

ABSTRACT

Lee, Cheolha Pedro. Robust Image Segmentation using Active Contours: Level Set Approaches.
(Under the direction of Dr. Wesley Snyder).

Image segmentation is a fundamental task in image analysis responsible for partitioning an image into multiple sub-regions based on a desired feature. Active contours have been widely used as attractive image segmentation methods because they always produce sub-regions with continuous boundaries, while the kernel-based edge detection methods, e.g. Sobel edge detectors, often produce discontinuous boundaries. The use of level set theory has provided more flexibility and convenience in the implementation of active contours. However, traditional edge-based active contour models have been applicable to only relatively simple images whose sub-regions are uniform without internal edges.

A partial solution to the problem of internal edges is to partition an image based on the statistical information of image intensity measured within sub-regions instead of looking for edges. Although representing an image as a piecewise-constant or unimodal probability density functions produces better results than traditional edge-based methods, the performances of such methods is still poor on images with sub-regions consisting of multiple components, e.g. a zebra on the field. The segmentation of this kind of multispectral images is even a more difficult problem. The object of this work is to develop advanced segmentation methods which provide robust performance on the images with non-uniform sub-regions.

In this work, we propose a framework for image segmentation which partitions an image based on the statistics of image intensity where the statistical information is represented as a mixture of probability density functions defined in a multi-dimensional image intensity space. Depending on the method to estimate the mixture density functions, three active contour models are proposed: unsupervised multi-dimensional histogram method, half-supervised multivariate Gaussian mixture density method, and supervised multivariate Gaussian mixture density method. The implementation of active contours is done using level sets.

The proposed active contour models show robust segmentation capabilities on images where traditional segmentation methods show poor performance. Also, the proposed methods provide a means of autonomous pattern classification by integrating image segmentation and statistical pattern classification.

Robust Image Segmentation using Active Contours: Level Set Approaches

by

Cheolha Pedro Lee
Dept. of Electrical and Computer Engineering
North Carolina State University

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering

Raleigh

2005

Approved By:

Dr. Hamid Krim

Dr. Griff Bilbro

Dr. John Franke

Dr. Cliff Wang

Dr. Wesley Snyder
Chair of Advisory Committee

Biography

Cheolha Pedro Lee was born in February, 1974 in Chonju, South Korea. He graduated from Chonbuk National University with the Bachelor of Engineering degree in Control and Instrumentation Engineering in February, 1999. He moved to the United States for graduate school in June 1999, and has been studying image processing and computer vision since then. He received the Master of Science degree in Electrical and Computer Engineering from the University of Tennessee at Knoxville in August, 2001, and transferred to North Carolina State University, Raleigh, North Carolina for the PhD program. His current research interests include image processing, computer vision, and pattern classification.

Acknowledgements

I would like to thank my advisor, Dr. Wesley Snyder, for his guidance and friendship. He has shown me what the role model of an educator is. Although he already has the experience of more than 30 years, he is always willing to learn new knowledge, and I am the beneficiary of the knowledge. He has been not only a good teacher but also a good friend. I do not think any other graduate student would have a friendly relation with their advisor as I have with Dr. Snyder. I also appreciate the members of my graduate committee, Dr. Cliff Wang, Dr. Griff Bilbro, Dr. Hamid Krim, and Dr. John Franke for their academic advice and the effort to review this work.

Although I did not mention in the first place, I would like to express the most gratitude to my wife Soomi Kim for her endless love, trust, and support. There have been many challenging moments, but she has always encouraged and supported me no matter how difficult they were. Without her, I could not finish this work. I also thank my parents, grand mother, and parents-in-law for their support and love.

Table of Contents

List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Image Segmentation and Active Contours	2
1.2 Multispectral Images	4
1.3 Motivations	6
2 Image Segmentation	9
2.1 Edge-based Segmentation	9
2.2 Region-based Segmentation	12
2.3 Other Segmentation Methods	14
3 Active Contours	16
3.1 Snakes	17
3.2 Level Set Methods	19
3.3 Edge-based Active Contours	22
3.4 Region-based Active Contours	25
3.5 Active Contours Integrating Edge- and Region-based Segmentation	29
4 Region-based Segmentation	30
4.1 Image, Subset, and Sub-region	30
4.2 The Base Segmentation Model	33
4.3 Proposed Region-based Segmentation Model	35
5 Probability Density Model	38
5.1 Mixture Density Function	38
5.2 Multivariate Mixture Density Function	40
6 Density Estimation Methods	44
6.1 Parametric Density Estimation Methods	45
6.1.1 EM Algorithm	46
6.1.2 EM Algorithm with MML	49
6.2 Non-parametric Density Estimation	51

6.2.1	Histogram	52
7	Level Set Implementation	55
7.1	The Base Active Contour Model	55
7.2	Proposed Active Contour Model	57
8	Unsupervised Multi-dimensional Histogram	59
8.1	Unsupervised Image Segmentation	59
8.2	Multi-dimensional Histogram Density Function	60
8.3	Contour Evolution	62
8.4	Algorithm	63
8.5	Experiments	66
9	Half-supervised Multivariate Gaussian Mixture	76
9.1	Multivariate Gaussian Mixture Density Function Estimated by EM	76
9.2	Half-supervised Image Segmentation	78
9.3	Algorithm	81
9.4	Experiments	85
10	Supervised Multivariate Gaussian Mixture	100
10.1	Supervised Image Segmentation	100
10.2	Multivariate Gaussian Mixture Density Function Estimated by EM with MML .	102
10.3	Algorithm	103
10.4	Experiments	109
11	Conclusions	120
Bibliography		123

List of Tables

1.1	Medical imaging scenario 1: an X-ray image of a hand. Segmentation and pattern classification as sequential and separate procedures.	3
1.2	Medical imaging scenario 2: an MR image of a brain. Segmentation and pattern classification as an integrated procedure.	3
2.1	Path searching in Canny edge detector	11
4.1	The criteria of general image segmentation	31
4.2	The criteria of region-based image segmentation	32
8.1	The input and output variables used in the unsupervised multi-dimensional histogram method	63
8.2	The input and output variables used in the multi-dimensional histogram density function	66
9.1	The input and output variables used in the half-supervised multivariate Gaussian mixture density method	81
9.2	The input and output variables used in the estimation of a mixture of multivariate Gaussian density functions using a classic EM method	84
10.1	The input and output variables used in the supervised active contour model using multivariate Gaussian mixture densities	103
10.2	The input and output variables used in the estimation of a mixture of multivariate Gaussian density functions using an advanced EM method with MML	106

List of Figures

1.1	A multispectral image $\mathbf{I}(x, y) : \Omega \rightarrow \Re^B$	5
1.2	An example of unimodal (solid) and multimodal (dotted) distributions	5
1.3	Examples of gray images with uniform and non-uniform sub-regions: (a) a triangle with uniform intensity, (b) a zebra with white and black strips	6
1.4	Examples of a multispectral image with uniform and non-uniform sub-regions: (a) a toy car painted gray (RGB), (b) a toy tank covered by a camouflage pattern (RGB)	7
2.1	Examples of gradient kernels along: (a) vertical direction, (b) horizontal direction	10
2.2	Sobel operators along: (a) vertical direction, (b) horizontal direction	11
2.3	Pixel aggregation: (a) original image with seeds underlined; (b) segmentation result with $\tau = 4$	12
3.1	An example of classic snakes	18
3.2	Level set evolution and the corresponding contour propagation: (a) topological view of level set $\phi(x, y)$ evolution, (b) the changes on the zero level set $C : \phi(x, y) = 0$	19
3.3	Initial contours and corresponding signed distance: (a) the initial contour C_0 , (b) the initial level set function $\phi_0(x, y)$ determined by the signed distance $\pm D((x, y), N_{x,y}(C_0))$	20
3.4	The change of topology observed in the evolution of level set function and the propagation of corresponding contours: (a) the topological view of level set $\phi(x, y)$ evolution, (b) the changes on the zero level set $C : \phi(x, y) = 0$	22
4.1	Subsets $\{\Omega_0, \Omega_1\}$ and sub-regions $\{\Psi_0, \Psi_1, \Psi_2\}$	33
5.1	A multimodal image and its ground truth: (a) a zebra, (b) the hand-segmented image	39
5.2	A multimodal distribution of image intensity and its representation using a unimodal Gaussian distribution: (a) zebra and (b) background of figure 5.1(a)	39
5.3	A unimodal distribution in a two-dimensional image intensity space and its reconstruction: (a) $p(\mathbf{I})$, (b) $g(\mathbf{I}) = p(I_1)p(I_2)$	41
5.4	A multimodal distribution in a two-dimensional image intensity space and its reconstruction: (a) $p(\mathbf{I}) = \alpha p_1(\mathbf{I}) + (1 - \alpha)p_2(\mathbf{I})$, (b) $g(\mathbf{I}) = \{\alpha p_1(I_1) + (1 - \alpha)p_2(I_1)\}\{\alpha p_1(I_2) + (1 - \alpha)p_2(I_2)\}$	42

6.1	The performance of EM algorithm according to the number of sub-classes assumed: (a) $K = 4$, (b) $K = 10$	48
6.2	The advanced EM method proposed by Figueiredo and Jain applied to the same data used in figure 6.1. The estimation starts with $K_{init} = 32$ and converges to $K = 4$.	51
6.3	An example of the histogram density function $h(\mathcal{I})$: (a) $\Delta\mathbf{I} = 1$, (b) $\Delta\mathbf{I} = 3$	53
7.1	Subsets and contours defined by two level set functions, $\{\phi_1, \phi_2\}$	56
8.1	Statistical distribution of image intensity: (a) a wood pattern, (b) the histogram	62
8.2	The iterative procedure of proposed active contour model: (top) contour evolution, (bottom) corresponding segments	67
8.3	Synthetic textured image: (a) a textured gray image, (b) the ground truth image	68
8.4	Method 1 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets	68
8.5	Method 2 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets	69
8.6	Proposed method applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets	69
8.7	Statistical distribution of image intensity within class 2, the small rectangle: (a) method 1, (b) method 2, (c) proposed method	70
8.8	Synthetic textured image: (a) a textured gray image, (b) the ground truth image	71
8.9	Method 1 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets	72
8.10	Method 2 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets	72
8.11	Proposed method applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets	73
8.12	An outdoor RGB image	73
8.13	Method 1 applied to an outdoor RGB image: (a) the final stage of contour evolution, (b) the segmented subsets	74
8.14	Method 2 applied to an outdoor RGB image: (a) the final stage of contour evolution, (b) the segmented subsets	75
8.15	Proposed method applied to an outdoor RGB image: (a) the final stage of contour evolution, (b) the segmented subsets	75
9.1	Density estimation using a supervised EM method	78
9.2	A complicated synthetic RGB image: (a) an RGB image with two camouflage patterns, (b) the ground truth image	79
9.3	Training stage of the proposed method: (a) 6 samples measured for background, (b) 3 samples measured for the rectangle	80
9.4	A complicated synthetic RGB image: (a) an RGB image with two camouflage patterns, (b) the ground truth image	86
9.5	Method 1 applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets	87

9.6	Method 2 applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets	87
9.7	Training stage of the proposed method: (a) 6 samples measured for background, (b) 3 samples measured for the core	88
9.8	Proposed method applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets	88
9.9	Statistical distribution of image intensity within the background measured at green channel: (a) method 1, (b) method 2, (c) proposed method	89
9.10	Statistical distribution of image intensity within the core measured at green channel: (a) method 1, (b) method 2, (c) proposed method	90
9.11	A complicated synthetic RGB image	91
9.12	Method 1 applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets	91
9.13	Method 2 applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets	92
9.14	Training stage of the proposed method: (a) 3 samples measured for background, (b) 6 samples measured for the tank	92
9.15	Proposed method applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets	93
9.16	A complicated outdoor gray image; two zebras	93
9.17	Method 1 applied to the zebra image: (a) the final stage of contour evolution, (b) the segmented subsets	94
9.18	Method 2 applied to the zebra image: (a) the final stage of contour evolution, (b) the segmented subsets	94
9.19	Training stage of the proposed method: (a) 2 samples measured for background, (b) 4 samples measured for zebras	95
9.20	Proposed method applied to the zebra image: (a) the final stage of contour evolution, (b) the segmented subsets	96
9.21	A complicated indoor RGB image; hand	96
9.22	Method 1 applied to the hand image: (a) the final stage of contour evolution, (b) the segmented subsets	97
9.23	Method 2 applied to the hand image: (a) the final stage of contour evolution, (b) the segmented subsets	97
9.24	Training stage of the proposed method: (a) 3 samples measured for hand, (b) 2 samples measured for the donut shaped object	98
9.25	Proposed method applied to the hand image: (a) the final stage of contour evolution, (b) the segmented subsets	99
10.1	Synthetic textured image: (a) a textured gray image, (b) the ground truth image	109
10.2	Method 1 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets	110
10.3	Method 2 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets	110
10.4	Training stage of the proposed method: (a) reference image for background, (b) reference image for the rectangle	111

10.5 Proposed method applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets	112
10.6 Statistical distribution of image intensity within background: (a) method 1, (b) method 2, (c) proposed method	112
10.7 Statistical distribution of image intensity within the small rectangle: (a) method 1, (b) method 2, (c) proposed method	113
10.8 A complicated outdoor image: (a) a zebra, (b) the ground truth image	114
10.9 Method 1 applied to a complicated outdoor image: (a) the final stage of contour evolution, (b) the segmented subsets	114
10.10 Method 2 applied to a complicated outdoor image: (a) the final stage of contour evolution, (b) the segmented subsets	115
10.11 Training stage of the proposed method: (a) training samples for background, (b) training samples for the zebra	116
10.12 Proposed method applied to a complicated outdoor image: (a) the final stage of contour evolution, (b) the segmented subsets	116
10.13 Complicated indoor RGB images: (a) reference image, (b) test 1, (c) test 2 . . .	117
10.14 Training stage of the proposed method: (a) training samples for background, (b) training samples for the tank	118
10.15 Proposed method applied to test 1 image: (a) the final stage of contour evolution, (b) the segmented subsets	118
10.16 Proposed method applied to test 2 image: (a) the final stage of contour evolution, (b) the segmented subsets	119

Chapter 1

Introduction

Vision is the most advanced sense among the five senses of human beings, and plays the most important role in human perception. Although the sensitivity of human vision is limited within the visible band, imaging machines can operate on the images generated by sources that human vision cannot associate with. Thus, *machine vision*¹ encompasses a wide and varied field of applications, even in areas where human vision cannot function, e.g. infrared (IR), ultraviolet (UV), X-ray, magnetic resonance imaging (MRI), ultrasound.

Although there is no clear distinction among image processing, image analysis, and computer vision, usually they are considered as hierarchies in the processing continuum. The low-level processing, which involves primitive operations such as noise filtering, contrast enhancement, and image sharpening, is considered as *image processing*. Note both its inputs and outputs are images. The mid-level processing, which involves segmentation and pattern classification, is considered as **image analysis** or image understanding [1]. Note its input generally are images, but its outputs are attributes extracted from those images, e.g. edges, contours, and the identity of individual objects², called *class*. The high-level processing, which involves ‘making sense’ of an ensemble of recognized objects and performing the cognitive functions at the far end of the processing continuum, is considered as *computer vision* [1]. We discuss the technologies used in the image analysis, and propose novel segmentation methods through this document.

¹Here, *machine vision* refers to any image processing technology not directly involved with human vision.

²In image analysis, an *object* usually refers to a matter or a body separated from the background in an image, while a *sub-region* refers to a set of pixels as a part of an image though these two terms are often interchangeably used.

1.1 Image Segmentation and Active Contours

In most image analysis operations, pattern classifiers require individual objects to be separated from the image, so the description of those objects can be transformed into a suitable form for computer processing. **Image segmentation** is a fundamental task, responsible for the separating operation. The function of segmentation is to partition an image into its constituent and disjoint sub-regions³, which are uniform according to their properties, e.g. intensity, color, and texture. Segmentation algorithms are generally based on either discontinuity among sub-regions, i.e. edges, or uniformity⁴ within a sub-region, though there are some segmentation algorithms relying on both discontinuity and uniformity.

The distinction between image segmentation and pattern classification is often not clear. The function of segmentation is simply to partition an image into multiple sub-regions, while the function of pattern classification is to identify the partitioned sub-regions. Thus, segmentation and pattern classification usually function as separate and sequential processes as shown in table 1.1. However, they might function as an integrated process as shown in table 1.2 depending on the image analysis problem and the performance of the segmentation method. In either way, segmentation critically affects the results of pattern classification, and often determines the eventual success or failure of the image analysis.

Since segmentation is an important task in image analysis, it is involved in most image analysis applications, particularly those related to pattern classification, e.g. medical imaging, remote sensing, security surveillance, military target detection. The level to which segmentation is carried depends on the problem being solved. That is, segmentation should stop when the region of interest (ROI) in the application have been isolated. Due to this property of problem dependence, autonomous segmentation is one of the most difficult tasks in image analysis. Noise and mixed pixels caused by the poor resolution of sensor images make the segmentation problem even more difficult. In this document, we propose novel segmentation methods using a variational framework, called active contours.

