involves analysis of the sizes of structures embedded in an image. Two major concepts of scales, global and local scales have been addressed. Global scales refer to resolutions at which objects in an image exhibit distinctive global patterns while local scales are related to sizes of local structures. This dissertation presents techniques for analysis of global and local scales and multiscale boundary extraction.

A frequency domain approach for global scale analysis based on the differential power spectrum normalized variance ratio (DPSNVR) is developed. A family of multiscale differential masks of a continuum of sizes is used to generate multiscale representation of an image. For global scale detection, the proposed method depends on local minima of the DPSNVR graph. The technique is robust, reliable, and provides accurate estimate of scales even in the presence of noise. Experimental results indicate that the dominant global scales are reflected by local minima of the DPSNVR graph while weak global scales are reflected by local extrema of the slope of the DPSNVR graph. The scale parameters obtained from local minima of the DPSNVR graph are suitable for multiscale boundary extraction.

The second issue related to the problem of scales in image analysis is sizing local structures. Local scale analysis determines the size as well as position of structures based on local maxima of the scale-space differential structure image. Information obtained from local scale analysis can be used for blur estimation and adaptive scale selection for analysis of local structures. In general, local scale analysis requires exhaustive search over scale-space domain and hence is computationally expensive. An alternate method for utilizing multiscale data based on summation of image features with respect to the scale parameter is proposed. This leads to the development of the new concept, the multiple scale differential masks derived by the summation of normalized differential masks of exponentially increasing sizes. The method yields details at several scales simultaneously and therefore preserves structures at those scales. From structures that survive over a

range of scales, the operator eliminates the problem of shape distortion occurring when the large size difference operators are employed. On the other hand, spurious details that do not survive over change of scales are eliminated. The multiple scale differential mask requires a range of scales of interest to be specified instead of a single scale resulting in increasing the scale selection tolerance. In addition, the computational effort needed to compute the multiple scale differential masks is slightly greater than that required in the computation of the conventional multiscale differential mask.

Finally, a generalized boundary extraction algorithm is given. The method is developed from the model of particle motion in a combined orthogonal velocity field. Two velocity fields, the tangential and the normal compressive velocity fields, required in the model are generated from the boundary localization field and the normal velocity field. General expressions as well as suitable choices of the fields are given. To eliminate the inertial effect of the earlier method, the edge localization algorithm is separated into 2 independent steps, one locates edges in the tangential direction and the second step locates edges in the normal direction. Boundary extraction is based on minimization of the absolute values of the boundary localization field along the particle trajectory. The properties of the tangential velocity field and the normal compressive velocity field guarantee that the particle will travel along the paths close to object boundaries. The proposed boundary extraction method is simple, fast, reliable, and yields results with subpixel accuracy.

Beside edge localization, other related issues including estimation of boundary points and boundary representation are also addressed. The boundary points estimation method eliminates redundant boundary points to yield efficient boundary representation. Two schemes, one based on the minimum allowed distance and another based on image features such as corners, are implemented. The method based on image features can significantly reduce the number of boundary representation points while maintaining the overall object shape. For boundary representation, extracted boundaries are represented in the form of planar graphs using doubly-connected edge lists which can represent boundaries of objects of arbitrary topologies.

The methods addressed in this dissertation are parts of intermediate level image analysis. Using the proposed scale analysis techniques for scale selection in conjunction with the boundary extraction algorithm, multiscale image segmentation is achieved. In real applications, further processes such as image interpretation utilizing data obtained from image segmentation are employed to accomplish the final goal of image analysis.

APPENDIX A. 2-DIMENSIONAL SURFACE APPROXIMATION

Bilinear Polynomial Interpolation

For the 2-dimensional equally spaced discretized field f, bilinear polynomial interpolation estimates the field value at a particular point (x, y) from the field values at the 4 neighboring observable points (n, m), $(n+1, m)$, $(n, m+1)$, and $(n+1, m+1)$ as shown in Figure A.1-a by

$$\begin{aligned} f(x, y) &= \sum_{i=0}^{1}\sum_{j=0}^{1} b_i(\Delta x) b_j(\Delta y) f(n+i, m+j) \\ &= \sum_{i=0}^{1}\sum_{j=0}^{1} B_{i,j} f(n+i, m+j) \end{aligned}$$

where $\Delta x = x - n$, $\Delta y = y - m$ while $b_i, i = 0, 1$ are the polynomial interpolation bending functions. For linear interpolation, these functions are given by

$$b_0(u) = 1 - u$$

and

$$b_1(u) = u,$$

and $B_{i,j} = b_i(\Delta x) b_j(\Delta y)$. Note that indices n and m in Equation A.1 are integer parts of x and y respectively where $n < x < n+1$ and $m < y < m+1$. Geometrically, interpolation coefficients $B_{i,j}$ are areas of rectangles as illustrated in Figure A.1-a.

Bicubic B-Spline Polynomial Interpolation

Although bilinear interpolation provides a continuous approximation of the 2-dimensional field, the method uses only 4 neighboring points to estimate the field value causing discontinuities in the first order derivatives of the field. An alternative method that provides the continuous approximations of the field and its derivatives up to order 2 is the bicubic