Active contours are connectivity-preserving relaxation [2] methods, applicable to the

³*Partitions, sub-regions, parts, sections, objects, and segments* are often interchangeably used. The term *sub-regions* will be consistently used in this document.

⁴The terms *uniformity* and *homogeneity* are often interchangeably used. The term *uniformity* will be consistently used in this document.

Table 1.1: Medical imaging scenario 1: an X-ray image of a hand. Segmentation and pattern classification as sequential and separate procedures.

Input data: an X-ray image of a hand <ol style="list-style-type: none"> 1. Segmentation: separate bones from the X-ray image. <ul style="list-style-type: none"> • Supervised method: trained features or sample data of bones are provided. • Unsupervised method: separate bright regions from the background. • Result: bones are extracted, but we do not know what kinds of bones they are. 2. Shape description: describe the extracted bones in a form of numerical features 3. Pattern classification: identify each bone based on the features
Output data: the identity of bones, e.g. thumb, index finger, ring finger, etc.

Table 1.2: Medical imaging scenario 2: an MR image of a brain. Segmentation and pattern classification as an integrated procedure.

Input data: an MR image of a brain <ol style="list-style-type: none"> 1. Segmentation & pattern classification: partition white and gray matters in the MR image. <ul style="list-style-type: none"> • Supervised: trained features or sample data of white and gray matters are provided. • Unsupervised: partition the brightest regions and brighter regions from the background.
Output data: extracted white and gray matter.

image segmentation problems. Active contours have been used for image segmentation and boundary tracking since the first introduction of *snakes* by Kass et al. [3]. The basic idea is to start with initial boundary shapes represented in a form of closed curves, i.e. contours, and iteratively modify them by applying shrink/expansion operations according to the constraints

of the image. Those shrink/expansion operations, called *contour evolution*, are performed by the minimization of an energy function like traditional region-based segmentation methods or by the simulation of a geometric partial differential equation (PDE) [4].

An advantage of active contours as image segmentation methods is that they partition an image into sub-regions with continuous boundaries, while the edge detectors based on threshold or local filtering, e.g. Canny [5] or Sobel operator, often result in discontinuous boundaries. The use of level set theory has provided more flexibility and convenience in the implementation of active contours. Depending on the implementation scheme, active contours can use various properties used for other segmentation methods such as edges, statistics, and texture. In this document, the proposed active contour models using the statistical information of image intensity within a sub-region.

1.2 Multispectral Images

A **multispectral image**⁵ is defined as a function on a two-dimensional spatial domain Ω , given by

$$\mathbf{I}(x, y) : \Omega \rightarrow \Re^B, \quad (1.1)$$

where the input of the function is a two-dimensional vector denoting the coordinates (x, y) , and the output is a vector-valued image intensity $\mathbf{I} \in \Re^B$. B denotes the dimension of \mathbf{I} , and is equivalent to the number of spectral bands. Figure 1.1 shows an example of a B -band image. In this document, we define a multispectral image as a general form of images and a scalar image as a particular case of multispectral images when $B = 1$. The most common example of multispectral images is an RGB image, consisting of three spectral bands: red, green, and blue. Hyperspectral images, used in remote sensing, are other examples of multispectral images. A set of images, measured by physically different sensors and registered, are also examples of multispectral images. As sensor fusion approaches become more popular in industrial and medical imaging, there will be more chances to encounter multispectral images in image analysis.

⁵Four terms, *multispectral images*, *multi-channel images*, *vector-valued images* [6], and *multi-valued images* [7], are often interchangeably used. Only the term *multispectral images* will be used in this document to avoid a confusion between *vector-valued images* and *vector (format) images* [8]. Every image function introduced in this document forms a *bitmap image*.

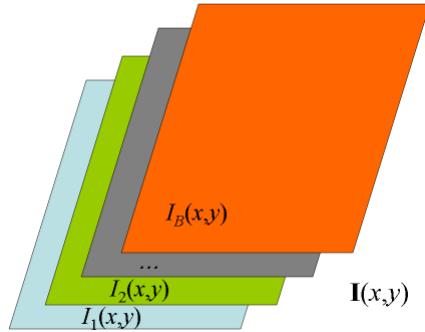


Figure 1.1: A multispectral image $\mathbf{I}(x, y): \Omega \rightarrow \Re^B$

In image processing, particularly medical image processing, *modality* refers to the type of input [9], such as the type of sensors, e.g. MRI, CT, or the bandwidth of spectrum. Thus, a *multimodal image* often refers to a multispectral image, where each channel is measured by different modalities. In statistical pattern classification, a statistical distribution consisting of, or representable by, multiple sub-classes⁶ is called *multimodal distribution*, while the other case is called *unimodal distribution*. Figure 1.2 shows examples of the two cases. The probability

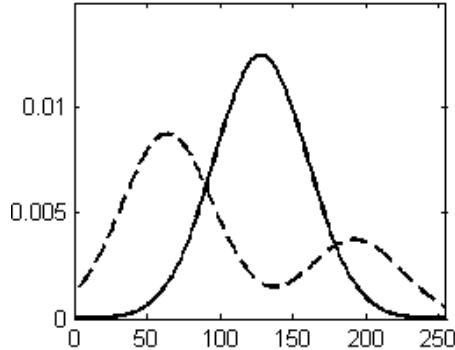


Figure 1.2: An example of unimodal (solid) and multimodal (dotted) distributions

density function (PDF) $p(I)$, presented as the solid curve, shows a unimodal distribution, while the mixture density function $p(I) = \alpha p_1(I) + (1 - \alpha)p_2(I)$, presented as the dotted curve, shows a multimodal distribution. A multi-dimensional statistical distribution, e.g. a two-dimensional Gaussian distribution, is called a *multivariate* distribution instead of a multimodal distribution. Note that the terminology of modality is different in image processing and statistical pattern

⁶Three terms *sub-classes*, *components*, and *modes* are often interchangeably used. The term *sub-classes* will be consistently used in this document.

classification. In this document, we use the terminology used in statistical pattern classification, so unimodal or multimodal refers to the statistical property of image intensity whether composed of a single class or multiple sub-classes, and univariate or multivariate refers to the dimensionality of image intensity such as a scalar image or a multispectral image.

1.3 Motivations

Although most image segmentation methods as well as active contours assume that each sub-region in the image has a uniform property, we often encounter images with non-uniform sub-regions. Figure 1.3 shows examples of an image with uniform and non-uniform sub-regions. The zebra shown in figure 1.3(b) consists of white and black strips, while the triangle shown



Figure 1.3: Examples of gray images with uniform and non-uniform sub-regions: (a) a triangle with uniform intensity, (b) a zebra with white and black strips

in figure 1.3(a) has a uniform image intensity. The statistical distribution of image intensity within the triangle and zebra would be similar to the graph shown in figure 1.2. Since the statistical distribution of image intensity within the zebra has at least two modes, i.e. one for black stripes and the other for white stripes, the segmentation method should be able to recognize a mixture of sub-classes as the class representing the zebra. Otherwise, the segmentation method would produce an over-segmentation result separating the white and black stripes or an under-segmentation result not separating the white stripes and the background. For this kind of problems, we propose advanced image segmentation methods using the statistical information of image intensity, where the statistical distribution of image intensity is represented as a **mixture density function**.

The image segmentation problem of images with non-uniform sub-regions becomes even more difficult when the image has multiple bands. Figure 1.4 shows examples of an RGB image with uniform and non-uniform sub-regions. The toy car shown in figure 1.4(a) is painted with



Figure 1.4: Examples of a multispectral image with uniform and non-uniform sub-regions: (a) a toy car painted gray (RGB), (b) a toy tank covered by a camouflage pattern (RGB)

uniform gray color, and the background is also uniform. The toy tank shown in figure 1.4(b) is painted with multiple different colors due to the camouflage pattern. The statistical distribution of vector-valued image intensity within the toy car has a single mode, while the statistical distribution of vector-valued image intensity within the toy tank has multiple modes in a multi-dimensional image intensity space. For this kinds of problems, we need to estimate the statistics of vector-valued image intensity as **a mixture of multivariate density functions**. As the estimation of a multivariate mixture density function is a tough problem and requires high computation, image segmentation using the multivariate mixture density functions is even a more difficult problem. We propose smart ways to deal with this problem.

Image segmentation has often been considered as a preprocessing of pattern classification as shown in table 1.1, but they are not necessarily separate procedures in the case of statistical pattern classification and region-based segmentation as shown in table 1.2. A few active contour models [10, 6, 11] have integrated those two procedures as an unsupervised segmentation, which partitions an image based on the statistics of image intensity within each subset. We propose two methods which **integrate image segmentation and statistical pattern classification** as a supervised segmentation, which partitions an image based on the image intensity at each

pixel and the statistical information of training samples. This integration reduces the processing time and helps to build an autonomous pattern recognition system.

The proposed active contour models are aimed at providing robust segmentation results for complicated image analysis, i.e. multispectral images with non-uniform sub-regions, but they are also applicable to any image segmentation problem. Possible applications are multi-sensor radiology in medical imaging, hyperspectral image segmentation in remote sensing, and color image segmentation.

Chapter 2

Image Segmentation: background

There are two main approaches in image segmentation: *edge-* and *region-* based. Edge-based segmentation partitions an image based on discontinuities among sub-regions, while region-based segmentation does the same function based on the uniformity of a desired property within a sub-region. In this chapter, we briefly discuss existing image segmentation technologies as background.

2.1 Edge-based Segmentation

Edge-based segmentation looks for discontinuities in the intensity of an image. It is more likely edge detection or boundary detection rather than the literal meaning of image segmentation, introduced in section 1.1. An *edge* can be defined as the boundary between two regions with relatively distinct properties. The assumption of edge-based segmentation is that every sub-region in an image is sufficiently uniform so that the transition between two sub-regions can be determined on the basis of discontinuities alone. When this assumption is not valid, region-based segmentation, discussed in the next section, usually provides more reasonable segmentation results.

Basically, the idea underlying most edge-detection techniques is the computation of a local derivative operator. The *gradient vector* of an image $I(x, y)$, given by

$$\nabla I = \begin{bmatrix} \partial I / \partial x \\ \partial I / \partial y \end{bmatrix} : \Omega \rightarrow \Re^2 , \quad (2.1)$$

is obtained by the partial derivatives $\partial I / \partial x$ and $\partial I / \partial y$ at every pixel location. The local derivative operation can be done by convolving an image with kernels shown in figure 2.1. The



Figure 2.1: Examples of gradient kernels along: (a) vertical direction, (b) horizontal direction

magnitude of the first derivative

$$|\nabla I| = \sqrt{(\partial I / \partial x)^2 + (\partial I / \partial y)^2} : \Omega \rightarrow \Re \quad (2.2)$$

determines the presence of edges in an image¹.

The *Laplacian* of an image function $I(x, y)$ is the sum of the second-order derivatives, defined as

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} : \Omega \rightarrow \Re. \quad (2.3)$$

The general use of the Laplacian is in finding the location of edges using its zero-crossings [12]. A critical disadvantage of the gradient operation is that the derivative enhances noise. As a second-order derivative, the Laplacian is even more sensitive to noise. An alternative is convolving an image with the Laplacian of a Gaussian (LoG) function [13], given by

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) : \Omega \rightarrow \Re, \quad (2.4)$$

where a two-dimensional Gaussian function with the standard deviation σ is defined as

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) : \Omega \rightarrow \Re. \quad (2.5)$$

The LoG function produces smooth edges as the Gaussian filtering provides smoothing effect [14].

Sobel operation is performed by convolving an image with kernels shown in figure 2.2. Sobel operators have the advantage of providing both a derivative and a smoothing effect [12, 15]. The smoothing effect is a particularly attractive feature of the Sobel operators compared to the

¹Although the literal meaning of the term *gradient* is the gradient vector ∇I , it often refers to the magnitude of gradient $|\nabla I|$ in many publications.

<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>-1</td><td>-2</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>1</td></tr> </table>	-1	-2	-1	0	0	0	1	2	1	(a)	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-2</td><td>0</td><td>2</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> </table>	-1	0	1	-2	0	2	-1	0	1	(b)
-1	-2	-1																			
0	0	0																			
1	2	1																			
-1	0	1																			
-2	0	2																			
-1	0	1																			

Figure 2.2: Sobel operators along: (a) vertical direction, (b) horizontal direction

gradient kernels shown in figure 2.1 because the derivative enhances noise.

Canny edge detector [16, 5] is based on the extrema of the first derivative of the Gaussian operator applied to an image. The operator first smoothes the image to eliminate noise, and then finds high gradient regions. After non-maximum suppression, the edges are finally determined by two thresholds, i.e. τ_{min} and τ_{max} as shown in table 2.1. Canny edge detector

Table 2.1: Path searching in Canny edge detector

- If $|\nabla I(x, y)| > \tau_{max}$, then $I(x, y)$ is an edge pixel.
- If $\tau_{min} < |\nabla I(x, y)| < \tau_{max}$,
 - If there is a path from (x, y) to neighbor (\mathbf{x}) and $|\nabla I(\mathbf{x})| > \tau_{min}$, then $I(x, y)$ is an edge pixel.
 - Otherwise, $I(x, y)$ is a non-edge pixel.
- If $|\nabla I(x, y)| < \tau_{min}$, then $I(x, y)$ is a non-edge pixel.

is known as an optimal edge detector because it satisfies the criteria of low error rate, good localization of edge points, and a single response to a single edge pixel [17].

Edge detection by gradient operations generally work well only in the images with sharp intensity transitions and relatively low noise. Due to its sensitivity to noise, some smoothing operation is generally required as preprocessing, and the smoothing effect consequently blurs the edge information. However, the computational cost is relatively lower than other segmentation methods because the computation can be done by a local filtering operation, i.e. convolution of an image with a kernel. Edge-based active contour models, discussed in section 3.3, use the magnitude of gradient $|\nabla I|$ to determine the position of edges.

2.2 Region-based Segmentation

Region-based segmentation looks for uniformity within a sub-region, based on a desired property, e.g. intensity, color, and texture. *Clustering* techniques encountered in pattern classification literature have similar objectives and can be applied for image segmentation [18].

Region growing [19] is a technique that merges pixels or small sub-regions into a larger sub-region. The simplest implementation of this approach is *pixel aggregation* [12], which starts with a set of seed points and grows regions from these seeds by appending neighboring pixels if they satisfy the given criteria. Figure 2.3 shows a simple example of pixel aggregation. Segmentation

2	4	8
3	5	<u>9</u>
4	6	7

(a)

2	2	9
2	2	9
2	9	9

(b)

Figure 2.3: Pixel aggregation: (a) original image with seeds underlined; (b) segmentation result with $\tau = 4$

starts with two initial seeds, and then the regions grow if they satisfy a criterion such as

$$|I(x, y) - I(\text{seed})| < \tau . \quad (2.6)$$

Despite the simple nature of the algorithm, there are fundamental problems in region growing: the selection of initial seeds and suitable properties to grow the regions. Selecting initial seeds can be often based on the nature of applications or images. For example, the ROI is generally brighter than the background in IR images. In this case, choosing bright pixels as initial seeds would be a proper choice.

Additional criteria that utilize properties to grow the regions lead region growing into more sophisticated methods, e.g. region competition. *Region competition* [20, 21] merges adjacent sub-regions under criteria involving the uniformity of regions or sharpness of boundaries. Strong criteria tend to produce over-segmented results, while weak criteria tend to produce poor segmentation results by over-merging the sub-regions with blurry boundaries. An alternative of region growing is *split-and-merge* [22], which partitions an image initially into a set of arbitrary, disjointed sub-regions, and then merge and/or split the sub-regions in an attempt to satisfy the segmentation criteria.

Another common approach in region-based segmentation is characterizing statistical uniformity of sub-regions using parametric models, so called *statistical estimation*. With this approach, two sub-regions are considered to be uniform, and consequently merged, if they can be represented by a single instance of the model, i.e. if they have common parameter values within a threshold. In practice, the parameters of a sub-region cannot be observed directly but can only be inferred from the observed data and the knowledge of the imaging process. In statistical approaches, this inference is often made using *Bayes's rule* [23] and the conditional PDF $p(\mathbf{I}(x, y)|\boldsymbol{\theta}_m)$, which presents the conditional probability that certain data $\mathbf{I}(x, y)$ (or statistics derived from the data) will be observed, given that sub-region m has the parameter values of $\boldsymbol{\theta}_m$. In typical statistical region merging algorithms [24], stochastic estimates in the parameter space are obtained for different sub-regions, and merging decisions are based on the similarity of these parameters.

A limitation of most estimation-based segmentation methods is that they do not explicitly represent the uncertainty in the estimated parameter values and, therefore, are prone to error when parameter estimates are poor. A *Bayesian probability of homogeneity* directly exploits all of the information contained in the statistical image models, instead of estimating parameter values [25]. The probability of homogeneity is based on the ability to formulate a prior probability density on the parameter space, and measures homogeneity by taking the expectation of the data likelihood over a posterior parameter space.

Image segmentation is often approached by *edge-preserving smoothing* operations as well as the partitioning operation. Edge-preserving smoothing techniques can be classified roughly two approaches [26]: Markov random field (MRF) including energy-based methods [27, 28] and diffusion-based methods [29, 30]. Both approaches show similar restoration characteristics because the diffusion-based methods can be viewed as an energy-based method that uses only the prior energy term at a given temperature [31]. Snyder et al. [32, 33, 34] proposed an edge-preserving smoothing method for image segmentation based on the technology called *mean field annealing* (MFA) [31, 35, 36, 37, 38, 39], and the same segmentation method was extended to vector-valued images by Han et al. [26, 40]. MFA is an energy-based method for finding the minimum of complex functions which typically have many minima [41]. For the image segmentation problem, a proper energy function is defined intending to keep the edges and to smooth the rest of areas in the image. The segmentation is performed by minimizing

the energy function using MFA. MFA approximates a stochastic algorithm called simulated annealing (SA) [42], which has shown to converge to the global minimum, even for non-convex problems [43]. Hiriyannaiah et al. [44] derived MFA using the analogy to physics for the restoration of piecewise-constant images, and Bilbro et al. [43] did the same job applying the MFA to the images with varying gray values.

Region-based approaches are generally less sensitive to noise, and usually produce more reasonable segmentation results as they rely on global properties rather than local properties, but their implementation complexity and computational cost can be often quite large. Statistical segmentation methods, both estimation-based and Bayesian-based, have been extended to many active contour models including the proposed models. Those active contour models based on statistical segmentation will be discussed in section 3.4.

2.3 Other Segmentation Methods

The *watershed* algorithm [47, 48] is a morphology-based segmentation method [49, 50, 51]. It is based on the assumption that any gray-tone image can be considered as a topographic surface [52]. If we flood this surface from its minima preventing the merge of the waters coming from different sources, the surface is eventually separated as two different sets: the catchment basins and the watershed lines. If we apply this transformation to the magnitude of image gradient $|\nabla I|$, the catchment basins correspond to the uniform sub-regions in the image and the watershed lines correspond to the edges. The flooding operation is simulated using morphological distance operators [53, 54, 55].

Fusions of different principles have produced good results. There have been a few approaches to integrate region- and edge-based segmentation [56, 57], and also an approach to integrate region- and morphology-based segmentation called watersnakes [58].

Texture is another feature that we can use to determine the segmentation criteria. Images can be considered as either a collection of pixels in the spatial domain or the sum of sinusoids of infinite extent in the spatial-frequency domain. Gabor observed that the spatial representation and the spatial-frequency representation are just opposite extremes of a continuum of possible joint space/spatial-frequency representations [59]. In a joint space/spatial-frequency represen-

tations for images, frequency is considered as a local phenomenon that can vary with position throughout the image. The human visual system is performing a form of local spatial-frequency analysis on the retinal image, and the analysis is done by a bank of bandpass filters [60].

The same approach can be used to partition textured images in image analysis. Perceptually significant texture differences presumably correspond to differences in the local spatial-frequency content using the space/spatial-frequency paradigm. Texture segmentation is done by two steps: decomposing an image into a joint space/spatial-frequency representation with a bank of bandpass filters and using this information to locate the regions of similar local spatial-frequency content. The response of the filter bank generates a kind of multispectral images, where each band represents the response of the textured image at a particular spatial-frequency bandwidth. The multi-channel filtering has been implemented by the convolution of the image with a stack of two-dimensional Gabor filters [61, 62, 63, 64, 65] or wavelets [66, 67].

Chapter 3

Active Contours: background

The technique of active contours has become quite popular for a variety of applications, particularly image segmentation and motion tracking, during the last decade. This methodology is based upon the utilization of deformable contours which conform to various object shapes and motions. This chapter provides a theoretical background of active contours and an overview of existing active contour methods.

There are two main approaches in active contours based on the mathematic implementation: *snakes* and *level sets*. Snakes explicitly move predefined snake points based on an energy minimization scheme, while level set approaches move contours implicitly as a particular level of a function. More details about these two approaches will be discussed respectively in section 3.1 and 3.2. As image segmentation methods, there are two kinds of active contour models according to the force evolving the contours: edge- and region-based. Edge-based active contours use an edge detector, usually based on the image gradient, to find the boundaries of sub-regions and to attract the contours to the detected boundaries. Edge-based approaches are closely related to the edge-based segmentation discussed in section 2.1. Region-based active contours use the statistical information of image intensity within each subset instead of searching geometrical boundaries. Region-based approaches are also closely related to the region-based segmentation discussed in section 2.2. More details of these two active contour models are respectively discussed in section 3.3 and section 3.4.

3.1 Snakes

The first model of active contour was proposed by Kass et al. [3] and named **snakes** due to the appearance of contour¹ evolution. Let us define a contour parameterized by arc length s as

$$C(s) \equiv \{(x(s), y(s)) : 0 \leq s \leq L\} : \mathfrak{R} \rightarrow \Omega, \quad (3.1)$$

where L denotes the length of the contour C , and Ω denotes the entire domain of an image $I(x, y)$. The corresponding expression in a discrete domain approximates the continuous expression as

$$C(s) \approx C(n) = \{(x(n), y(n)) : 0 \leq n \leq N, s = 0 + n\Delta s\}, \quad (3.2)$$

where $L = N\Delta s$. An energy function $E(C)$ can be defined on the contour such as

$$E(C) = E_{int} + E_{ext}, \quad (3.3)$$

where E_{int} and E_{ext} respectively denote the *internal energy* and *external energy* functions. The internal energy function determines the regularity, i.e. smooth shape, of the contour. A common choice for the internal energy is a quadratic functional given by

$$\begin{aligned} E_{int} &\equiv \int_0^L \alpha|C'(s)|^2 + \beta|C''(s)|^2 ds \\ &\approx \sum_{n=0}^N (\alpha|C'(n)|^2 + \beta|C''(n)|^2) \Delta s. \end{aligned} \quad (3.4)$$

Here α controls the tension of the contour, and β controls the rigidity of the contour. The external energy term determines the criteria of contour evolution depending on the image $I(x, y)$, and can be defined as

$$E_{ext} \equiv \int_0^L E_{img}(C(s)) ds \approx \sum_{n=0}^N E_{img}(C(n)) \Delta s, \quad (3.5)$$

where $E_{img}(x, y)$ denotes a scalar function defined on the image plane, so the local minimum of E_{img} attracts the snakes to edges. A common example of the edge attraction function is a function of image gradient, given by

$$E_{img}(x, y) = \frac{1}{\lambda|\nabla G_\sigma * I(x, y)|} : \Omega \rightarrow \mathfrak{R}, \quad (3.6)$$

¹Although snakes can be defined as opened curves, we are interested in only the case of closed curves, i.e. contours $C(0) = C(L)$, because our objective is image segmentation.

where G_σ denotes a Gaussian smoothing filter with the standard deviation σ , and λ is a suitably chosen constant. Solving the problem of snakes is to find the contour C that minimizes the total energy term E with the given set of weights α , β , and λ . In numerical experiments, a set of snake points residing on the image plane are defined in the initial stage, and then the next position of those snake points are determined by the local minimum E . The connected form of those snake points is considered as the contour. Figure 3.1 shows an example of classic snakes [69]. There are about 70 snakes points in the image, and the snake points form a contour

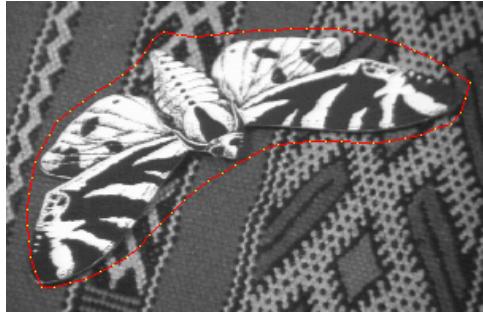


Figure 3.1: An example of classic snakes

around the moth. The snakes points are initially placed at further distance from the boundary of the object, i.e. the moth. Then, each point moves towards the optimum coordinates, where the energy function converges to the minimum. The snakes points eventually stop on the boundary of the object.

The classic snakes provide an accurate location of the edges only if the initial contour is given sufficiently near the edges because they make use of only the local information along the contour. Estimating a proper position of initial contours without prior knowledge is a difficult problem. Also, classic snakes cannot detect more than one boundary simultaneously because the snakes maintain the same topology during the evolution stage. That is, snakes cannot split to multiple boundaries or merge from multiple initial contours. Level set theory [4] has given a solution for this problem.

3.2 Level Set Methods

Level set theory, a formulation to implement active contours, was proposed by Osher and Sethian [4]. They represented a contour implicitly via a two-dimensional Lipschitz-continuous function $\phi(x, y) : \Omega \rightarrow \mathbb{R}$ defined on the image plane. The function $\phi(x, y)$ is called *level set function*, and a particular level, usually the zero level, of $\phi(x, y)$ is defined as the contour, such as

$$C \equiv \{(x, y) : \phi(x, y) = 0\}, \quad \forall (x, y) \in \Omega, \quad (3.7)$$

where Ω denotes the entire image plane. Figure 3.2(a) shows the evolution of level set function $\phi(x, y)$, and figure 3.2(b) shows the propagation of the corresponding contours C . As the level

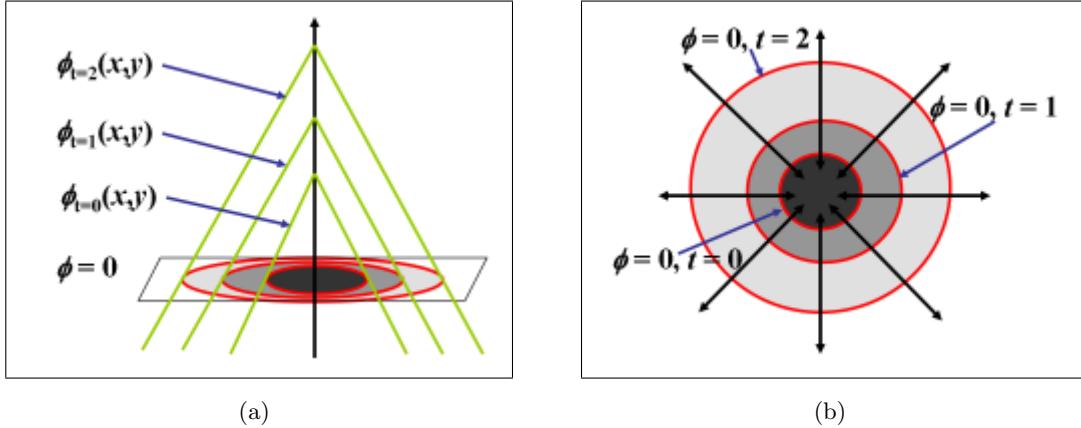


Figure 3.2: Level set evolution and the corresponding contour propagation: (a) topological view of level set $\phi(x, y)$ evolution, (b) the changes on the zero level set $C : \phi(x, y) = 0$

set function $\phi(x, y)$ increases from its initial stage, the corresponding set of contours C , i.e. the red contour, propagates toward outside.² With this definition, the evolution of the contour is equivalent to the evolution of the level set function, i.e. $\partial C / \partial t = \partial \phi(x, y) / \partial t$. The advantage of using the zero level is that a contour can be defined as the border between a positive area and a negative area, so the contours can be identified by just checking the sign of $\phi(x, y)$. The initial level set function $\phi_0(x, y) : \Omega \rightarrow \mathbb{R}$ may be given by the signed distance from the initial contour such as,

$$\begin{aligned} \phi_0(x, y) &\equiv \{\phi(x, y) : t = 0\} \\ &= \pm D((x, y), N_{x,y}(C_0)) \end{aligned}, \quad \forall (x, y) \in \Omega, \quad (3.8)$$

²In figure 3.2, the amount of level set evolution is set as a constant along the entire domain, $\partial \phi(x, y) / \partial t = c$, $\forall (x, y) \in \Omega$, for easy understanding. Normally, $\partial \phi(x, y) / \partial t$ is a function of spatial coordinates (x, y) .

where $\pm D(a, b)$ denotes a signed distance between a and b , and $N_{x,y}(C_0)$ denotes the nearest neighbor pixel on initial contours $C_0 \equiv C(t = 0)$ from (x, y) . Figure 3.3(a) shows an example of initial contours C_0 , and figure 3.3(b) shows the initial level set function $\phi_0(x, y)$ as the signed distance computed from the initial contour C_0 . $\phi_0(x, y)$ increases, i.e. become brighter,

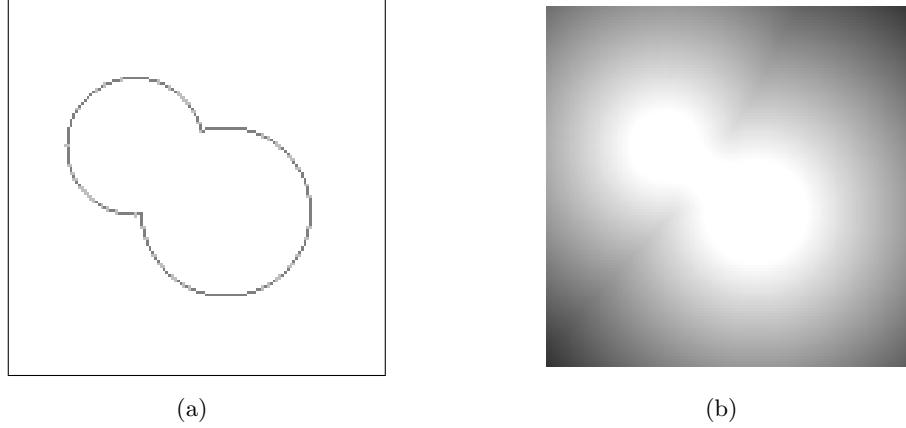


Figure 3.3: Initial contours and corresponding signed distance: (a) the initial contour C_0 , (b) the initial level set function $\phi_0(x, y)$ determined by the signed distance $\pm D((x, y), N_{x,y}(C_0))$

as a pixel (x, y) is located further inwards from the initial contours C_0 , while $\phi_0(x, y)$ decreases, i.e. become darker, as the pixel is located further outwards from the initial contours. The initial level set function is zero at the initial contour points given by, $\phi_0(x, y) = 0, \forall (x, y) \in C_0$.

The deformation of the contour is generally represented in a numerical form as a PDE. A formulation of contour evolution using the magnitude of the gradient of $\phi(x, y)$ was initially proposed by Osher and Sethian [71, 72, 4], given by

$$\frac{\partial \phi(x, y)}{\partial t} = |\nabla \phi(x, y)|(\nu + \varepsilon \kappa(\phi(x, y))), \quad (3.9)$$

where ν denotes a constant speed term to push or pull the contour, $\kappa(\cdot) : \Omega \rightarrow \mathbb{R}$ denotes the mean curvature of the level set function $\phi(x, y)$ given by

$$\begin{aligned} \kappa(\phi(x, y)) &= \operatorname{div} \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) \\ &= \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}, \end{aligned} \quad (3.10)$$

where ϕ_x and ϕ_{xx} denote the first- and second-order partial derivatives of $\phi(x, y)$ respect to x , and ϕ_y and ϕ_{yy} denote the same respect to y . The role of the curvature term is to control the

regularity of the contours as the internal energy term E_{int} does in the classic snakes model, and ε controls the balance between the regularity and robustness of the contour evolution.

Another form of contour evolution was proposed by Chan and Vese [10, 73]. The length of the contour $|C|$ can be approximated by a function of $\phi(x, y)$ [74, 75], such as

$$\begin{aligned} |C| \approx L_\epsilon(\phi(x, y)) &= \int_{\Omega} |\nabla H_\epsilon(\phi(x, y))| dx dy \\ &= \int_{\Omega} \delta_\epsilon(\phi(x, y)) |\nabla \phi(x, y)| dx dy, \end{aligned} \quad (3.11)$$

where $H_\epsilon(\cdot)$ denotes the regularized form of the unit step function³ $H(\cdot): \Omega \rightarrow \Re$ given by

$$H(x, y) = \begin{cases} 1, & \text{if } \phi(x, y) \geq 0 \\ 0, & \text{if } \phi(x, y) < 0 \end{cases}, \quad \forall (x, y) \in \Omega, \quad (3.12)$$

and $\delta_\epsilon(\cdot)$ denotes the derivative of $H_\epsilon(\cdot)$. Since the unit step function produces either 0 or 1 depending on the sign of the input, the derivative of the unit step function produces non-zero only where $\phi(x, y) = 0$, i.e. on the contour C . Consequently, the integration shown in equation 3.11 is equivalent to the length of contours on the image plane. The associated Euler-Lagrange equation [76] obtained by minimizing $L_\epsilon(\cdot)$ with respect to ϕ and parameterizing the descent directions by an artificial time t is given by

$$\frac{\partial \phi(x, y)}{\partial t} = \delta_\epsilon(\phi(x, y)) \kappa(\phi(x, y)). \quad (3.13)$$

The contour evolution motivated by the equation above can be interpreted as the motion by mean curvature minimizing the length of the contour. Therefore, equation 3.9 is considered as the motion motivated by PDE, while equation 3.13 is considered as the motion motivated by energy minimization.