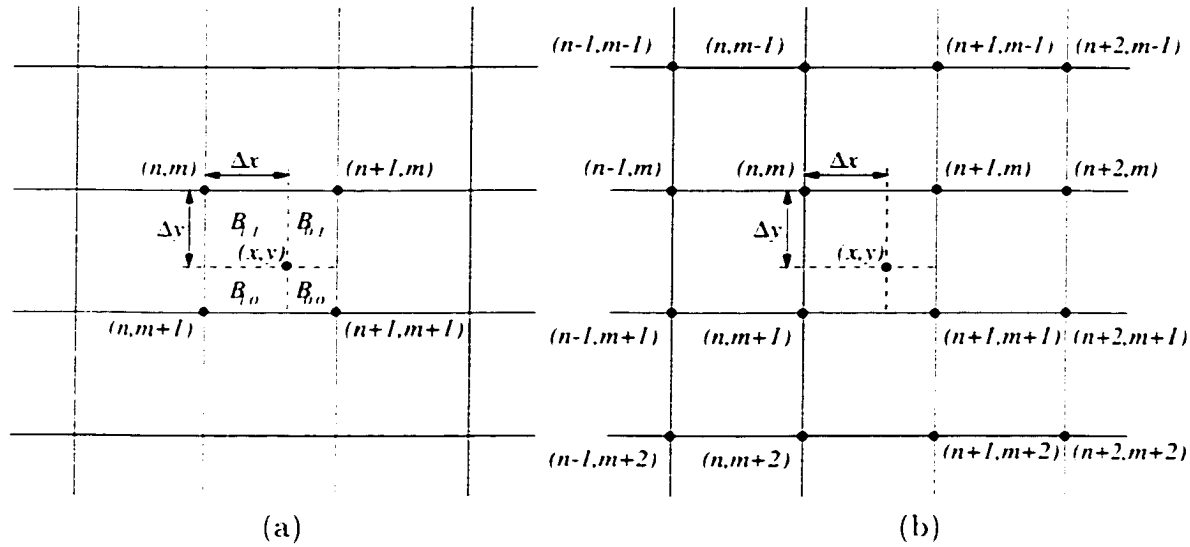


Figure A.1 (a) Interpolation points and coefficients of bilinear interpolation and (b) control points of bicubic B-spline interpolation.

B-spline polynomial approximation:

$$f(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} b_i(\Delta x) b_j(\Delta y) f(n+i-1, m+j-1) \qquad (A.1)$$

where the bending functions b_i are given by

$$\begin{aligned} b_0(u) &= \frac{1}{6}(1-u)^3 \\ b_1(u) &= \frac{1}{6}(4-6u^2+3u^3) \\ b_2(u) &= \frac{1}{6}(1+3u+3u^2-3u^3) \end{aligned}$$

and

$$b_3(u) = \frac{1}{6}u^3.$$

In Equation A.1, the neighboring points $(n+i-1, m+j-1)$, $i,j = 0,\ldots,3$, are not used as the interpolated points but control points of the B-spline surface. As a result, the approximated surface does not pass through but is confined to the convex hulls formed by these points. Despite this configuration, the bicubic B-spline surface provides an accurate polynomial approximation of a 2-dimensional discrete field. Using the field values at 16 neighboring points shown in Figure A.1-b, the bicubic B-spline interpolation yields a smoother continuous approximation compared to that obtained using bilinear interpolation.

APPENDIX B. DATA STRUCTURES AND PROCEDURES FOR CONSTRUCTING BOUNDARY REPRESENTATION

Design of data structures and schemes for representing complete boundary information is crucial. In general, sequences of boundary points are used for representing boundaries. However, this representation is inefficient in the case of boundaries of connected regions with junctions. Due to different topologies of connected regions, special procedures are needed for handling junctions. In addition to the problem of handling junctions, detection of a state when a particle reaches the previously extracted boundary requires exhaustive search to find the extracted boundary point which is closest to the current particle position. However, when the number of extracted boundary points becomes large, the computational time needed increases linearly. Therefore, a heuristic algorithm is needed to reduce the search time.

Doubly-Connected Edge List

In general, a planar graph is used for representing a network and a boundary. A graph consists of vertices, edges, and faces as shown in Figure B.1-a. A vertex stores position information and properties associated with it while an edge is a line segments connecting a pair of vertices. A face is a maximal connected subset of a plane that does not contain a vertex inside. In other words, a face is an open polygonal region bounded by edges and vertices. A graph is consider to be planar if there is no crossing between a pair of edges. In image segmentation, a face corresponds to a segmented region while edges and vertices represent boundaries between regions. One popular structure that used for graph representation is the *doubly-connected edge list* [19]. The doubly-connected edge list consists of 3 collections of records: vertex, half-edge, and face lists. Data structures of the doubly-connected edge list are summarized below:

1. **A vertex record**. Each vertex record contains a coordinate of vertex v and the half-edge pointer, namely the *incident edge*, to one of half-edges that has v as its origin.

2. **A half-edge record**. Each edge is represented by the corresponding pair between half-edge and its twin as shown in Figure B.1-b. Each half-edge has direction opposite to that of its twin and bounds to the face on its left side. A record of the half-edge $\vec{e}$ contains a vertex pointer to the *origin* of $\vec{e}$ and 3 half-edge pointers to its *twin*, *next*, and *previous* half-edges respectively as shown in Figure B.1-b. It also stores a face pointer to the face that $\vec{e}$ bounds, namely the *incident face*. Each half-edge has its unique twin, next, and previous half-edges.

3. **A face record**. Each face record contains one half-edge pointer, namely the *incident edge*, selected from one of the half-edges having f as its incident face. This structure is slightly different from one that used in [19] where two half-edge pointers, inner and outer half-edges are used, [19].

The advantages of using the doubly-connected edge list are that the structure can represent N-dimensional graphs of all topologies and support constructing and information inquiring operations. The overall structure of the doubly-connected edge list is demonstrated by the example in Figure B.2. Tables B.1 to B.3 show all lists of records of the graph in Figure B.2. A boundary of each face can be tracked by following half-edges, starting from the incident edge of the face, until the loop is completed. In cooperating with the proposed boundary extraction algorithm, the doubly-connected edge list supports all operations needed for constructing boundary representation which are listed below.

Initialization

For a given starting point p_0, the boundary graph is created as two half-edges are generated and connected to p_0. When the boundary is not terminated, there always exist the half-edges that attaches to the end vertex of the boundary which are ready to connect to the next extracted boundary point.

1. Create two new half-edges, $\vec{e}_1$ and $\vec{e}_2$.
2. Assign p_0 to the *origin* of $\vec{e}_1$.
3. Assign $\vec{e}_1$ to the *incident edge* of p_0.
4. Assign $\vec{e}_1$ to the *twin*, *next*, and *previous* of $\vec{e}_2$.
5. Assign $\vec{e}_2$ to the *twin*, *next*, and *previous* of $\vec{e}_1$.
6. Add p_0 to the vertex list and $\vec{e}_1$ and $\vec{e}_2$ to the half-edge list.