An outstanding characteristic of level set methods is that contours can split or merge as the topology of the level set function changes. Therefore, level set methods can detect more than one boundary simultaneously, and multiple initial contours can be placed. Figure 3.4(a) shows an example of the topological changes on a level set function, while figure 3.4(b) shows how the initially separated contours merge as the topology of level set function varies. This flexibility and convenience provide a means for an autonomous segmentation by using a predefined set of initial contours. The computational cost of level set methods is high because the computation

³This unit step function is often referred to *Heaviside* function.

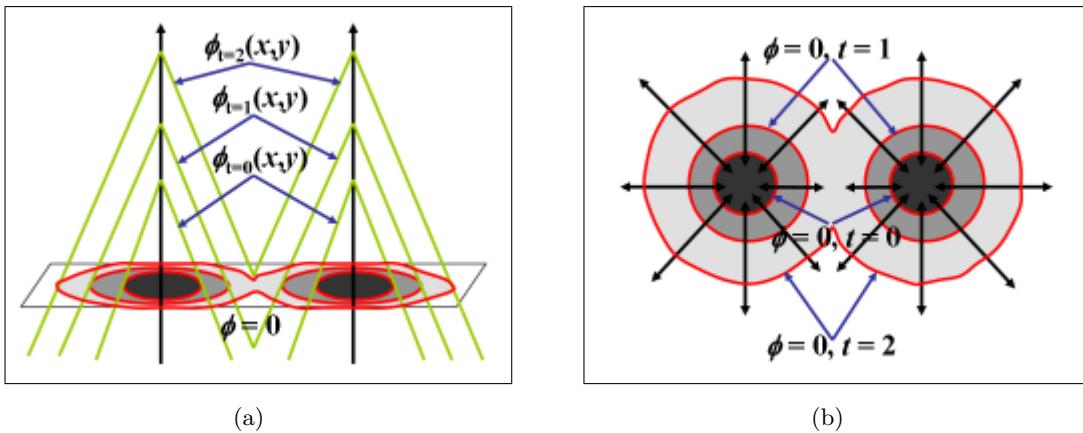


Figure 3.4: The change of topology observed in the evolution of level set function and the propagation of corresponding contours: (a) the topological view of level set $\phi(x, y)$ evolution, (b) the changes on the zero level set $C : \phi(x, y) = 0$

should be done on the same dimension as the image plane Ω . Thus, the convergence speed is relatively slower than other segmentation methods, particularly local filtering based methods. The high computational cost can be compensated by using multiple initial contours. The use of multiple initial contours increases the convergence speed by cooperating with neighbor contours quickly. Level set methods with faster convergence, called *fast marching methods* [77], have been studied intensively for the last decade. Because of these attractive properties, we implement the proposed active contour model using the level set method.

3.3 Edge-based Active Contours

Edge-based active contours are closely related to the edge-based segmentation. Most edge-based active contour models consist of two parts: the regularity part, which determines the shape of contours, and the edge detection part, which attracts the contour towards the edges.

Geometric active contour model was proposed by Caselles et al. [81] adding an additional term, called *stopping function*, to the speed function shown in equation 3.9. It was the first level set implemented active contour model for the image segmentation problem. Malladi et al. [82, 78] proposed a similar model given by

$$\frac{\partial \phi(x, y)}{\partial t} = g(I(x, y))(\kappa(\phi(x, y)) + \nu)|\nabla \phi(x, y)|, \quad (3.14)$$

where $g(\cdot) : \Omega \rightarrow \mathbb{R}$ denotes the stopping function, i.e. a positive and decreasing function of the image gradient. A simple example of the stopping function is given by

$$g(I(x, y)) = \frac{1}{1 + |\nabla I(x, y)|^n}, \quad (3.15)$$

where n is given as 1 in [81] and 2 in [82]. Note that $|\nabla I(x, y)|$ can be interchangeably used with E_{img} shown in equation 3.6. The contours move in the normal direction with a speed of $g(I(x, y))(\kappa(\phi(x, y)) + \nu)$, and therefore stops on the edges, where $g(\cdot)$ vanishes. The curvature term $\kappa(\cdot)$ maintains the regularity of the contours, while the constant term ν accelerates and keeps the contour evolution by minimizing the enclosed area [83].

Geodesic active contour model was proposed by Caselles et al. [84, 85] after the geometric active contour model. Kichenassamy et al. [86] and Yezzi et al. [87] also proposed a similar active contour model. Based on the principle of the classic dynamic systems, solving the active contour problem is equivalent to finding a path of minimal distance, called *geodesic curve* [88] given by

$$\frac{\partial C}{\partial t} = (g(I(x, y))\kappa(\phi(x, y)) - \nabla g(I(x, y)) \cdot \mathcal{N})\mathcal{N}, \quad (3.16)$$

where \mathcal{N} denotes the inward unit normal⁴ given by

$$\mathcal{N} = -\frac{\nabla\phi}{\|\nabla\phi\|}. \quad (3.17)$$

From the relation between a contour and a level set function and the level set formulation of the steepest descent method, solving this geodesic problem is equivalent to searching for the steady state of the level set evolution equation [84, 89] given by

$$\frac{\partial\phi(x, y)}{\partial t} = g(I(x, y))(\kappa(\phi(x, y)) + \nu)|\nabla\phi(x, y)| + \nabla g(I(x, y)) \cdot \nabla\phi(x, y), \quad (3.18)$$

where ν is an additional speed term to accelerate the evolution.⁵ The equivalence between classic snakes and geodesic active contours has been also shown by other authors [90, 91, 92, 93] in slightly different views. We can notice that the geodesic active contour model shown in equation 3.18 is identical to the geometric active contour model shown in equation 3.14 except for the advection term $\nabla g(I(x, y)) \cdot \nabla\phi(x, y)$. Geodesic active contours have been the most popular methods among the edge-based active contour models, and their applications have been extended to multispectral images by Sapiro.

⁴Some authors define \mathcal{N} as the outward unit normal with an opposite sign.

⁵The level set function still slowly converges even without ν .

Color snakes is the geodesic active contour model particularly for multispectral images proposed by Sapiro [88, 94, 95, 89]. In order to detect the edges in multispectral images, a special gradient function based on Riemannian geometry, called *color gradient* function, is used instead of the traditional image gradient function. A simple example of the color gradient function is given by

$$g(\mathbf{I}(x, y)) = \frac{1}{1 + (\lambda_+ - \lambda_-)}, \quad (3.19)$$

where λ_+ and λ_- respectively present the maximal and the minimal rate of changes on the multispectral image $\mathbf{I}(x, y)$. λ_+ and λ_- are the eigenvalues of the *metric tensor* [96] given by

$$\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \sum_b \begin{bmatrix} \left(\frac{\partial \mathbf{I}_b}{\partial x} \right)^2 & \frac{\partial \mathbf{I}_b}{\partial x} \frac{\partial \mathbf{I}_b}{\partial y} \\ \frac{\partial \mathbf{I}_b}{\partial y} \frac{\partial \mathbf{I}_b}{\partial x} & \left(\frac{\partial \mathbf{I}_b}{\partial y} \right)^2 \end{bmatrix}, \quad (3.20)$$

where I_b denotes the b -th band of the multispectral image $\mathbf{I}(x, y)$. In the case of scalar images, i.e. gray images, $\lambda_+ = |\nabla I|^2$ and $\lambda_- = 0$, so the stopping function shown in equation 3.19 become identical to equation 3.15. Sapiro also introduced *color self-snakes* [88, 97, 95, 89], which diffuse the image in the same way that we evolve the level set function. This is an edge-preserving smoothing method like MFA or image diffusion using active contours.

Due to the structure of the speed functions and the stopping functions in equation 3.9 and equation 3.18, edge-based active contour models have a few disadvantages compared to the region-based active contour models, discussed in the next section. Because of the constant term ν , edge-based active contour models evolve the contour towards only one direction, either inside or outside. Therefore, an initial contour should be placed completely inside or outside of ROI, and some level of a prior knowledge is still required. Also, edge-based active contours inherit some disadvantages of the edge-based segmentation methods due to the similar technique used. Since both edge-based segmentation and edge-based active contours rely on the image gradient operation, edge-based active contours may skip the blurry boundaries, and they are sensitive to local minima or noise as edge-based segmentation does. *Gradient vector flow fast geodesic active contours* [98, 99] proposed by Paragios replaced the edge detection (boundary attraction) term with *gradient vector field* [100, 101, 102, 103, 104], that refers to a spatial diffusion of the boundary information and guides the propagation to the object boundaries from both sides, to give more freedom from the restriction of initial contour position.

3.4 Region-based Active Contours

Most region-based active contour models consist of two parts: the regularity part, which determines the smooth shape of contours, and the energy minimization part, which searches for uniformity of a desired feature within a subset. A nice characteristic of region-based active contours is that the initial contours can be located anywhere in the image as region-based segmentation relies on the global energy minimization rather than local energy minimization. Therefore, less prior knowledge is required than edge-based active contours.

*Piecewise-constant active contour*⁶ model was proposed by Chan and Vese [10, 73] using the *Mumford-Shah segmentation* model [107, 108]. Piecewise-constant active contour model moves deformable contours minimizing an energy function instead of searching edges. A constant approximates the statistical information of image intensity within a subset, and a set of constants, i.e. a piecewise-constant, approximate the statistics of image intensity along the entire domain of an image. The energy function measures the difference between the piecewise-constant and the actual image intensity at every image pixel. The level set evolution equation is given by

$$\frac{\partial\phi(x,y)}{\partial t} = \delta_\epsilon(\phi(x,y))[\nu\kappa(\phi(x,y)) - \{(I(x,y) - \mu_1)^2 - (I(x,y) - \mu_0)^2\}], \quad (3.21)$$

where μ_0 and μ_1 respectively denote the mean of the image intensity within the two subsets, i.e. the outside and inside of contours. The final partitioned image can be represented as a set of piecewise-constants, where each subset is represented as a constant. This method has shown the fastest convergence speed among region-based active contours due to the simple representation. Lee et al. [109] showed an improvement to the piecewise-constant active contour model on illuminated images by proposing an alternative energy function.

Piecewise-smooth active contour model, an extension of piecewise-constant model using a set of smoothed partial images, was also proposed by Chan and Vese [110, 111, 112, 113, 114]. The same segmentation principles used for piecewise-constant model partitions an image, but a smoothed partial images instead of constants represent each subset. The level set evolution

⁶Although the titles of [10, 73] are “Active Contours without Edges,” all region-based active contours introduced in this section also do not use edge information. Thus, we refer to this method as *piecewise-constant active contour* model through this document.

equation is given by

$$\frac{\partial \phi(x, y)}{\partial t} = \delta_\epsilon(\phi(x, y)) \begin{bmatrix} \nu\kappa(\phi(x, y)) - \{(I(x, y) - \mu_1(x, y))^2 - (I(x, y) - \mu_0(x, y))^2\} \\ -\omega(|\nabla\mu_1(x, y)|^2 - |\nabla\mu_0(x, y)|^2) \end{bmatrix}, \quad (3.22)$$

where $\mu_0(x, y)$ and $\mu_1(x, y)$ respectively denote the smoothed images within the outside and inside of contours. The segmentation result of piecewise-smooth active contours is similar to the segmentation by color self-snakes because of the similar approach.

Although traditional region-based active contours partition an image into multiple sub-regions, those multiple regions belong to only two subsets: either the inside or the outside of contours. Chan and Vese proposed *multi-phase active contour* model⁷ [110, 76, 115, 116, 117], which increases the number of subsets⁸ that active contours can find simultaneously. Multiple active contours evolve independently based on the piecewise-constant model shown in equation 3.21 or the piecewise-smooth model shown in equation 3.22, and multiple subsets are defined by a group of disjoint combination of the level set functions. For example, N level set functions define maximum 2^N subsets of the entire region. An example of subsets defined by 4-phase active contours is

$$\begin{bmatrix} \Omega_0 \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \equiv \left\{ (x, y) : \begin{bmatrix} \phi_2(x, y) < 0, \phi_1(x, y) < 0 \\ \phi_2(x, y) < 0, \phi_1(x, y) > 0 \\ \phi_2(x, y) > 0, \phi_1(x, y) < 0 \\ \phi_2(x, y) > 0, \phi_1(x, y) > 0 \end{bmatrix} \right\}, \quad (3.23)$$

where $\{\Omega_0, \Omega_1, \Omega_2, \Omega_3\}$ denote the four subsets defined by two level set functions $\{\phi_1, \phi_2\}$,

⁷Here, the term *multi* refers to more than two instead of more than one. Two terms *phase* and *subset* are considered identical in the image segmentation problem, and used interchangeably in this document.

⁸A *sub-region* refers to a group of image pixels that have similar property and reside within the same boundaries. A *subset* refers to a group of the subregions that have similar property but does not necessarily reside within the same boundary.

i.e. two active contours. The level set evolution equation for this case is given by

$$\begin{aligned} \frac{\partial\phi_1(x,y)}{\partial t} &= \delta_\epsilon(\phi_1(x,y)) \left\{ \nu\kappa(\phi_1(x,y)) - \left[\begin{array}{l} \left\{ \begin{array}{l} (I(x,y) - \mu_3)^2 - \\ (I(x,y) - \mu_2)^2 \end{array} \right\} H_2+ \\ \left\{ \begin{array}{l} (I(x,y) - \mu_1)^2 - \\ (I(x,y) - \mu_0)^2 \end{array} \right\} (1 - H_2) \end{array} \right] \right\} \\ \frac{\partial\phi_2(x,y)}{\partial t} &= \delta_\epsilon(\phi_2(x,y)) \left\{ \nu\kappa(\phi_2(x,y)) - \left[\begin{array}{l} \left\{ \begin{array}{l} (I(x,y) - \mu_3)^2 - \\ (I(x,y) - \mu_1)^2 \end{array} \right\} H_1+ \\ \left\{ \begin{array}{l} (I(x,y) - \mu_2)^2 - \\ (I(x,y) - \mu_0)^2 \end{array} \right\} (1 - H_1) \end{array} \right] \right\} \end{aligned} \quad (3.24)$$

where $H_n \equiv H_\epsilon(\phi_n(x,y))$ and $\{\mu_0, \mu_1, \mu_2, \mu_3\}$ denote the mean of image intensity within the corresponding subsets $\{\Omega_0, \Omega_1, \Omega_2, \Omega_3\}$. Rousson and Deriche [118, 11, 119] and Yezzi et al. [120, 121] proposed a similar multi-phase active contour model for segmentation problems, and Samson et al. [122, 123, 124] also proposed a multi-phase active contour model for pattern classification problems. Multi-phase active contours provide a means to integrate segmentation and pattern classification tasks. m -phase active contours partition the image into multiple sub-regions ($\gg m$), and they simultaneously identify those regions into m -subsets, i.e. *classes*. Depending whether training samples are provided or not, supervised or unsupervised segmentation can actually perform supervised or unsupervised pattern classification. This provides a way to the autonomous pattern classification technology reducing the number of procedures and processing time.

The same segmentation principle can be extended to multispectral images by taking the mean of energy functions measured at each band [6, 125]. The level set evolution equation of 2-phase active contour model is given by

$$\frac{\partial\phi(x,y)}{\partial t} = \delta_\epsilon(\phi(x,y))[\nu\kappa(\phi(x,y)) - \sum_b^B \{\omega_{1b}(\mathbf{I}_b(x,y) - \mu_{1b})^2 - \omega_{0b}(\mathbf{I}_b(x,y) - \mu_{0b})^2\}], \quad (3.25)$$

where $\boldsymbol{\mu}_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{ib}, \dots, \mu_{iB}]^T$ denotes the mean vector of the vector valued image intensity $\mathbf{I}(x,y)$ within the corresponding subset Ω_i .

Rousson and Deriche proposed a variational formulation obtained from a Bayesian segmentation model [118, 11, 119]. While the piecewise-constant active contour model uses a group of constants to represent subsets, this method implicitly uses a conditional PDF of a given value

$\mathbf{I}(x, y)$ with respect to the hypothesis, i.e. a unimodal (multivariate) Gaussian distribution, given by

$$p(\mathbf{I}(x, y) | \{\boldsymbol{\mu}_i, \Sigma_i\}) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{I}(x, y) - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{I}(x, y) - \boldsymbol{\mu}_i)\right), \quad (3.26)$$

where $\boldsymbol{\mu}_i$ and Σ_i are the parameters of a (multivariate) Gaussian distribution. Taking the negative log form of the PDF as an energy function, the level set evolution equation minimizing the energy function is given by

$$\frac{\partial \phi(x, y)}{\partial t} = \delta_\epsilon(\phi(x, y)) [\nu \kappa(\phi(x, y)) - \{e_1(x, y) - e_0(x, y)\}], \quad (3.27)$$

where the objective functions e_i are given by

$$\begin{aligned} e_i(x, y) &\equiv -\log p(\mathbf{I}(x, y) | \boldsymbol{\mu}_i, \Sigma_i) \\ &\simeq \log |\Sigma_i| + (\mathbf{I}(x, y) - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{I}(x, y) - \boldsymbol{\mu}_i). \end{aligned} \quad (3.28)$$

The extension of this method for multispectral images is trivial as the energy function is based on the multivariate PDF. A unique feature of this method is that multivariate PDF consider each band of multispectral images as independent dimension of an image intensity space. Multivariate PDF does not make big improvement in this active contour model, but it does make a significant improvement in the proposed active contour models using a mixture density function. More detail of the improvement will be discussed in chapter 5.2.