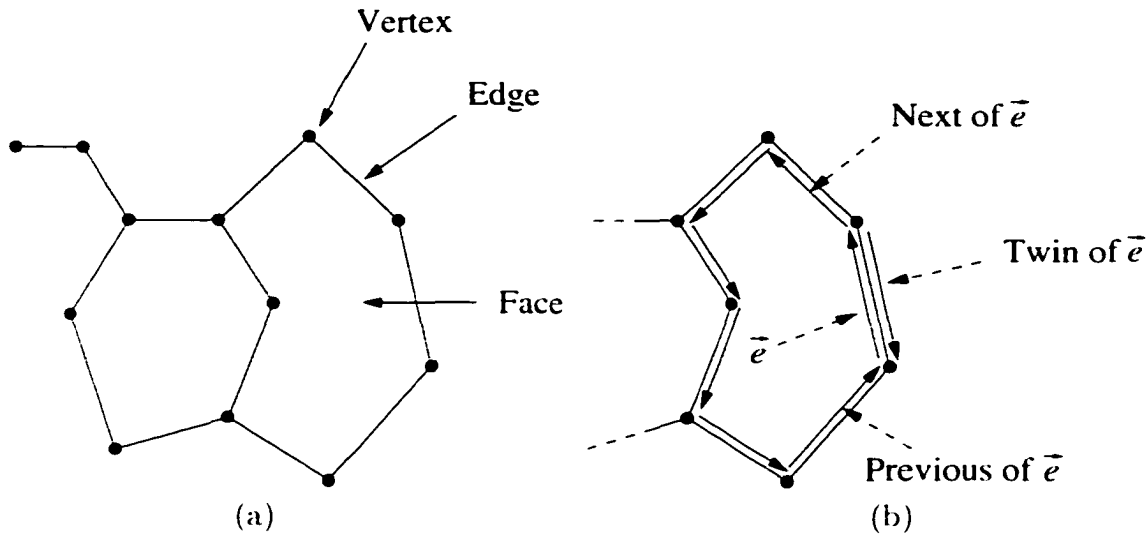


Figure B.1 (a) Planar graph. (b) half-edge and its twin, previous, and next.

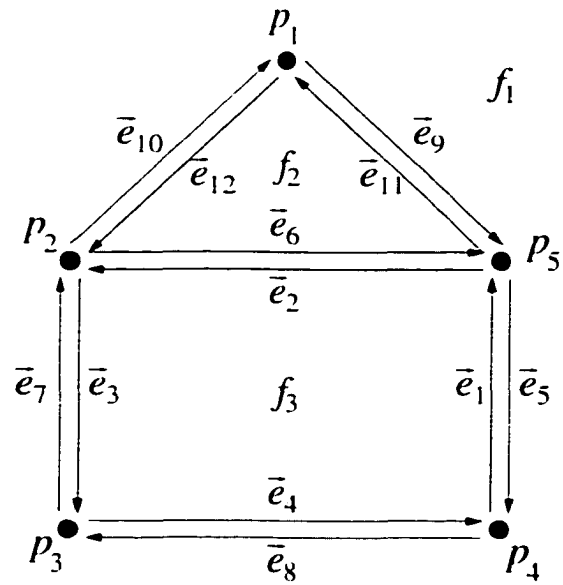


Figure B.2 A graph represented by a doubly-connected edge list.

Table B.1 A face list of the graph in Figure B.2.

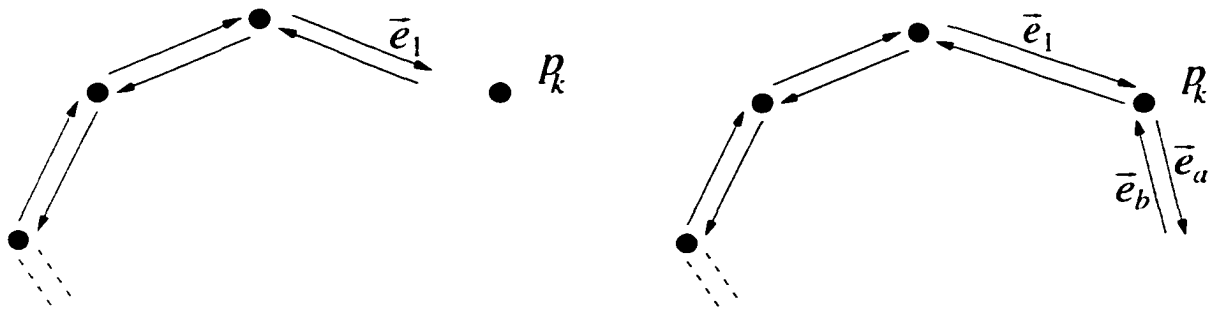
Face	Incident edge
f_1	$\vec{e}_9$
f_2	$\vec{e}_6$
f_3	$\vec{e}_1$

Table B.2 A vertex list of the graph in Figure B.2.

Vertex	Coordinates	Incident edge
p_1	(1.5.3.5)	$\vec{e}_9$
p_2	(0.0.2.0)	$\vec{e}_3$
p_3	(0.0.0.0)	$\vec{e}_4$
p_4	(3.0.0.0)	$\vec{e}_1$
p_5	(3.0.2.0)	$\vec{e}_2$

Table B.3 A half-edge list of the graph in Figure B.2.

Half-edge	Origin	Twin	Next	Previous	Incident face
$\vec{e}_1$	p_4	$\vec{e}_5$	$\vec{e}_2$	$\vec{e}_4$	f_3
$\vec{e}_2$	p_5	$\vec{e}_6$	$\vec{e}_3$	$\vec{e}_1$	f_3
$\vec{e}_3$	p_2	$\vec{e}_7$	$\vec{e}_4$	$\vec{e}_2$	f_3
$\vec{e}_4$	p_3	$\vec{e}_8$	$\vec{e}_1$	$\vec{e}_3$	f_3
$\vec{e}_5$	p_5	$\vec{e}_1$	$\vec{e}_8$	$\vec{e}_9$	f_1
$\vec{e}_6$	p_2	$\vec{e}_2$	$\vec{e}_{11}$	$\vec{e}_{12}$	f_2
$\vec{e}_7$	p_3	$\vec{e}_3$	$\vec{e}_{10}$	$\vec{e}_8$	f_1
$\vec{e}_8$	p_4	$\vec{e}_4$	$\vec{e}_7$	$\vec{e}_5$	f_1
$\vec{e}_9$	p_1	$\vec{e}_{11}$	$\vec{e}_5$	$\vec{e}_{10}$	f_1
$\vec{e}_{10}$	p_2	$\vec{e}_{12}$	$\vec{e}_9$	$\vec{e}_7$	f_1
$\vec{e}_{11}$	p_5	$\vec{e}_9$	$\vec{e}_{12}$	$\vec{e}_6$	f_2
$\vec{e}_{12}$	p_1	$\vec{e}_10$	$\vec{e}_6$	$\vec{e}_{11}$	f_2



(a) (b)

Figure B.3 Extending the boundary: (a) before, (b) after.

Extending a Boundary

To add the newly extracted boundary point p_k to the current boundary, two new half-edges linking p_k to the boundary are created. Let $\vec{e}_1$ be the incident half-edge of the vertex at the end of the boundary to be extended. As shown in Figure B.3, extension of the boundary can be done as follows:

1. Create two new half-edges $\vec{e}_a$ and $\vec{e}_b$.
2. Assign p_k to the *origin* of $\vec{e}_a$ and the *origin* of the *twin* of $\vec{e}_1$.
3. Assign $\vec{e}_a$ to the *incident edge* of p_k.
4. Assign $\vec{e}_b$ to the *twin* and *next* of $\vec{e}_a$ and assign $\vec{e}_a$ to the *twin* and *previous* of $\vec{e}_b$.
5. Assign $\vec{e}_a$ to the *next* of $\vec{e}_1$ and assign $\vec{e}_1$ to the *previous* of $\vec{e}_a$.
6. Assign the *twin* of $\vec{e}_1$ to the *next* of $\vec{e}_b$ and assign $\vec{e}_b$ to the *previous* of the *twin* of $\vec{e}_1$.
7. Add p_k to the vertex list and $\vec{e}_a$ and $\vec{e}_b$ to the half-edge list.

Closing a Boundary

When the particle reaches its starting point, the boundary is closed by linking the half-edges to the two end vertices of the boundary as shown in Figure B.4. Let p_0 and p_n be the start and end vertices respectively.

1. $\vec{e}_0$:= the *incident edge* of p_0

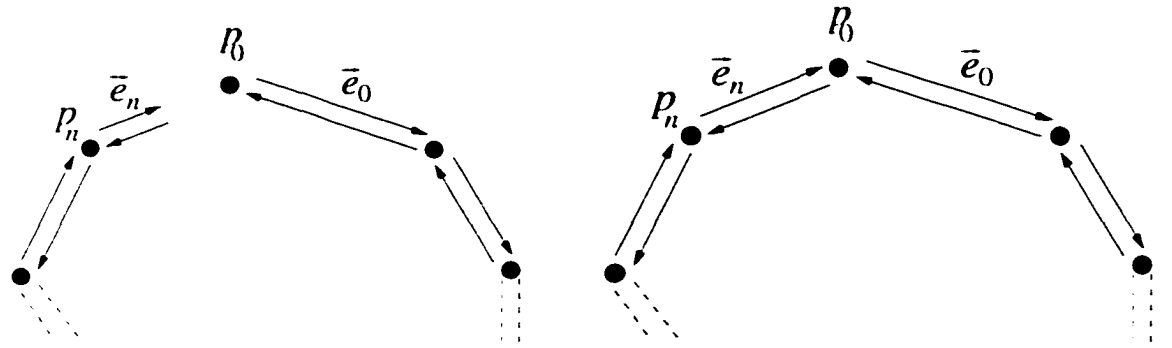


Figure B.4 Closing the boundary: (a) before, (b) after.

2. $\vec{e}_n$:= the *incident edge* of p_n

3. Assign p_0 to the *origin* of the *twin* of $\vec{e}_n$

4. Assign $\vec{e}_n$ to the *previous* of $\vec{e}_0$ and assign $\vec{e}_0$ to the *next* of $\vec{e}_n$.

5. Assign the *twin* of $\vec{e}_n$ to the *next* of the *twin* of $\vec{e}_0$ and assign the *twin* of $\vec{e}_0$ to the *previous* of the *twin* of $\vec{e}_n$.

6. Update *incident faces* of $\vec{e}_0$ and its *twin*.

Inserting a Junction.

Inserting a junction can be done by manipulating the records. After a junction is inserted, the incident faces of half-edges connected to the junction are needed to be updated. Let p_0, $\vec{e}_1$, and $\vec{e}_2$ are a vertex and half-edges where a junction will be inserted as shown in Figure B.5-a. Let p_n and $\vec{e}_n$ be the end vertex and its incident edge of the current boundary that will connect to p_0 to create a junction. The algorithm for creating a junction is given below:

1. Assign p_0 to the *origin* of the *twin* of $\vec{e}_n$.

2. Assign $\vec{e}_2$ to the *next* of $\vec{e}_n$ and assign $\vec{e}_n$ to the *previous* of $\vec{e}_2$.

3. Assign $\vec{e}_1$ to the *previous* of the *twin* $\vec{e}_n$ and assign the *twin* of $\vec{e}_n$ to the *next* of $\vec{e}_1$.

4. Update *incident faces* of $\vec{e}_n$ and its *twin*.

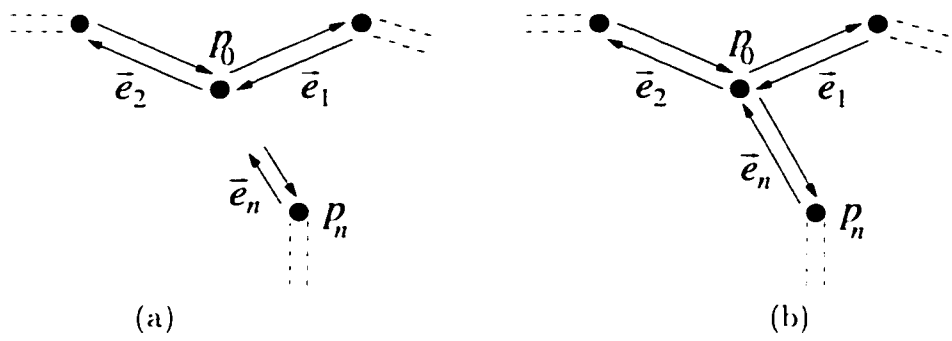


Figure B.5 Inserting a junction: (a) before, (b) after.

Updating an Incident Face

Updating the incident face is performed when boundary is closed or a junction is inserted. The incident faces of all half-edges connected to the half-edge $\vec{e}_0$ and its *twin* that make the last connection are needed to be updated.

1. $\vec{e}_1 := \vec{e}_0$.
2. **if** the *incident face* of $\vec{e}_1$ is null
   then Create a new face f_1.
   Add f_1 to the face list.
3. **else**
   $f_1 :=$ the *incident face* of $\vec{e}_1$.
4. Assign $\vec{e}_1$ to the *incident edge* of f_1.
5. $have2faces := true$.
6. **do**
   if the *incident face* of $\vec{e}_1 \neq$ null and the *incident face* of $\vec{e}_1 \neq f_1$
   then Remove the *incident face* of $\vec{e}_1$ from the face list.
   Assign f_1 to the *incident face* of $\vec{e}_1$.
   if $\vec{e}_1 =$ the *twin* of $\vec{e}_0$
   then $have2faces := false$.
   $\vec{e}_1 :=$ the *next* of $\vec{e}_1$.