Due to the global energy minimization, region-based active contours generally do not have any restriction on the placement of initial contours. That is, region-based active contour can detect interior boundaries regardless of the position of initial contours. The use of pre-defined initial contours provides a method of autonomous segmentation. Also, they are less sensitive to local minima or noise than edge-based active contours. However, due to the assumption of uniform image intensity, most methods are applicable only to images where each subset is representable by a simple expression, e.g. single Gaussian distribution or a constant. If a subset, i.e. class, consists of multiple distinctive sub-classes, e.g. the zebra shown in figure 1.3(b), these methods would produce over-segmented or under-segmented results. We propose novel region-based active contour models which produce improved results using multivariate mixture density functions.

3.5 Active Contours Integrating Edge- and Region-based Segmentation

In order to improve the segmentation performance, the integration of edge- and region-based information sources using active contours has been proposed by a few authors. *Geodesic active region* is a supervised active contour model, proposed by Paragios [126, 127, 128], integrating edge- and region-based segmentation module in an energy function. A statistical analysis based on the Minimum Description Length (MDL) criterion and the Maximum Likelihood (ML) principle for the observed density function, i.e. an image histogram, indicates the number of sub-regions and the statistical PDF within those sub-regions using a mixture of Gaussian elements. Regional probability is estimated from the statistical PDF based on prior knowledge, i.e. training samples. Then, the boundary information is determined by a *probabilistic edge detector*, estimated from the regional probabilities of neighborhood [129, 130]. For example, an image pixel is more likely an edge pixel if the neighborhood pixels, located on the opposite sides, have high regional probabilities for a different class. The level set evolution equation is given by

$$\frac{\partial \phi(x, y)}{\partial t} = |\nabla \phi(x, y)| \left[\begin{array}{l} \alpha \sum_b \omega_b \log \left(\frac{p_1(I_b(x, y))}{p_2(I_b(x, y))} \right) + \\ (1 - \alpha) \left\{ g(p_B(x, y)) \kappa(\phi(x, y)) + \nabla g(p_e(x, y)) \cdot \frac{\nabla \phi(x, y)}{|\nabla \phi(x, y)|} \right\} \end{array} \right] \quad (3.29)$$

where $I_b(x, y)$ denotes b -th band of a multispectral image $\mathbf{I}(x, y)$, and $p_i(I_b(x, y))$ denotes the regional probability presenting a probability that a pixel $I_b(x, y)$ belongs to sub-region i , $p_e(x, y)$ denotes the probabilistic edge detector presenting a probability that a boundary pixel is located at (x, y) , and $g(p_e)$ denotes a positive and decreasing function of the probability. The regional probability is computed from each band and accumulated. A detail to determine both probabilities, $p_i(I_b(x, y))$ and $p_e(x, y)$, is explained in [130]. The geodesic active region model is later applied to a medical imaging problem [131, 132] with a gradient vector flow-based boundary component. The approach was based on a coupled propagation of two active contours, and integrates visual information with anatomical constraints.

Jehan-Besson et al. also proposed an active contour model [133, 134] minimizing an energy criterion involving both region and boundary functionals. These functionals are derived through a shape derivative approach instead of classical calculus of variation. They focus on statistical property, i.e. the PDF of the color histogram of a sub-region. Active contours are propagated minimizing the distance between two histograms for matching or tracking purposes.

Chapter 4

Region-based Segmentation using Active Contours

Region-based segmentation looks for uniformity within a sub-region based on a desired feature, e.g. intensity, color, and texture. Region-based active contour models have shown attractive characteristics, such as the unrestricted position of initial contours, the automatic detection of interior boundaries, and reasonable segmentation due to global energy minimization though the segmentation results are still case dependent. Region-based active contours evolve deformable shapes based on two forces: energy minimization based on the statistical properties, which pursues the uniformity within each subset, and curvature motion motivated by level set function, which keeps the regularity of active contours. In this chapter, we propose a novel segmentation principle based on regional energy minimization using the statistics of image intensity. First, we redefine the terminology used in the image segmentation problem in section 4.1. Then, we discuss the base segmentation model, proposed by Mumford-Shah, of the proposed active contour model in section 4.2. Finally, we discuss the proposed segmentation principle and its implementation in section 4.3.

4.1 Image, Subset, and Sub-region

Let us redefine the notation of terms used in our segmentation model.¹ As introduced in chapter 1, an image $\mathbf{I}(x, y)$ is the native input data of the image analysis, given as a form of

¹These notations are not necessarily identical to the terms introduced in other publications.

a function defined on a two-dimensional spatial domain. We define a multispectral image as a general form of images and a scalar image as a particular case of multispectral images. A multispectral image $\mathbf{I}(x, y)$ can be defined as a set of vectors given by

$$\mathbf{I}(x, y) \equiv [I_1(x, y), I_2(x, y), \dots, I_b(x, y), \dots, I_B(x, y)]^T : \Omega \rightarrow \Re^B, \quad (4.1)$$

where $I_b(x, y)$ denotes a scalar image measured at band b . Let the vector-valued image intensity of $\mathbf{I}(x, y)$ be a multi-dimensional random variable $\mathbf{I} \in \Re^B$ where B denotes the dimension of \mathbf{I} and is equivalent to the number of optical bands measured.

Let Ψ represent the entire region of an image $\mathbf{I}(x, y)$. Image segmentation is a task to partition the entire region Ψ into n sub-regions, $\{\Psi_1, \Psi_2, \dots, \Psi_i, \dots, \Psi_n\}$, with the criteria shown in table 4.1. C_i denotes the boundary wrapping sub-region Ψ_i . The first and second condi-

Table 4.1: The criteria of general image segmentation

- | |
|--|
| <ol style="list-style-type: none"> 1. $C = \bigcup_{i=1}^n C_i$. 2. $C_i \cap C_j \neq \emptyset$ if Ψ_i and Ψ_j are neighbors. 3. $\Psi = \left(\bigcup_{i=1}^n \Psi_i \right) \cup C$. 4. $\mathbf{I}(x, y)$ are connected, $\forall (x, y) \in \Psi_i, \forall i$. 5. $\Psi_i \cap \Psi_j = \emptyset, \forall i, j, \text{ if } i \neq j$. |
|--|

tions indicate the property of boundaries wrapping sub-regions $\{\Psi_i\}$. As each sub-region has a boundary, the boundaries of two neighbor sub-regions are overlapped. C denotes the entire set of boundaries.² The third condition indicates that the segmentation must be complete; that is, every image pixel should be an element of a sub-region Ψ_i or boundaries C . The fourth condition requires that all image pixels in a sub-region must be connected in a predefined sense; that is, they should be located at the inside of a boundary. The fifth condition indicates that the sub-regions must be disjoint each other, so an image pixel should be an element of only one sub-region. Here, we can notice the difference between the image segmentation problem and the pattern classification problem. A data sample can be a member of multiple classes in pattern classification, but an image pixel should be a member of only one sub-region in image

²The width of a boundary is often considered as infinitely small in continuous expression.

segmentation.

Table 4.1 lists the criteria of general image segmentation, and we here introduce slightly different criteria for region-based segmentation. Let a *set* Ω , instead of a *region* Ψ , represent the entire domain of an image $\mathbf{I}(x, y)$. The region-based image segmentation is a task to partition the entire set Ω of an image into m subsets, $\{\Omega_1, \Omega_2, \dots, \Omega_i, \dots, \Omega_m\}$, with the criteria shown in table 4.2. The only difference between a subset Ω_i and a sub-region Ψ_j is that a subset Ω_i does

Table 4.2: The criteria of region-based image segmentation

- | |
|---|
| <ol style="list-style-type: none"> 1. $C = \bigcup_{i=1}^m C_i$. 2. $C_i \cap C_j \neq \emptyset$ if Ω_i and Ω_j are neighbors. 3. $\Omega = \left(\bigcup_{i=1}^m \Omega_i \right) \cup C$. 4. $\Omega_i \cap \Omega_j = \emptyset$ for $\forall i, j$ if $i \neq j$. |
|---|

1. $C = \bigcup_{i=1}^m C_i$.
2. $C_i \cap C_j \neq \emptyset$ if Ω_i and Ω_j are neighbors.
3. $\Omega = \left(\bigcup_{i=1}^m \Omega_i \right) \cup C$.
4. $\Omega_i \cap \Omega_j = \emptyset$ for $\forall i, j$ if $i \neq j$.

not necessarily form a spatial unit. That is, Ω_i may contain multiple sub-regions Ψ_j residing in different spatial locations on the entire set Ω of an image. Following expression shows the relation between n subsets and m sub-regions:

$$(\Omega = \Psi) \supseteq \Omega_i \supseteq \Psi_j \supseteq (x, y), \quad (4.2)$$

where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$.³ $(\Omega = \Psi)$ denotes the entire set of an image as the largest possible spatial unit, while (x, y) denotes an image pixel as the smallest possible spatial unit. Figure 4.1 shows an example that sub-regions and subsets are not identical. The entire set of the image $(\Omega = \Psi)$ consists of two subsets $\{\Omega_0, \Omega_1\}$ and three sub-regions $\{\Psi_0, \Psi_1, \Psi_2\}$. A subset Ω_1 exists at two different spatial locations, where each of them is independently marked as Ψ_1 and Ψ_2 . Therefore, two main approaches of segmentation, i.e. edge- and region-based, can be reintroduced such that edge-based segmentation partitions an image $\mathbf{I}(x, y)$ into multiple sub-regions Ψ_j searching for discontinuities among sub-regions, while region-based segmentation partitions an image $\mathbf{I}(x, y)$ into multiple subsets Ω_i searching for uniformity within a subset Ω_i .

³According to the definition, $m \leq n$ in general, but it can be $m \ll n$ depending on the image.

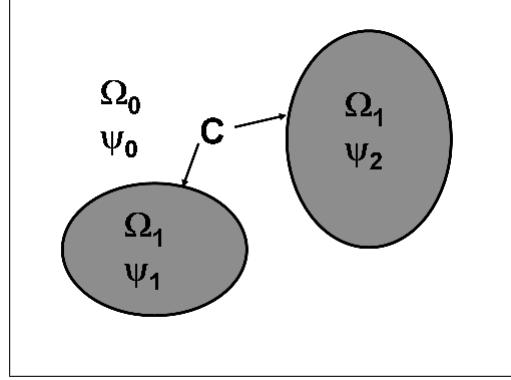


Figure 4.1: Subsets $\{\Omega_0, \Omega_1\}$ and sub-regions $\{\Psi_0, \Psi_1, \Psi_2\}$

4.2 The Base Segmentation Model

Mumford and Shah [107, 108] posed the image segmentation problem as a variational problem to find an optimal piecewise-smooth approximation $f(x, y)$ of the given scalar image $I(x, y)$ and a set of boundaries C , such that the approximation $f(x, y)$ varies smoothly within the connected components of the subsets excluding the boundaries, i.e. $\Omega \setminus C$. They proposed to solve the variational segmentation problem by minimizing the following global energy function [76]

$$E^{MS}(f, C) \equiv \int_{\Omega} |I(x, y) - f(x, y)|^2 dx dy + \mu \int_{\Omega \setminus C} |\nabla f(x, y)|^2 dx dy + \nu |C|, \quad (4.3)$$

with respect to two terms, the approximation f of the given image and the variational boundaries C . The global energy function E^{MS} consists of three parts. The minimization of the first part approximates the image $I(x, y)$ with an alternative expression $f(x, y)$ by minimizing the squared difference between the two expressions $|I(x, y) - f(x, y)|^2$. The second part piecewisely smoothes $f(x, y)$ by minimizing $|\nabla f(x, y)|^2$ on $\Omega \setminus C$. C has the role of approximating the edges of $I(x, y)$. The third part smoothes C by minimizing⁴ the length $|C|$. The existence and regularity of the solution of the problem above is proven in [138, 107].

The global piecewise approximation $f(x, y)$ can be represented as a sum of sub-approximations given by

$$f(x, y) \equiv \sum_i f_i \chi_i(x, y), \quad \forall (x, y) \in \Omega, \quad (4.4)$$

where f_i approximates the given multispectral image $\mathbf{I}(x, y)$ within Ω_i , and $\chi_i(x, y) : \Omega \rightarrow \{0, 1\}$

⁴In the original publication [110, 107, 108], the integration respect to one-dimensional Hausdorff measure was used instead of $|C|$. The equivalence between two expressions $|C| = \int_C d\mathcal{H}^1$ is shown in [76].

is determined by the spatial domain of Ω_i such as

$$\chi_i(x, y) = \begin{cases} 1, & \forall(x, y) \in \Omega_i \\ 0, & \forall(x, y) \notin \Omega_i \end{cases}, \quad \forall i. \quad (4.5)$$

Note that f_i is not necessarily an image function but any expression that represents the feature, used as the region-based segmentation criteria, of the image within Ω_i .

The global energy function $E^{MS}(f, C)$ given in equation 4.3 can be simplified and generalized by ignoring the smoothing term and defining independent objective functions for each subset, such that

$$E(\{f_i\}, C) \equiv \sum_i \int_{\Omega_i} e(x, y|f_i) dx dy + \nu |C|, \quad (4.6)$$

where the variational contour C determines the domain of variational subsets Ω_i . The objective function $e(x, y|f_i) : \Omega \rightarrow \mathbb{R}$ determines the condition of region-based segmentation within each Ω_i , e.g. the uniformity of image intensity \mathbf{I} , and how well the approximation f_i represents the given image. Better f_i results in lower e for each Ω_i , consequently lower E . The minimum of E is achieved by two sequential minimization procedures. First, the minimization of E with respect to each of f_i , while C is fixed, finds the best representations of each Ω_i of the given image $\mathbf{I}(x, y)$ minimizing the objective function $e(x, y|f_i)$. Then, the minimization of E with respect to C , while $\{f_i\}$ are fixed, smoothes the variational boundaries C minimizing $|C|$. Combination of these two minimizations leads to the region-based active contour evolution, which moves a set of contours C satisfying the segmentation constraints on the given image.

Depending on the objective function $e(x, y|f_i)$ and representation f_i , various active contour models can be achieved from the global energy function shown in equation 4.6. The energy function of piecewise-constant active contour model [6] can be transformed to equation 4.6 with an objective function

$$e(x, y|f_i \equiv \boldsymbol{\mu}_i) = \frac{1}{B} \sum_{b=1}^B (I_b(x, y) - \mu_{i,b})^2, \quad \forall(x, y) \in \Omega_i, \quad \forall i, \quad (4.7)$$

where f_i is given as a vector $\boldsymbol{\mu}_i \equiv [\mu_1, \dots, \mu_b, \dots, \mu_B]^T \in \mathbb{R}^B$. The optimal f_i minimizing E is the mean vector of $\mathbf{I}(x, y)$ within Ω_i . The energy function of [11] also can be transformed to the same form with an objective function

$$e(x, y|f_i \equiv \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}) = \log |\boldsymbol{\Sigma}_i| + (\mathbf{I}(x, y) - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{I}(x, y) - \boldsymbol{\mu}_i), \quad \forall(x, y) \in \Omega_i, \quad \forall i, \quad (4.8)$$

where f_i is given as a set of parameters, i.e. the mean vector μ_i and covariance matrix Σ_i of a multivariate Gaussian PDF. The optimal f_i minimizing E is the mean vector μ and covariance matrix Σ of $\mathbf{I}(x, y)$ within Ω_i .

4.3 Proposed Region-based Segmentation Model

The objective function of a piecewise-constant active contour model shown in equation 4.7 measures the average Euclidean distance between the vector-valued image intensity $\mathbf{I}(x, y)$ and the mean vector μ_i of $\mathbf{I}(x, y)$ within a subset Ω_i . The objective function shown in equation 4.8 measures the Mahalanobis distance [23] between $\mathbf{I}(x, y)$ and μ_i weighted by the covariance Σ_i . Thus, both objective functions produce the minimum if $\mathbf{I}(x, y) = \mu_i$, and produce larger values as $\mathbf{I}(x, y)$ and μ_i are located further from each other in the image intensity space. The assumption of these approaches is that there is a constant value, i.e. $f_i = \mu_i$ or $f_i = \{\mu_i, \Sigma_i\}$, which best represents Ω_i , and consequently the entire image is representable as a piecewise-constant expression f . However, many images we encounter in image segmentation problems, particularly the images with non-uniform subsets such as the zebra in figure 1.3(b), are too complicated to be represented as a piecewise-constant expression. Therefore, we propose an objective function, which uses a conditional PDF of $\mathbf{I}(x, y)$ instead of a distance between $\mathbf{I}(x, y)$ and $\{f_i\}$. With the proposed objective function, there is no certain best representation f_i but a PDF representation p_i for each Ω_i . Depending on the method to estimate the PDF, there are two approaches, unsupervised and supervised. Unsupervised method partitions images looking for uniform statistics of image intensity within a subset, while supervised method does the same job looking for similarity between the statistics of image intensity and the corresponding training samples for a subset.