7. **while** $\vec{e}_1 \neq \vec{e}_0$.

8. **if** $have2face = true$

9. **then** $\vec{e}_1$:= the *twin* of $\vec{e}_0$.

10. **if** the *incident face* of $\vec{e}_1$ is null

then Create a new face f_1.

Add f_1 to the face list.

11. **else**

f_1 := the *incident face* of $\vec{e}_1$.

12. Assign $\vec{e}_1$ to the *incident edge* of f_1.

13. **do**

14. **if** the *incident face* of $\vec{e}_1 \neq$ null and the *incident face* of $\vec{e}_1 \neq f_1$

then Remove the *incident face* of $\vec{e}_1$ from the face list.

15. Assign f_1 to the *incident face* of $\vec{e}_1$.

16. $\vec{e}_1$:= the *next* of $\vec{e}_1$.

17. **while** $\vec{e}_1 \neq$ the *twin* of $\vec{e}_0$.

Detection of the Condition for Creating a Junction

In order to detect the condition for creating a new junction, an exhaustive search is conducted to find the boundary point closest to the current particle position. However, when the number of extracted boundary points becomes large, computational time required for the exhaustive search grows linearly. A heuristic algorithm that reduces searching time can be done by dividing the whole image into smaller areas and assigning to each area a list of boundary points located inside the area. When the particle enters each area, only the boundary points in the area are needed to be tested. A collection of such areas is called the *trajectory map*. A trajectory map contains an array of size $(N+1) \times (M+1)$ of T-pixels where $N \times M$ is a corresponding image size. Each T-pixel occupies an area of size 1 image pixel which overlaps each neighboring image pixel by a quarter pixel as shown in Figure B.6. Since the size of a T-pixel is small enough so that

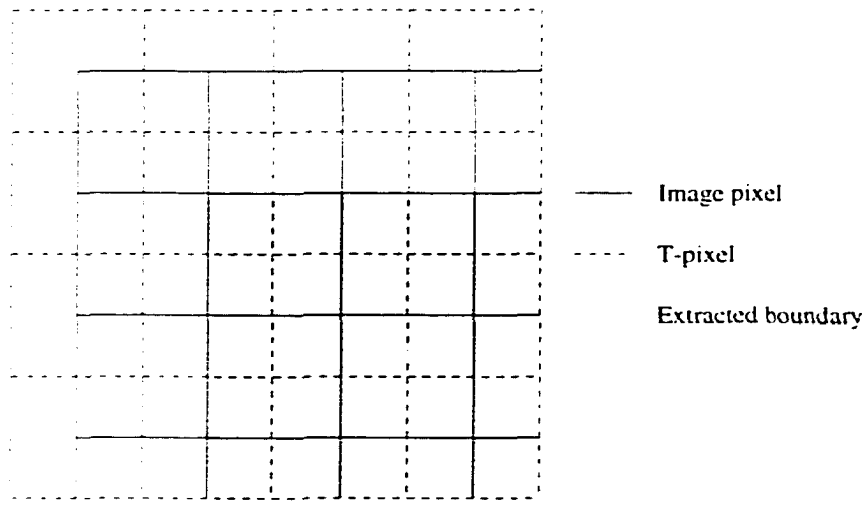


Figure B.6 A trajectory map.

no more than one particle trajectory can pass through each T-pixel, instead of recording all boundary points inside the T-pixel, only one vertex pointer, namely the *center vertex*, to the boundary point closest to the center of the T-pixel and inside the T-pixel is recorded. Since vertices are connected by half-edges, all vertices inside each T-pixel can be tracked via the center vertex. If no boundary passes through the T-pixel, the null value will be assigned to the center vertex pointer. The overlapping configuration between image pixels and T-pixels is to ensure that the particle trajectory is most likely to pass to the center of the T-pixel since boundary points usually locate near borders of image pixels. The pseudo-code for detecting the condition for creating a junction and terminating the boundary is summarized below.

When the particle enters a T-pixel at position p_0, the condition for creating a junction is needed to be tested only if the number of connected vertices of the *center vertex* of T is greater than 1. Let $\vec{e}_0$ be the *incident edge* of the *center vertex* of T. The procedure returns the pointer to the vertex inside T that is closest to p_0 as well as the distance and the status of the condition for creating a junction.

1. **if** $\vec{e}_0$ is null or the number of connected vertices of the *center vertex* < 2

   then *creating_junction* $:= false$

2. **else**

3. $\vec{e}_2 := \vec{e}_0$.

4. $\vec{e}_1 := \vec{e}_0$.

5. p_1 := the *origin* of $\vec{e}_1$.

6. d_{min} := the distance between p_0 and p_1.

7. $p_{min} := p_0$.

8. **do**

9. **do**

10. $\vec{e}_1$:= the *next* of $\vec{e}_1$.

11. p_1 := the *origin* of $\vec{e}_1$.

12. **if** p_1 is inside T **then**

13. **if** the distance between p_0 and p_1 is less than d_{min}
then d_{min} := the distance between p_0 and p_1.
$p_{min} := p_1$.

14. **while** p_1 is inside T.

15. $\vec{e}_2$:= the *next* of the *twin* of $\vec{e}_2$.

16. $\vec{e}_1$:= the *next* of $\vec{e}_2$.

17. **while** $\vec{e}_2 \neq \vec{e}_0$.

18. **if** $d_{min} \leq \epsilon$
then *creating_junction* := *true*.

19. **else**
creating_junction := *false*.

20. **return** *creating_junction*, p_{min}, and d_{min}.

Complete Boundary Extraction Algorithm

The pseudo code for the overall boundary extraction algorithm is given in this section. The inputs are the tangential and normal compressive velocity fields, the tangential step size α, a list of starting points P_{start}, and the feature image used for marking the boundary points.

```
1.  while the number of extracted boundaries < N_max
2.      do
3.          Get p_0 from the list P_start.
4.      until p_0 is in the T-pixel that has no boundary on it.
5.      Add p_0 into the vertex list.
6.      Creating the boundary using p_0 as a starting point.
7.      Assign p_0 to the T-pixel that p_0 lies on.
8.      direction := forward.
9.      track2way = false
10.     do
11.         p_1 := p_0.
12.         while p_1 is inside an image
13.             if direction = forward
                    then p_1 := find_next_forward_boundary_point(p_1).
14.             else
                        p_1 := find_next_backward_boundary_point(p_1).
15.             if p_1 is outside an image
                    then Update the incident faces of the current boundary.
                         break.
```

16. **else if** the boundary point recording condition at p_1 is met
then Extend the current boundary to p_1.
Add p_1 to the vertex list.