If an image pixel is likely be an element of a subset, the objective function should produce a lower value under the assumption of $\mathbf{I}(x, y) \in \Omega_i$. If an image pixel is unlikely be an element of the subset, the objective function should produce higher value under the same assumption. The objective function

$$e(x, y|p_i) = -\log p_i, \quad \forall(x, y) \in \Omega, \quad \forall i, \quad (4.9)$$

provides the desired feature where p_i denotes an **unsupervised** multivariate conditional PDF

of vector-valued image intensity \mathbf{I} on the condition that $\mathbf{I}(x, y)$ is an element of Ω_i , given by

$$p_i \equiv p(\mathbf{I}(x, y)|(x, y) \in \Omega_i), \quad \forall i. \quad (4.10)$$

If p_i is fixed as a Gaussian distribution, equation 4.9 is equivalent to equation 4.8. Note that the proposed objective function may have multiple local maxima, while the objective functions in equation 4.7 and 4.8 have only a global minimum. The corresponding global energy function is given by

$$E(\{p_i\}, C) \equiv \sum_i \int_{\Omega_i} e(x, y|p_i) dx dy + \nu |C|. \quad (4.11)$$

The minimum of E is achieved by a similar way to minimize equation 4.6. First, instead of minimizing E with respect to each f_i , we estimate the conditional probability p_i from the given image, which is equivalent to finding the best fitting f_i . The estimated p_i provides a force moving the active contour C to the proper position where C divides the entire domain Ω of the image $\mathbf{I}(x, y)$ into multiple subsets $\{\Omega_i\}$, i.e. satisfying the segmentation constraints. Then, the minimization of E with respect to C , while $\{p_i\}$ are fixed, smoothes the variational boundaries C minimizing $|C|$.

If some level of prior knowledge, i.e. training samples, of an object is available, we can estimate the conditional PDF $p(\mathbf{I}|\Omega_i)$ directly from the training samples, such that

$$p(\mathbf{I}|\Omega_i) \approx \hat{p}(\mathbf{I}|\Omega_i) = p(\mathbf{I}|\mathcal{I}_i), \quad (4.12)$$

where $\mathcal{I}_i = \{\mathbf{I}_1, \dots, \mathbf{I}_n, \dots, \mathbf{I}_N\}$ denotes the ideally sampled training data or a statistical template of the object, which is supposed to be isolated as a subset Ω_i in the result of the segmentation. $p(\mathbf{I}|\mathcal{I}_i) : \Re^B \rightarrow \Re$ denotes a multivariate conditional PDF of a vector-valued image intensity \mathbf{I} on the condition that the multispectral image pixel is an element of training samples \mathcal{I}_i . An objective function similar to equation 4.9 can be used with a **supervised** conditional PDF given by

$$e_i(x, y) = -\log p(\mathbf{I}(x, y)|\mathbf{I}(x, y) \in \mathcal{I}_i), \quad \forall (x, y) \in \Omega, \quad \forall i, \quad (4.13)$$

where $e_i(x, y)$ measures the similarity between the statistics of \mathbf{I} and \mathcal{I}_i . Since the best fitting PDF is given from training samples instead of the given image, the minimization of E with respect to p_i is not necessary in supervised methods, and the global energy function shown in equation 4.11 can be simplified as a function of C only, such as

$$E(C) \equiv \sum_i \int_{\Omega_i} e_i(x, y) dx dy + \nu |C|. \quad (4.14)$$

This will simplify the segmentation procedure, consequently reducing the convergence time of active contours. Since the best fitting $p(\mathbf{I}|\Omega_i)$ is given, the segmentation result will be also more robust than the result of unsupervised methods.

Various expressions can be used for $p(\mathbf{I}|\Omega_i)$ as long as they satisfy the unit volume condition:

$$\int_{\mathbf{I}} p(\mathbf{I}|\Omega_i) = 1, \quad \text{or} \quad \sum_{\mathbf{I}} p(\mathbf{I}|\Omega_i) = 1, \quad \forall i. \quad (4.15)$$

For example, either a parametric continuous PDF, e.g. a Gaussian density function $p(\mathbf{I}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, or a non-parametric discrete PDF can be used as long as they are integrated as one. The details of proposed model for $p(\mathbf{I}|\Omega_i)$ will be discussed in chapter 5, and the estimation methods will be discussed in chapter 6.

Chapter 5

Probability Density Model

The objective of region-based segmentation is to partition the entire set Ω of an image into multiple disjoint subsets $\{\Omega_i\}$ based on the uniformity of a desired feature within each Ω_i . Unfortunately, image intensity within a subset Ω_i is not always uniform in practical applications. Subsets often form a mixture of multiple sub-classes, e.g. the zebra shown in figure 1.3(b) or the camouflage pattern on the toy tank shown in figure 1.4(b). In this chapter, we propose a multivariate mixture density function as the statistical model of vector-valued image intensity used for image segmentation. Section 5.1 discusses the need and use of a mixture density function as the statistical model of scalar image intensity. Section 5.2 extends the use of mixture density function to the case of multispectral images comparing the multivariate mixture density function and the product of marginal density functions which has been commonly used in other segmentation methods.

5.1 Mixture Density Function

The assumption of segmentation models introduced in section 4.2 is that the statistical information of image intensity within each subset is representable by a simple expression f_i , such as a vector-valued constant μ_i as shown in equation 4.7 or a small number of parameters $\{\mu_i, \Sigma_i\}$ as shown in equation 4.8. However, the statistical distribution of image intensity of a subset is often unrepresentable in a simple form. Figure 5.1 shows an example of an image with non-uniform subsets and its hand-segmented result. Figure 5.1(a) shows a result of diffusion applied on figure 1.3(b). The background is quite uniform, and the statistical distribution (histogram) of image intensity within the background forms a unimodal density function as shown



Figure 5.1: A multimodal image and its ground truth: (a) a zebra, (b) the hand-segmented image

in figure 5.2(b). The actual histogram of the background presented as the solid line can be rea-

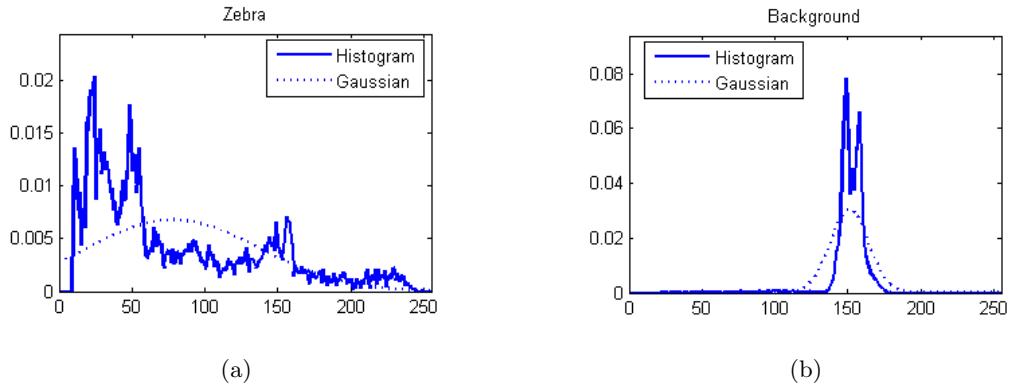


Figure 5.2: A multimodal distribution of image intensity and its representation using a unimodal Gaussian distribution: (a) zebra and (b) background of figure 5.1(a)

sonably represented with a Gaussian density function presented as the dotted line. However, the zebra consists of black and white stripes, and the statistical distribution of image intensity within the zebra is not unimodal as shown in figure 5.2(a). The higher peaks located about $I \approx 30$ and $I \approx 50$ in figure 5.2(a) present the black stripes of zebra, and the relatively lower peaks located about $I \approx 100$ and $I \approx 150$ present the white stripes of zebra. Thus, a Gaussian density function cannot simply represent the actual statistical distribution (histogram) of image intensity within the zebra though it can represent the background.

We propose to use **a mixture of probability density functions** as the feature representing the statistical property of image intensity I within a subset Ω_i , such that the conditional

PDF of I on the condition $I(x, y) \in \Omega_i$ is given by

$$p(I|\Omega_i) \equiv p_i(I) = \sum_{k=1}^K \alpha_k p_i(I|k), \quad \forall i, \quad (5.1)$$

where K denotes the number of sub-classes of the mixture, and $p_i(I|k) : \Re \rightarrow \Re$ denotes the conditional PDF of I on the condition that $I(x, y)$ is an element of Ω_i and $I(x, y)$ is generated by the sub-class k . The weights $\{\alpha_k\}$ of each sub-class, often called mixing probabilities, satisfy

$$\sum_{k=1}^K \alpha_k = 1. \quad (5.2)$$

Using a mixture density function, the histogram shown in figure 5.2(a) can be represented as a mixture of at least two simple parametric density functions.

5.2 Multivariate Mixture Density Function

We proposed to use a mixture density function for scalar images in section 5.1. How about multispectral images? The mixture density function of scalar image intensity I shown in equation 5.1 is a complicated statistical model, and a mixture density function of vector-valued image intensity \mathbf{I} is an even more complicated statistical model. For example, the histogram of 24bit RGB image intensity requires $2^{24} = 16777216$ histogram bins, while the histogram of 8bit gray image requires 256 histogram bins. Because of the complexity and the high computational cost, the active contour models employing mixture density functions [139, 126, 140, 141] as well as traditional region-based active contour models [142, 6] have used alternative expressions rather than dealing with multi-dimensional image intensity space. Based on the assumption that each band is independent, a common alternative expression of $p(\mathbf{I})$ is the product of *marginal* density functions measured at each dimension, i.e. spectral bands, such as

$$g(\mathbf{I}) \equiv \prod_{b=1}^B p(I_b). \quad (5.3)$$

If $p(\mathbf{I})$ is a mixture density function, the equivalent expression can be expressed as

$$g(\mathbf{I}) \equiv \prod_{b=1}^B \sum_{k=1}^K \alpha_k p(I_b|k). \quad (5.4)$$

Using this expression, the proposed objective functions shown in equation 4.9 or 4.13 are given by

$$\begin{aligned} e_i(x, y) &\equiv -\log \prod_b \sum_{k=1}^K \alpha_k p_i(I_b(x, y)|k), \quad \forall(x, y) \in \Omega, \quad \forall i, \\ &= -\sum_b \log \sum_{k=1}^K \alpha_k p_i(I_b(x, y)|k). \end{aligned} \quad (5.5)$$

The computation of $g(\mathbf{I})$ is relatively easier than $p(\mathbf{I})$ because the dimensionality of $p(\mathbf{I})$ is as low as $\Re \rightarrow \Re$ even though $g(\mathbf{I})$ and $p(\mathbf{I})$ have the same dimensionality as $\Re^B \rightarrow \Re$. The expressions similar to $g(\mathbf{I})$ and $\log g(\mathbf{I})$ have been used in a few other active contour models designed for multispectral images [6, 142], and we had previously proposed a similar expression [141] using the product of normalized histograms measured at each band. This is not a bad idea because the statistical meaning of $p(\mathbf{I})$ and $g(\mathbf{I})$ are equivalent as long as $p(\mathbf{I})$ is a unimodal distribution as shown in figure 5.3. Figure 5.3(a) presents a PDF of a two-dimensional random

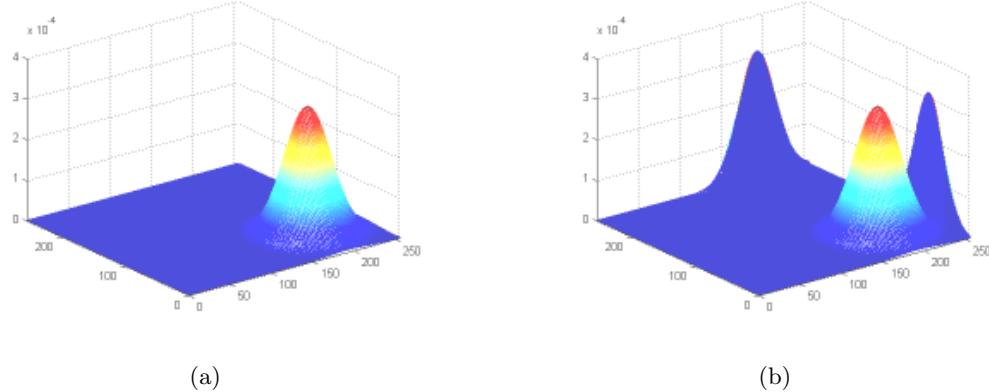


Figure 5.3: A unimodal distribution in a two-dimensional image intensity space and its reconstruction: (a) $p(\mathbf{I})$, (b) $g(\mathbf{I}) = p(I_1)p(I_2)$

variable $\mathbf{I} \in \Re^2$, which is distributed by a two-dimensional Gaussian density function $p(\mathbf{I}|\boldsymbol{\mu}, \Sigma)$. In figure 5.3(b), the shades on the walls present the marginal density functions of \mathbf{I} , respectively given by $p(I_1)$ and $p(I_2)$. The three-dimensional curve presents the product of $p(I_b)$ given by $g(\mathbf{I}) = p(I_1)p(I_2)$. As the two expressions $p(\mathbf{I})$ and $g(\mathbf{I})$ are equivalent, we can alternatively use $g(\mathbf{I})$ for $p(\mathbf{I})$ in this case.

Unfortunately, $g(\mathbf{I})$ for $p(\mathbf{I})$ are not equivalent any more if $p(\mathbf{I})$ is a mixture density function as shown in figure 5.4. Figure 5.4(a) presents a PDF of $\mathbf{I} \in \Re^2$, distributed by a mixture of

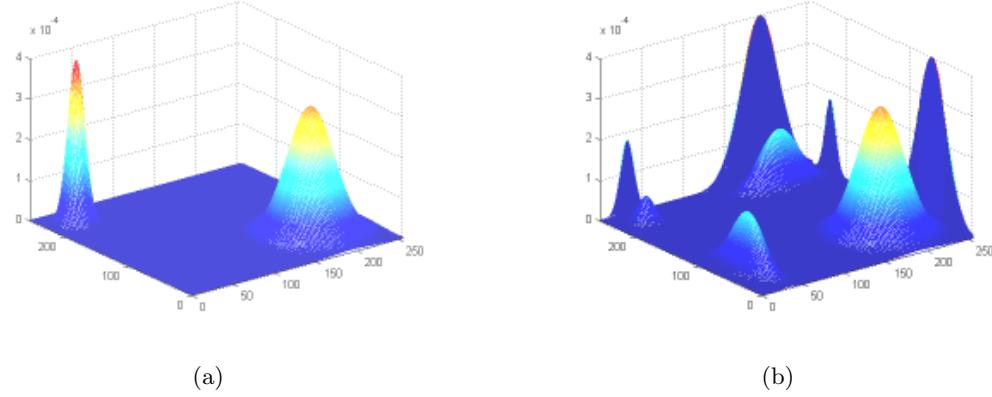


Figure 5.4: A multimodal distribution in a two-dimensional image intensity space and its reconstruction: (a) $p(\mathbf{I}) = \alpha p_1(\mathbf{I}) + (1 - \alpha)p_2(\mathbf{I})$, (b) $g(\mathbf{I}) = \{\alpha p_1(I_1) + (1 - \alpha)p_2(I_1)\}\{\alpha p_1(I_2) + (1 - \alpha)p_2(I_2)\}$

two different two-dimensional Gaussian functions $p(\mathbf{I}) = \alpha p(\mathbf{I}|\boldsymbol{\mu}_1, \Sigma_1) + (1 - \alpha)p(\mathbf{I}|\boldsymbol{\mu}_2, \Sigma_2)$. The shades on the walls present $p(I_1) = \alpha p_1(I_1) + (1 - \alpha)p_2(I_1)$ and $p(I_2) = \alpha p_1(I_2) + (1 - \alpha)p_2(I_2)$, and the three-dimensional curve presents $g(\mathbf{I}) = p(I_1)p(I_2) = \{\alpha p_1(I_1) + (1 - \alpha)p_2(I_1)\}\{\alpha p_1(I_2) + (1 - \alpha)p_2(I_2)\}$. As we can see in figure 5.4(a) and 5.4(b), the product of marginal density functions measured at each dimension, i.e. $g(\mathbf{I})$, is not equivalent to the true PDF $p(\mathbf{I})$ any more. Although there are only two modes in the true PDF $p(\mathbf{I})$, there are four modes in $g(\mathbf{I})$. The two excessive modes likely result in classifying the out-class image pixels as the given class, equivalent to the *false alarm* [143] in pattern classification. Since $\int g(\mathbf{I})$ is fixed as 1, the two excessive modes result in reducing the power of the two correct modes in $g(\mathbf{I})$. This incident likely results in excluding the in-class image pixels from the given class, equivalent to the *misclassification* [143] in pattern classification. Therefore, $g(\mathbf{I})$, i.e. the alternative expression of $p(\mathbf{I})$, is not a proper choice for the multispectral images with non-uniform subsets though it is a reasonable choice for the multispectral images with uniform subsets. Therefore, we propose to use **a mixture of multivariate probability density functions** as the statistical information of vector-valued image intensity within a subset, such that

$$\begin{aligned} p_i(\mathbf{I}) &\equiv p(\mathbf{I}|\Omega_i) \\ &= \sum_{k=1}^K \alpha_k p_i(\mathbf{I}|k), \quad \forall i, \end{aligned} \tag{5.6}$$

where $p_i(\mathbf{I}|k)$ denotes the conditional PDF of \mathbf{I} on the condition that $\mathbf{I}(x, y) \in \Omega_i$ and \mathbf{I} is generated by a sub-class k . The corresponding objective function equivalent to equation 4.9

using the proposed density model is given by

$$e(x, y|p_i) = -\log \sum_{k=1}^K \alpha_k p(\mathbf{I}(x, y)|(x, y) \in \Omega_i, k), \quad \forall (x, y) \in \Omega, \quad \forall i, \quad (5.7)$$

and the corresponding objective function equivalent to equation 4.13 using the proposed density model is given by

$$e_i(x, y) = -\log \sum_{k=1}^K \alpha_k p(\mathbf{I}(x, y)|\mathbf{I}(x, y) \in \mathcal{I}_i, k), \quad \forall (x, y) \in \Omega, \quad \forall i. \quad (5.8)$$

Chapter 6

Density Estimation Methods

We have proposed to use a multivariate mixture density function as the conditional PDF of vector-valued image intensity $p_i(\mathbf{I})$. In this chapter, we discuss the estimation methods of multivariate mixture density functions as a topic of statistical pattern recognition. Estimating a mixture density function, often called *mixture fitting*, has been one of the most difficult tasks in statistical pattern recognition. A multi-dimensional feature space makes the estimation problem even more difficult.