17. **if** p_1 is trapped in a local trap
then Update the incident faces of the current boundary.
break.

18. **else if** p_1 reaches the starting point p_k of some previous boundaries
then Close the boundary by linking
the current boundary to p_k.
Update the incident faces of the current boundary.
break.

19. **else if** the condition for creating the junction at p_1 is met
then Insert a junction by connecting the previously marked
vertex to the nearest vertex.
Update the incident faces of the current boundary.
break.

20. **else**

21. **if** p_1 is marked and is closest to the center of the current T-pixel
then Assign p_1 to the incident vertex of
the current T-pixel.

22. **if** $direction = forward$.
then $direction := backward$.

23. **else**
$track2way := true$.

24. **while** $track2way = false$.

Figure B.7 demonstrates the use of the doubly-connected edge list for boundary representation of multiple connected objects. The boundary of each region can be obtained by retrieving the boundary of the corresponding face as shown in Figures B.7-b to B.7-e.

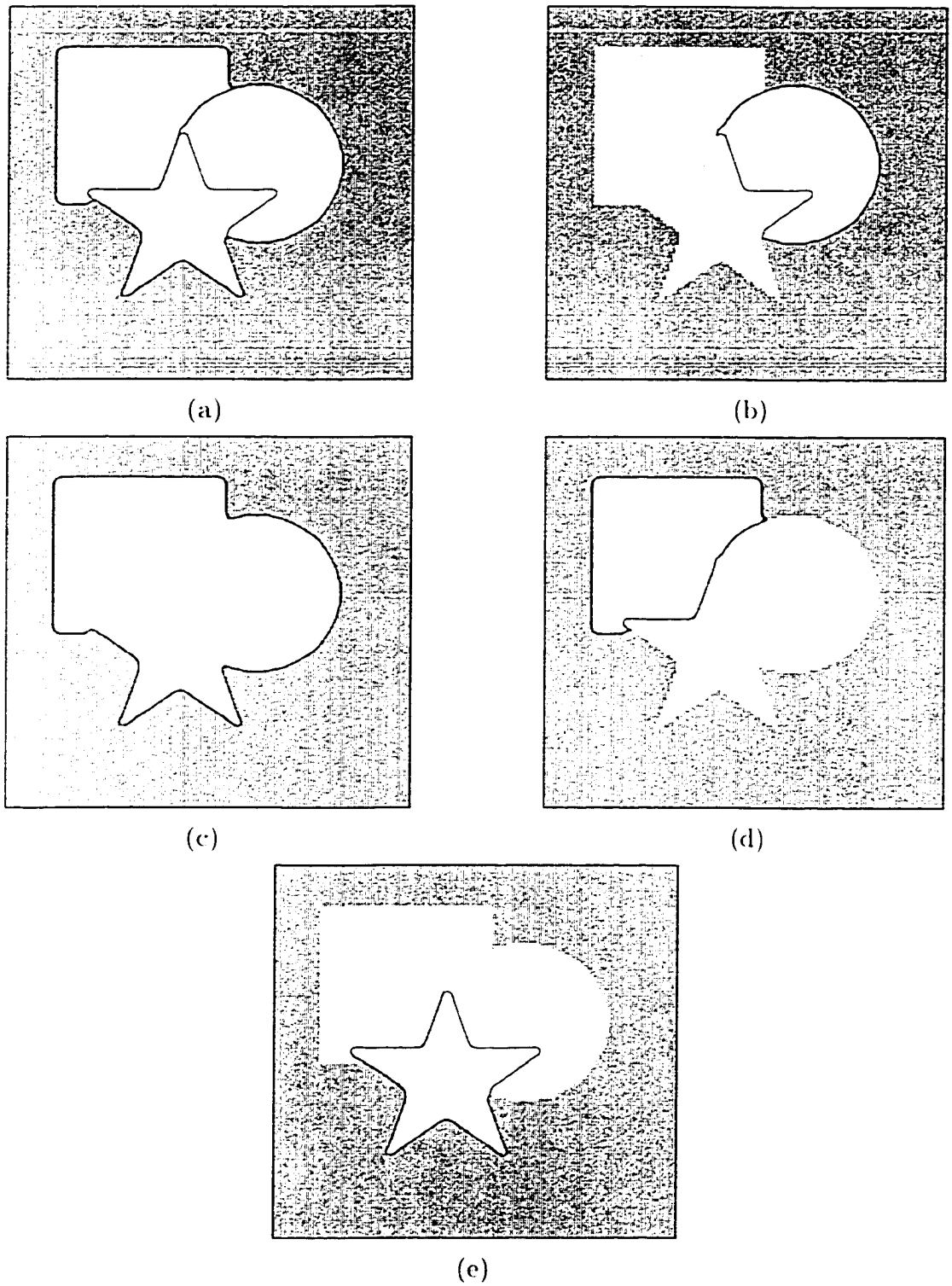


Figure B.7 Boundaries of multiple connected objects represented using a doubly-connected edge list: (a) boundaries of all faces, (b) to (e) boundary of each face.

BIBLIOGRAPHY

[1] A. Almansa and T. Lindenberg. Enhancement of fingerprint images using shape-adapted scale-space operators. In J. Sporring, M. Nielsen, L. Florack, and P. Johansen, editors, *Gaussian Scale-Space Theory*, pages 21–30. Kluwer Academic Publishers, Boston, 1997.

[2] E. Argyle. Techniques for edge detection. *Proceedings of the IEEE (Letters)*, pages 285–287, 1970.

[3] G. P. Ashkar and J. W. Modestino. The contour extraction problem with biomedical applications. *Computer Graphics and Image Processing*, 7:331–355, 1978.

[4] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):26–33, January 1986.

[5] F. Bergholm. Edge focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(6):726–741, November 1987.

[6] A. Blake and M. Isard. *Active Contours: the Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag, New York, 1998.

[7] D. Brzakovic. Spline models for boundary detection/description: Formulation and performance evaluation. *CVGIP: Graphical Models and Image Processing*, 53(4):392–401, July 1991.