Let N data samples \mathcal{I} be given as $\mathcal{I} \equiv \{\mathbf{I}(1), \mathbf{I}(2), \dots, \mathbf{I}(n), \dots, \mathbf{I}(N)\}$, where $\mathbf{I} \in \Re^B$ denotes a multi-dimensional random variable. The PDF of a random variable can be estimated from the data samples \mathcal{I} if the statistical distribution of the data samples approximates the true PDF $p(\mathbf{I})$, such as

$$p(\mathbf{I}) \approx p(\mathcal{I}). \quad (6.1)$$

The base assumption of this approximation is that each data sample $\mathbf{I}(n)$ is generated independently and ideally from the true PDF $p(\mathbf{I})$.

There are two approaches to estimate $p(\mathbf{I})$ based on the ways to represent a mixture density function: parametric and non-parametric. Parametric approaches consider $p(\mathbf{I})$ as a mixture of multiple continuous parametric density functions, while non-parametric approaches consider $p(\mathbf{I})$ as a discrete function. Section 6.1 discusses parametric density estimation methods, and section 6.2 discusses non-parametric density estimation.

6.1 Parametric Density Estimation Methods

In the parametric approaches, a collection of parameters

$$\boldsymbol{\Theta} = \begin{bmatrix} \{\alpha_1, \dots, \alpha_k, \dots, \alpha_K\}, \\ \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_k, \dots, \boldsymbol{\theta}_K\} \end{bmatrix} \quad (6.2)$$

represent a mixture density function $p(\mathbf{I})$, where each parameter set $\boldsymbol{\theta}_k$ specifies the conditional PDF of a vector-valued image intensity \mathbf{I} on the condition that \mathbf{I} is generated by sub-class k , and α_k specifies the relative weight of the corresponding sub-class. The mixture density function $p(\mathbf{I})$ can be approximated by estimating a parameter set $\boldsymbol{\Theta}$ from data samples \mathcal{I} , such that

$$p(\mathbf{I}) \equiv p(\mathbf{I}|\boldsymbol{\Theta}) \approx p(\mathcal{I}|\hat{\boldsymbol{\Theta}}) = \sum_{k=1}^K \hat{\alpha}_k p(\mathcal{I}|\hat{\boldsymbol{\theta}}_k), \quad (6.3)$$

where each $\boldsymbol{\theta}_k$ is estimated based on a particular stochastic model such as a Gaussian distribution $p(\mathbf{I}|\boldsymbol{\mu}, \Sigma)$. As a parametric density function, we propose to use **a mixture of multivariate Gaussian distributions** for $p(\mathbf{I}|\boldsymbol{\Theta})$, such that the estimated PDF of a sub-class is given by

$$p(\mathcal{I}|\hat{\boldsymbol{\theta}}_k) \equiv p(\mathcal{I}|\hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k). \quad (6.4)$$

The stochastic model above is an arbitrary random distribution whose only prior knowledge is that the statistics of each sub-class k is representable as a weighted multivariate Gaussian distribution $\alpha_k p(\mathbf{I}|\boldsymbol{\mu}_k, \Sigma_k)$. There are four sets of parameters to be estimated: the number of sub-classes K , the weight of each sub-class $\{\alpha_k\}$, the mean vector of each sub-class $\{\boldsymbol{\mu}_k\}$, and the covariance matrix of each sub-class $\{\Sigma_k\}$.

Based on the assumption that each data sample $\mathbf{I}(n)$ is generated independently, the likelihood measured from the data samples \mathcal{I} is given by

$$p(\mathcal{I}|\boldsymbol{\Theta}) = \prod_{n=1}^N p(\mathbf{I}(n)|\boldsymbol{\Theta}), \quad (6.5)$$

and the log-likelihood of a K-component mixture model is given from equation 6.3 and 6.5 as

$$\log p(\mathcal{I}|\boldsymbol{\Theta}) = \log \prod_{n=1}^N p(\mathbf{I}(n)|\boldsymbol{\Theta}) = \sum_{n=1}^N \log \sum_{k=1}^K \alpha_k p(\mathbf{I}(n)|\boldsymbol{\theta}_k). \quad (6.6)$$

It is well known that the maximum likelihood (ML) estimate [143, 145]

$$\hat{\boldsymbol{\Theta}}_{ML} = \arg \max_{\boldsymbol{\Theta}} \{\log p(\mathcal{I}|\boldsymbol{\Theta})\} \quad (6.7)$$

or the Bayesian maximum *a posteriori* (MAP)

$$\hat{\Theta}_{MAP} = \arg \max_{\Theta} \{ \log p(\mathcal{I}|\Theta) + \log p(\Theta) \} \quad (6.8)$$

cannot be found analytically for this mixture density estimation problem [146, 147]. Therefore, an iterative learning algorithm, e.g. the EM algorithm, has been recommended [143] as an attractive solution.

6.1.1 EM Algorithm

The *expectation-maximization* (**EM**) methods [148, 149, 143, 150, 151] have been popularly used to obtain the ML or MAP estimates of the mixture parameters. The EM algorithm is an iterative procedure which finds local maxima of $\log p(\mathcal{I}|\Theta)$ or $\log p(\mathcal{I}|\Theta) + \log p(\Theta)$ interpreting the observed samples \mathcal{I} as *incomplete data*. For the mixture density model, the *missing* part is given as a set of labels

$$\mathcal{Z} = \{\mathbf{z}(1), \mathbf{z}(2), \dots, \mathbf{z}(n), \dots, \mathbf{z}(N)\}, \quad (6.9)$$

where each label is a multi-dimensional binary vector

$$\mathbf{z}(n) = [z_1(n), z_2(n), \dots, z_k(n), \dots, z_K(n)]^T \in \{0, 1\}^K \quad (6.10)$$

indicating which sub-class produced the observed sample $\mathbf{I}(n)$. That is, $z_k(n) = 1$ if and only if $\mathbf{I}(n)$ is generated by the sub-class k . Otherwise, $z_k(n) = 0$. The complete log-likelihood, i.e. the one from which we could estimate Θ if the complete data $\mathcal{X} = \{\mathcal{I}, \mathcal{Z}\}$ was observed [151], is given by

$$\log p(\mathcal{I}, \mathcal{Z}|\Theta) = \sum_{n=1}^N \sum_{k=1}^K z_k(n) \log[\alpha_k p(\mathbf{I}(n)|\theta_k)]. \quad (6.11)$$

A time-varied estimate $\hat{\Theta}(t)$ is produced by iteratively applying two sequential steps, i.e. expectation and maximization, until the given convergence criterion is met.

In the expectation stage, often called *E-step*, the conditional expectation of the complete log-likelihood is computed given \mathcal{I} and the current estimate $\hat{\Theta}(t)$. Since the elements of \mathcal{Z} are binary, their conditional expectation is given by

$$\mathcal{W} \equiv E[\mathcal{Z}|\mathcal{I}, \hat{\Theta}(t)] \in \Re^K, \quad (6.12)$$

where each element is given by

$$\begin{aligned}
w_k(n) &\equiv E[z_k(n)|\mathcal{I}, \hat{\Theta}(t)] \\
&= Pr[z_k(n) = 1|\mathbf{I}(n), \hat{\Theta}(t)] \\
&= \frac{\hat{\alpha}_k(t)p(\mathbf{I}(n)|\hat{\theta}_k(t))}{\sum_{j=1}^K \hat{\alpha}_j(t)p(\mathbf{I}(n)|\hat{\theta}_j(t))},
\end{aligned} \tag{6.13}$$

with the unit volume condition

$$\sum_{k=1}^K w_k(n) = 1. \tag{6.14}$$

Equation 6.13 may be interpreted as an instance of Bayes law, i.e. $\hat{\alpha}_k(t)$ and $w_k(n)$ are respectively equivalent to the *a priori* probability and the *a posteriori* probability of $z_k(n) = 1$ after observing $\mathbf{I}(n)$. Since $\log p(\mathcal{I}, \mathcal{Z}|\Theta)$ is linear with respect to the missing \mathcal{Z} , the conditional expectation of the complete log-likelihood is given by substituting $\mathcal{W} \equiv E[\mathcal{Z}|\mathcal{I}, \hat{\Theta}(t)]$ into $\log p(\mathcal{I}, \mathcal{Z}|\Theta)$, such as

$$\begin{aligned}
Q(\Theta, \hat{\Theta}(t)) &\equiv E \left[\log p(\mathcal{I}, \mathcal{Z}|\Theta) | \mathcal{I}, \hat{\Theta}(t) \right] \\
&= \log p(\mathcal{I}, \mathcal{W}|\Theta) \\
&= \sum_{n=1}^N \sum_{k=1}^K w_k(n) \log[\alpha_k p(\mathbf{I}(n)|\theta_k)].
\end{aligned} \tag{6.15}$$

The estimated parameter set $\hat{\Theta}$ is the one maximizing the likelihood function above.

In the maximization stage, often called *M-step*, the estimated parameters are updated according to

$$\hat{\Theta}(t+1) = \arg \max_{\Theta} \{Q(\Theta, \hat{\Theta}(t)) + \log p(\Theta)\}, \tag{6.16}$$

in the case of MAP estimation, or

$$\hat{\Theta}(t+1) = \arg \max_{\Theta} \{Q(\Theta, \hat{\Theta}(t))\}, \tag{6.17}$$

for the ML criterion.

Despite its popular use, the EM algorithm for mixture density estimation has several drawbacks: First, the number of sub-classes K should be provided as a prior knowledge. This requirement limits the use of EM algorithm for unsupervised learning. Second, K is fixed during the estimation process, so the EM algorithm may converge to the boundary of the parameter space. For example, the weight α_k may become zero and consequently the corresponding

parameters specifying the sub-class k become singular if K is significantly higher than the unknown true number of sub-classes. Third, it is a local (greedy) method, thus sensitive to initial values of parameters $\hat{\theta}_k(t = 0)$. If $\hat{\theta}_k(t = 0)$ is too far way from the true parameter θ_k , the EM algorithm may not converge to the true or optimal parameters because the EM algorithm cannot move sub-classes across low likelihood regions¹ [146, 152]. Common solutions for these problems are: using multiple random starts and choosing the final estimates with the highest likelihood [153, 150, 151, 154] or initialization by clustering [153, 150, 151]. Figure 6.1 shows the results of the standard EM algorithm using different K . 1000 random samples \mathcal{I} were

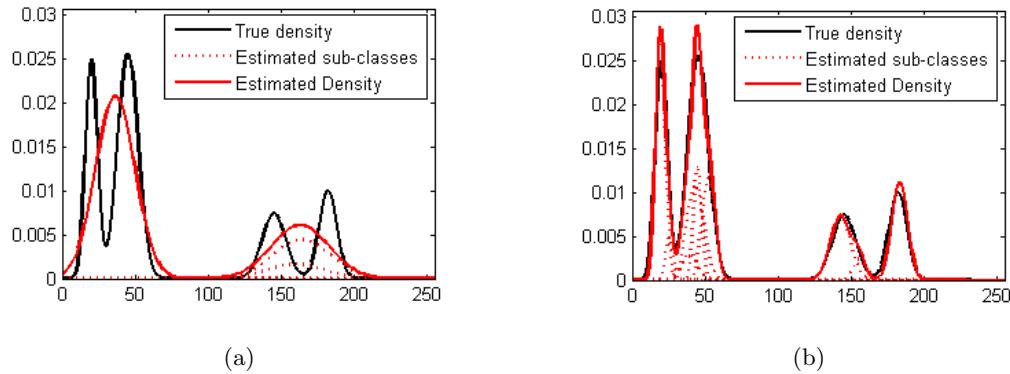


Figure 6.1: The performance of EM algorithm according to the number of sub-classes assumed:
(a) $K = 4$, (b) $K = 10$

generated from a mixture of 4 Gaussian density functions. The (black) solid curves present the true PDF $p(\mathbf{I})$ which generated the observed data samples. The (red) solid curves present the estimated density function $p(\mathcal{I}|\hat{\Theta})$, and the (red) dotted curves present the weighted density functions of sub-classes, given by $\hat{\alpha}_k p(\mathcal{I}|\hat{\theta}_k)$. A successful estimation is determined by how close the $p(\mathcal{I}|\hat{\Theta})$ is estimated to the $p(\mathbf{I})$. In figure 6.1(a), the EM algorithm cannot find any of the true sub-classes with $K = 4$ because the initializations of the parameters $\{\theta_k(t = 0)\}$ are too far away from the true position and too close from the neighbor sub-classes, while it finds all four true sub-classes with $K = 10$ as shown in figure 6.1(b).

¹Note the term *region*, used here, exists in a parameter space. The *region*, used in segmentation, exists on a spatial domain.

6.1.2 EM Algorithm with MML

The estimation of the number of sub-classes K [155] has been an important issue in mixture density estimation. If K is too high, the density estimation may fail to converge to the true density due to singular parameters, while the estimated density function cannot approximate the true underlying mixture density function if K is too low. Figueiredo and Jain proposed an advanced EM method [146] based on the *minimum encoding length* criteria, which automatically selects the number of sub-classes K by annihilating the weak candidates of sub-classes.

The **minimum encoding length** criteria are popular concepts in the encoding and error checking problem in the communication engineering area, and they are also widely applied in pattern classification area. According to the minimum encoding length criteria, e.g. Minimum Message Length (MML) [156, 157] and Minimum Description Length (MDL) [158], a good data generation model, i.e. a parameter set Θ , is a representation which minimizes the length of code required to describe the data samples \mathcal{I} [146, 159]. Let the data samples \mathcal{I} , known to have been generated according to $p(\mathcal{I}|\Theta)$, be encoded and transmitted. If $p(\mathcal{I}|\Theta)$ is fully known to both the transmitter and the receiver, they can both build the same code, and communication can proceed. However, if the parametric model Θ is unknown, the transmitter has to start by estimating and transmitting Θ . This leads to a two-part message, whose total length is given by

$$\mathfrak{L}(\Theta, \mathcal{I}) = \mathfrak{L}(\Theta) + \mathfrak{L}(\mathcal{I}|\Theta) . \quad (6.18)$$

The estimated parameter set $\hat{\Theta}$ is the one minimizing this *length function*.