[8] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):425–455, July 1986.

[9] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[10] A. K. Cherri and M. A. Karim. Optical symbolic substitution: Edge detection using prewitt, sobel, and roberts operators. *Applied Optics*, 28(21):4644–4648, November 1989.

[11] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.

[12] L. D. Cohen and I. Cohen. Finite element methods for active contour models and ballons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, November 1993.

[13] L. D. Cohen and R. Kimmel. Global minimum for active contour models: A minimum path approach. *International Journal of Computer Vision*, 24(1):57–78, 1997.

[14] A. C. F. Colchester, R. T. Ritchings, and N. D. Kodikara. Image segmentation using maximum gradient profiles orthogonal to edges. *Image and Vision Computing*, 8(3):211–217, 1990.

[15] D. B. Cooper. Maximum likelihood estimation of markov-process blob boundaries in noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(4):372–384, October 1979.

[16] D. B. Cooper, H. Elliott, F. Cohen, L. Reiss, and P. Symosek. Stochastic boundary estimation and object recognition. *Computer Graphics and Image Processing*, 12:326–256, 1980.

[17] T. Darrell and K. Wohn. Pyramid based depth from focus. In *Proc. IEEE Comput. Vision Pattern Recognition*, pages 504–509, 1988.

[18] S. Dayanand, W. R. Uttal, T. Shepherd, and C. Lunskis, Jr. A particle system model for combining edge information from multiple segmentation modules. *CVGIP: Graphical Models and Image Processing*, 56(5):219–230, May 1994.

[19] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, New York, 1997.

[20] J. H. Elder and S. W. Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716, July 1998.

[21] H. Elliott and L. Srinivasan. An application of dynamic programming to sequential boundary estimation. *Computer Graphics and Image Processing*. 17:291–314. 1981.

[22] N. Eua-Anant. A novel boundary extraction algorithm based on a vector image model. Master's thesis. Iowa State University. Ames. IA 50010. 1996.

[23] N. Eua-Anant and L. Udpa. Edge detection using image moment vectors. In *Proceedings of the Midwest Electro-Technology Conference*. pages 23–26. Iowa State University. 1995.

[24] N. Eua-Anant and L. Udpa. Boundary detection using simulation of particle motion in a vector image field. *IEEE Transactions on Image Processing*. 8(11):1560–1571. November 1999.

[25] N. Eua-Anant and L. Udpa. Scale detection based on statistical characteristics of edges in the scale space. In *Proceedings of the IEEE International Conference on Acoustics. Speech. and Signal Processing*. volume 6. pages 3297–3300. 1999.

[26] L. M. J. Florack. B. M. ter Haar Romeny. J. J. Koenderink. and M. A. Viergever. Scale and the differential structure of images. *Image and Vision Computing*. 10(6):376–388. July/August 1992.

[27] W. Frei and C.-C. Chen. Fast boundary detection: A generalization and a new algorithm. *IEEE Transactions on Computers*. C-26(10):988–998. October 1977.

[28] J. M. Gauch. *Multiresolution Image Shape Description*. Springer-Verlag New York. Inc.. New York. 1991.

[29] J. M. Gauch. Image segmentation and analysis via multiscale gradient watershed hierarchies. *IEEE Transactions on Image Processing*. 8(1):69–79. January 1999.

[30] D. Geiger. A. Gupta. L. A. Costa. and J. Vlontzos. Dynamic programming for detecting. tracking. and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 17(3):294–302. March 1995.

[31] R. C. Gonzalez and R. E. Wood. *Digital Image Processing*. Addison-Wesley Publishing Company. New York. 1992.

[32] A. Goshtasby. On edge focusing. *Image and Vision Computing*. 12(4):247–255. May 1994.

[33] A. Goshtasby and H.-L. Shyu. Edge detection by curve fitting. *Image and Vision Computing*. 13(3):169–177. April 1995.

[34] L. D. Griffin. A. C. F. Colchester. and G. P. Robinson. Scale and segmentation of grey-level images using maximum gradient paths. *Image and Vision Computing*. 10(6):389–402. 1992.

[35] S. R. Gunn and M. S. Nixon. A robust snake implementation: a dual active contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19(1):63–67. January 1997.

[36] R. M. Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-6(1):58–68. January 1984.

[37] R. M. Haralick and L. G. Shapiro. Image segmentaion techniques. *Computer Vision. Graphics. and Image Processing*. 29:100–132. 1985.

[38] A. Huertas and G. Medioni. Detection of intensity changes with subpixel accuracy using laplacian-gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-8(5):651–664. September 1986.

[39] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall. Inc.. 1989.

[40] R. Jain. R. Kasturi. and B. G. Schunck. *Machine Vision*. McGraw-Hill. Inc.. 1995.

[41] M. Kass. A. Witkin. and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*. pages 312–333. 1988.

[42] R. A. Kirsch. Computer determination of the constituent structure of biological images. *Computers and Biomedical Research*. 4:315–328. 1971.

[43] J. J. Koenderink. The structure of images. *Biological Cybernetics*. 50:363–370. 1984.

[44] K. F. Lai and R. T. Chin. Deformable contours: Modeling and extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 17(11):1084–1090. November 1995.

[45] T. Law. H. Itoh. and H. Seki. Image filtering. edge detection. and edge tracing using fuzzy reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 18(5):481–491. May 1996.

[46] M. D. Levine. *Vision in Man and Machine.* McGraw-Hill, Inc., 1985.

[47] L. M. Lifshitz and S. M. Pizer. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 12(6):529–540. June 1990.

[48] T. Lindenberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 12(3):234–254. March 1990.

[49] T. Lindenberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers. Boston. 1994.

[50] T. Lindenberg. *Scale-Space Theory in Computer Vision.* chapter 10. pages 249–259. Kluwer Academic Publishers. Boston. 1994.

[51] T. Lindenberg. *Scale-Space Theory in Computer Vision.* chapter 13. pages 317–347. Kluwer Academic Publishers. Boston. 1994.

[52] T. Lindenberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision.* 30(2):117–154. 1998.

[53] T. Lindenberg. Feature detection with automatic scale selection. *International Journal of Computer Vision.* 30(2):79–116. 1998.

[54] M. M. Lipschutz. *Schaum's Outline of Thoery and Problems of Differential Geometry.* McGraw-Hill, Inc., New York. 1969.

[55] C. Lorenz, I.-C. Carlsen, T. M. Buzug, C. Fassnacht, and J. Weese. A multi-scale line filter with automatic scale selection based on the hessian matrix for medical image segmentation. In B. ter Haar Romeny, L. Florack, J. Koenderink, and M. Viergever, editors. *Lecture Notes in Computer Science: Scale-Space Theory in Computer Vision.* volume 1252. pages 152–163. Springer. 1997.

[56] Y. Lu and R. C. Jain. Reasoning about edges in scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 14(4):450–468. April 1992.

[57] L. M. Luo, C. Hamitouche, J. L. Dillenseger, and J. L. Coatrieux. A moment-based three-dimensional edge operator. *IEEE Transactions on Biomedical Engineering.* 40(7):693–703. July 1993.

[58] E. P. Lyvers, O. R. Mitchell, M. L. Akey, and A. P. Reeves. Subpixel measurements using a moment-based edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 11(12):1293–1309. December 1989.

[59] R. Machuca and A. L. Gilbert. Finding edges in noisy scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-3(1):103–111, January 1981.

[60] I. D. G. Macleod. On finding structure in pictures. In S. Kaneff, editor, *Picture Language Machines*, pages 231–256. New York:Academic. 1970.

[61] D. Marr. *Vision: A Computational Investigation into Human Representation and Processing of Visual Information*. W. H. Freeman, San Francisco. 1982.

[62] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B*. 207:187–217. 1980.

[63] A. Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processing*. 1:169–182. 1972.

[64] A. Martelli. An application of heuristic search methods to edge and contour detection. *Communications of the ACM*. 19(2):73–83. February 1976.

[65] U. Montanari. On the optimal detection of curves in noisy pictures. *Communications of the ACM*. 14(5):335–345. February 1971.

[66] C. Murray and M. O'Malley. Segmentation of plant cell pictures. *Image and Vision Computing*. 11(3):155–162. April 1993.

[67] O. F. Olsen. Multi-scale watershed segmentation. In J. Sporring, M. Nielsen, L. Florack, and P. Johansen, editors, *Gaussian Scale-Space Theory*, pages 191–200. Kluwer Academic Publishers, Boston, 1997.

[68] A. P. Reeves. A moment based two-dimensional edge operator. In *Proc. IEEE Comput. Vision Pattern Recognition*, pages 312–317, 1983.

[69] G. S. Robinson. Edge detection by compass gradient masks. *Computer Graphics and Image Processing*. 6:492–501, 1977.

[70] R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*. 13(2):229–251. 1994.

[71] A. Rosenfeld. A nonlinear edge detection technique. *Proceedings of the IEEE*. pages 814-816. May 1970.

[72] A. Rosenfeld and M. Thurston. Edge detection for visual scene analysis. *IEEE Transactions on Computers*. C-20(5):562-569. May 1971.

[73] I. Rothe. H. Süsse. and K. Voss. The method of normalization to determine invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 18(4):366-3761. April 1996.

[74] M. Sonka. V. Hlavac. and R. Boyle. *Image Processing. Analysis and Machine Vision*. Chapman & Hall Computing. London. 1993.

[75] J. Sporring. M. Nielsen. L. Florack. and P. Johansen. editors. *Gaussian Scale-Space Theory*. Kluwer Academic Publishers. Boston. 1997.

[76] M. Subbarao and N. Gurumoorthy. Depth recovery from blurred edges. In *Proc. IEEE Comput. Vision Pattern Recognition*. pages 498-503. 1988.

[77] M. Subbarao and G. Surya. Depth from defocus: A spatial domain approach. *International Journal of Computer Vision*. 13(3):271-294. 1994.

[78] A. J. Tabatabai and O. R. Mitchell. Edge location to subpixel values in digital imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-6(2):188-201. March 1984.

[79] B. M. ter Haar Romeny. L. M. J. Florack. A. H. Salden. and M. A. Viergever. Higher order differential structure of images. *Image and Vision Computing*. 12(6):317-325. July/August 1994.

[80] J. Weickert. S. Ishikawa. and A. Imiya. On the history of gaussian scale-space axiomatics. In J. Sporring. M. Nielsen. L. Florack. and P. Johansen. editors. *Gaussian Scale-Space Theory*. pages 45-58. Kluwer Academic Publishers. Boston. 1997.

[81] D. J. Williams and M. Shah. Edge contours using multiple scales. *Computer Vision. Graphics. and Image Processing*. 51:256-274. 1990.

[82] A. P. Witkin. Scale-space filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence*. pages 1019-1022. Karlsruhe, Germany. 1983.

[83] A. L. Yuille and T. A. Poggio. Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):15–25, January 1986.

[84] W. Zhang and F. Bergholm. Multi-scale blur estimation and edge type classification for scene analysis. *International Journal of Computer Vision*, 24(3):219–250, 1997.

ACKNOWLEDGEMENTS

Many thanks must go to a number of people who have helped and supported me in the completion of my Ph. D. program. First and foremost, I would like to express my sincere gratitude to my major professor, Dr. Lalita Udpa, for her guidance, encouragement, support, and understanding throughout my graduate studies. With her guidance, knowledge, kindness, and tireless devotion, she has shown me how to become a great teacher. I am very grateful to have been able to work for her.

I would like to give special thanks to my committee members, Dr. Satish S. Udpa, Dr. William Lord, Dr. Joseph N. Gray, and Dr. Julie A. Dickerson for their excellent suggestions and encouragement. Thanks also to all the members of the Materials Assessment Research Group for their contributions, collaboration, and friendship throughout my research. I deeply appreciate the Department of Electrical and Computer Engineering, Iowa State University, and the Royal Thai Government for providing me financial support and wonderful opportunity to study here.

I would like to thank all my friends, especially, Supachai Pathumnakul, Vara Varavithya, Narong Praphairaksit, Supapan Sangnui, Sumate Chaiprapat, Jaruwan Klamklay, Dynaya Bhutipunthu, Suntichai Shevasuthisilp, Kulwara Meksawan, and Worapa Anothayanon for their great friendship that made my life in Ames enjoyable.

My heartfelt thanks directed to Pornpimon Saiyonk for her affectionate love, care, patience, and support throughout these years in graduate school. Finally, I would give all my gratitude to my parents, Subhon and Yaowalak Eua-Anant, my sister, Nawanant Piyavhatkul, and my brothers, Navaphol and Nawanat Eua-Anant, for their unconditional love and support during this time that I have been so far away from home.