According to the standard two-part code formulation of MDL and MML, the expected number of data samples generated by sub-class k is $N\alpha_k$, where N denotes the number of data samples. Thus, the optimal, in the MDL sense, code length of each θ_k is $(M/2)\log(N\alpha_k)$, where M denotes the number of parameters specifying θ_k . As the zero-weighted sub-class is considered to be removed, we need to code only non-zero weighted $\alpha_k \neq 0$ sub-classes. The optimal parameter set Θ is estimated by minimizing the cost function

$$\mathfrak{L}(\Theta, \mathcal{I}) = \frac{M}{2} \sum_{k:\alpha_k>0}^K \log \left(\frac{N\alpha_k}{12} \right) + \frac{K_{nz}}{2} \log \frac{N}{12} + \frac{K_{nz}(M+1)}{2} - \log p(\mathcal{I}|\Theta) , \quad (6.19)$$

with respect to Θ given by

$$\hat{\Theta} = \arg \min_{\Theta} \mathfrak{L}(\Theta, \mathcal{I}) , \quad (6.20)$$

where K_{nz} denotes the number of sub-classes with non-zero weight $\alpha_k \neq 0$. From the Bayesian point of view [160], the cost function shown above is equivalent, for fixed K_{nz} , to an *a posteriori* density resulting from the adoption of Dirichlet-type prior for the weights $\{\alpha_k\}$, given by [146]

$$p(\alpha_1, \dots, \alpha_K) \propto \exp \left\{ -\frac{M}{2} \sum_{k=1}^K \log \alpha_k \right\}, \quad (6.21)$$

and a flat prior leading to ML estimates for the parameters $\{\boldsymbol{\theta}_k\}$ specifying each sub-class k . The EM algorithm to minimize the cost function in equation 6.19, with fixed K_{nz} , has the following M-step:

$$\hat{\alpha}_k(t+1) = \frac{\max \left\{ 0, \left(\sum_{n=1}^N w_k(n) \right) - \frac{M}{2} \right\}}{\sum_{j=1}^K \max \left\{ 0, \left(\sum_{n=1}^N w_j(n) \right) - \frac{M}{2} \right\}}, \quad \forall k, \quad (6.22)$$

$$\hat{\boldsymbol{\theta}}_k(t+1) = \arg \max_{\boldsymbol{\theta}_k} Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}(t)), \quad \{k : \hat{\alpha}_k(t+1) > 0\}, \quad (6.23)$$

where the conditional expectation $w_k(n)$ are given by the E-step in equation 6.13. The M-step defined by equation 6.22 and equation 6.23 do the actual *component annihilation* by setting the weight α_k as zero.

The parameter set $\boldsymbol{\theta}_k$ specifying the sub-classes for which $\hat{\alpha}_k(t+1) = 0$ become irrelevant because any sub-class for which $\alpha_k = 0$ does not contribute to the log-likelihood in equation 6.6, thus the algorithm explicitly removes the weakest sub-class, decreasing K_{nz} . This prevents the estimation algorithm from approaching the boundary of the parameter space, one of the drawbacks of the standard EM algorithm for mixture density model. This algorithm also solves the initialization problem by starting with very high number of initial sub-classes K_{init} and removing weak sub-classes step by step. The initial parameters $\{\boldsymbol{\theta}_k(t=0)\}$ can be located anywhere along the whole parameter space. Although this algorithm still does not guarantee the global convergence, it does provide a flexible way to deal with.

The direct use of the standard EM algorithm with M-step in equation 6.22 and 6.23 has a failure mode, i.e. if the number of initial sub-classes K_{init} is too large, no sub-class has enough initial support $\sum_{n=1}^N w_k(n) > M/2, \forall k$ and consequently $\{\hat{\alpha}_k\}$ will be undetermined. *Component-wise EM* (CEM) algorithm [162] avoids this problem by updating each element of parameter set $\boldsymbol{\Theta} = \{\{\alpha_k\}, \{\boldsymbol{\theta}_k\}\}$ sequentially instead of simultaneously. That is, update $\alpha_1, \boldsymbol{\theta}_1$, recompute \mathcal{W} , and update α_2 and $\boldsymbol{\theta}_2$, recompute \mathcal{W} again, and so on. If one sub-class is annihilated $\hat{\alpha}_k(t+1) = 0$, its probability mass is immediately redistributed to the other

sub-classes with non-zero weight $\hat{\alpha}_k(t+1) \neq 0$. This consequently increases their chance of survival, and allows initialization with an arbitrarily large K_{init} .

Figure 6.2 shows the performance of this method applied to the same data samples used for the standard EM method in figure 6.1. 1000 random samples were generated by a mixture

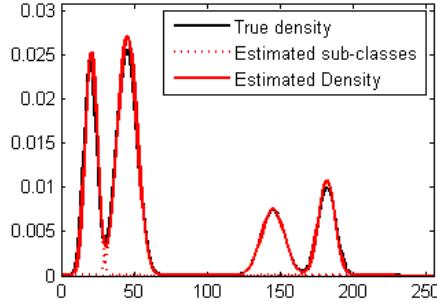


Figure 6.2: The advanced EM method proposed by Figueiredo and Jain applied to the same data used in figure 6.1. The estimation starts with $K_{init} = 32$ and converges to $K = 4$.

density function $p(\mathbf{I})$, presented as the (black) solid curves. Randomly chosen 32 sub-classes are initialized $K_{init} = 32$, and then the weakest sub-class is removed if a criterion is satisfied. After the iterative EM processing, it eventually converges to the four sub-classes $K = 4$, presented as the (red) dotted curves. The estimated mixture density function $p(\mathcal{I}|\hat{\Theta})$, presented as the (red) solid curves, successfully estimates the true mixture density function $p(\mathbf{I})$.

6.2 Non-parametric Density Estimation

In non-parametric approaches, a discrete function instead of a mixture of parametric density functions approximates the true PDF of data samples. Given independently and ideally generated data samples \mathcal{I} , the assumption of non-parametric approach is that we can approximate the statistical distribution of a random variable \mathbf{I} directly from the statistics of \mathcal{I} without any particular stochastic model, such that

$$p(\mathbf{I}) \approx \hat{p}(\mathbf{I}) = h(\mathcal{I}), \quad (6.24)$$

where $h(\cdot)$ denotes a non-parametric density function.

If N observed data samples $\mathcal{I} = \{\mathbf{I}(1), \mathbf{I}(2), \dots, \mathbf{I}(N)\}$ are generated from the true PDF $p(\mathbf{I})$, parametric learning algorithms estimate the stochastic model of random variable \mathbf{I} based on two assumptions: the statistics of \mathcal{I} follow a particular stochastic model, e.g. a Gaussian distribution $p(\mathbf{I}|\boldsymbol{\mu}, \Sigma)$; and the stochastic model can represent $p(\mathcal{I})$. If the data samples \mathcal{I} do not satisfy these assumptions, the estimated parametric expressions $p(\mathbf{I}|\hat{\boldsymbol{\mu}}, \hat{\Sigma})$ cannot approximate $p(\mathbf{I})$. Since non-parametric approaches do not rely on any particular stochastic model, they can estimate the true underlying PDF of data samples unless the data samples \mathcal{I} were observed in a biased manner. This is an attractive feature for a mixture density estimation because the true $p(\mathbf{I})$ is often difficult to express with a particular stochastic model.

6.2.1 Histogram

There are many non-parametric density estimation methods, e.g. Parzen windows, k -nearest neighbor, and histogram. **Histogram**, often called *frequency histogram*, is the most elemental method among them. Given N observed data samples \mathcal{I} , the histogram divides the dynamic range of image intensity \mathbf{I} into a given number of *bins*, and counts the number of data samples corresponding to each bin [163]. Because of this simple procedure, we use a **multi-dimensional histogram density function**² to estimate the discrete non-parametric PDF of image intensity, such that

$$h(\mathcal{I}) \equiv \left\{ \frac{1}{N} \frac{\text{hist}[\mathcal{I}]}{\Delta \mathbf{I}} \right\} : \Re^B \rightarrow \Re, \quad (6.25)$$

where $\text{hist}[\mathcal{I}]$ denotes a multi-dimensional histogram of \mathcal{I} , $\Delta \mathbf{I}$ denotes the volume of a histogram bin, and N denotes the number of data samples \mathcal{I} . $h(\mathcal{I})$ should satisfy the unit volume condition given by

$$\sum_I h(\mathcal{I}) = 1. \quad (6.26)$$

In the case of scalar images, $\Delta \mathbf{I}$ is equivalent to the width of a histogram bin, and the range of each $\Delta \mathbf{I}$ do not overlap, which is different from other non-parametric methods. The histogram density function $h(\mathcal{I})$ approximates $p(\mathbf{I})$ in a discrete manner. Figure 6.3 shows an example of the histogram density function $h(\mathcal{I})$ of the same data samples used in the previous graphs. The width of each bin $\Delta \mathbf{I}$ is set to 1 in figure 6.3(a), and set to 3 in figure 6.3(b). With the same number of data samples, $h(\mathcal{I})$ tends to be smoother losing its precision as $\Delta \mathbf{I}$ increases, while

²Although $h(\mathcal{I})$ shown in equation 6.25 is occasionally referred to *histogram* or *frequency histogram* in other publications [163], it is defined as *histogram density function* in this document. Histogram and frequency histogram are referred to the frequency number for each bin, i.e. the histogram in common sense.

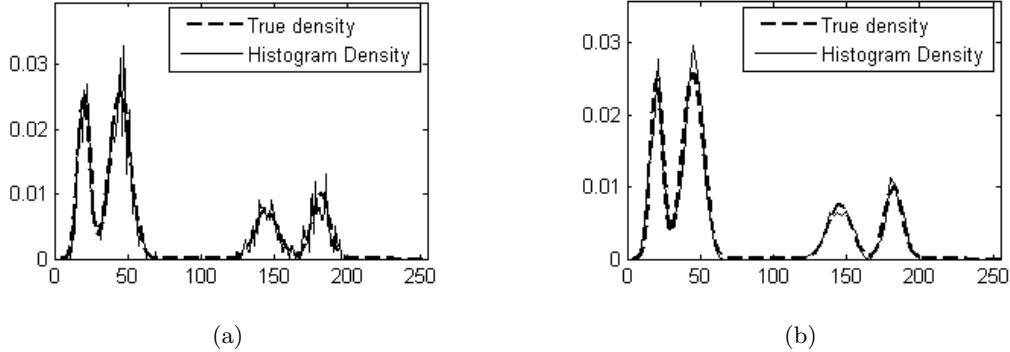


Figure 6.3: An example of the histogram density function $h(\mathcal{I})$: (a) $\Delta\mathbf{I} = 1$, (b) $\Delta\mathbf{I} = 3$

$h(\mathcal{I})$ tends to be more sparky as $\Delta\mathbf{I}$ decreases. If $\Delta\mathbf{I}$ is infinitely narrow and N is infinitely high at the same time, the discrete $h(\mathcal{I})$ can ideally approximate the true PDF $p(\mathbf{I})$, such that

$$p(\mathbf{I}) \approx \lim_{\Delta\mathbf{I} \rightarrow 0} \lim_{N \rightarrow \infty} h(\mathcal{I}). \quad (6.27)$$

The process of computing a histogram density function $h(\mathcal{I})$ is simple and easy to implement. Also, $h(\mathcal{I})$ does not require any iterative learning process, and consequently requires shorter computation time. Since active contours have an iterative process demanding long computation time, $h(\mathcal{I})$ is a quite attractive density estimation method for active contours. However, the accumulation procedure requiems a huge amount of memory depending on the dimensionality of \mathbf{I} . For example, a histogram of a 24bit RGB image intensity consists of $2^{24} = 16777216$ bins, 128Mbytes if 8bytes/bin.

The frequency of each bin of a histogram is independent ignoring the neighborhood in the image intensity space. That is, the frequency of the bin a is independent from the frequency of the bin $a + 1$. This localized frequency is a good feature for mixture density model because weak sub-classes can survive without being dominated by a strong neighbor sub-class as shown in figure 6.1(a). However, it also overfits the distribution of the histogram density function $h(\mathcal{I})$ to the data samples \mathcal{I} , and reduces the flexibility of the estimated PDF $p(\mathbf{I})$. That is, the $p(\mathbf{I})$ from a set of data samples \mathcal{I}_1 may be different from the $p(\mathbf{I})$ from the other set of data samples \mathcal{I}_2 even if both \mathcal{I}_1 and \mathcal{I}_2 are generated from a statistical model. A partial solution of this problem is to provide a large data set. This problem becomes even more serious particularly if the dimensionality of \mathbf{I} is high. Thus, a histogram density function needs extremely higher number of data samples as the dimension of data grows, which is called *the curse of*

dimensionality [164]. The high sensitivity to noise is another problem caused by the localized bins. Kernel density estimation methods, e.g. Parzen window [143], provide more flexible and smooth distribution than $h(\mathcal{I})$ at the expense of computation time.

Chapter 7

Active Contour Implementation using Level Set

The proposed segmentation methods are implemented in a form of active contours. We propose to use the region-based active contour model using level set theory. The level set implementation of the proposed active contour model is based on the *multi-phase active contour* model proposed by Chan and Vese [110, 76, 115]. In this chapter, we discuss the level set implementation of the proposed active contour models.

7.1 The Base Active Contour Model

Multi-phase active contours can partition an image into more than two subsets simultaneously. Let us redefine the entire domain Ω of an image $\mathbf{I}(x, y)$ as a disjoint set of subsets, such that

$$\Omega \equiv \bigcup_i \Omega_i, \quad (7.1)$$

where the interior regions include the contour pixels $C \in \Omega_{in}$. Then, each subset can be identified by a set of binary *identity functions*

$$\chi_i(x, y) \equiv \begin{cases} 1, & \text{if } (x, y) \in \Omega_i \\ 0, & \text{otherwise} \end{cases}, \quad \forall (x, y) \in \Omega, \quad \forall i \quad (7.2)$$

composed of a group of regularized unit step functions $H_j : \Omega \rightarrow \{0, 1\}$ given by

$$H_j \equiv H_\epsilon(\phi_j(x, y)) \approx \begin{cases} 1, & \text{if } \phi_j(x, y) \geq 0 \\ 0, & \text{if } \phi_j(x, y) < 0 \end{cases}, \quad \forall(x, y) \in \Omega, \quad \forall j. \quad (7.3)$$

The identity functions $\{\chi_i(x, y) : \Omega \rightarrow \{0, 1\}\}$ for the case of 2 subsets and 4 subsets are respectively defined as [76]

$$\begin{bmatrix} \chi_0(x, y) \\ \chi_1(x, y) \end{bmatrix} = \left\{ (x, y) : \begin{bmatrix} (1 - H_1) \\ H_1 \end{bmatrix} \right\}, \quad \forall(x, y) \in \Omega, \quad (7.4)$$

and

$$\begin{bmatrix} \chi_0(x, y) \\ \chi_1(x, y) \\ \chi_2(x, y) \\ \chi_3(x, y) \end{bmatrix} = \left\{ (x, y) : \begin{bmatrix} (1 - H_2)(1 - H_1) \\ (1 - H_2)H_1 \\ H_2(1 - H_1) \\ H_2H_1 \end{bmatrix} \right\}, \quad \forall(x, y) \in \Omega. \quad (7.5)$$

J level set functions $\{\phi_1, \dots, \phi_j, \dots, \phi_J\}$ can compose up to 2^J subsets $\{\Omega_1, \dots, \Omega_i, \dots, \Omega_{2^J}\}$ in this way. An example of subsets, defined by multi-phase level set functions, is shown in figure 7.1 where $\{\Omega_0, \Omega_1, \Omega_2, \Omega_3\}$ denote the four subsets defined by two level set functions $\{\phi_1, \phi_2\}$.

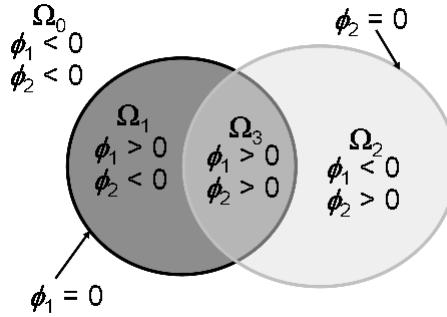


Figure 7.1: Subsets and contours defined by two level set functions, $\{\phi_1, \phi_2\}$

Using the identity functions $\{\chi_i(x, y)\}$, the integration over each subset Ω_i in the global energy function E shown in equation 4.11 and 4.14 can be transformed to the integration over the entire image plane Ω , such as

$$\int_{\Omega_i} e_i(x, y) dx dy = \int_{\Omega} e_i(x, y) \chi_i(x, y) dx dy, \quad \forall i, \quad (7.6)$$

which makes the computation much easier. Also, the length of contours $|C_j|$ is equivalent to the integration of $|\nabla H_j|$ over Ω , such that

$$|C_j| = \int_{\Omega} |\nabla H_{\epsilon}(\phi_j(x, y))| dx dy, \quad \forall j, \quad (7.7)$$

where C_j denotes a set of active contours formed by the corresponding level set function $\phi_j(x, y)$ as $C_j \equiv \{(x, y) : \phi_j(x, y) = 0\}$. The global energy function of the multi-phase active contour model and the associated Euler-Lagrange equation obtained by minimizing the energy function E with respect to $\Phi = \{\phi_1, \dots, \phi_j, \dots, \phi_J\}$, which are introduced in [76, 115], can be generalized with an arbitrary form of objective functions $e_i(x, y)$, such as

$$\begin{aligned} E &\equiv \sum_{i=1}^{2^J} \int_{\Omega_i} e_i(x, y) dx dy + \nu \sum_{j=1}^J |C_j| \\ &= \sum_{i=1}^{2^J} \int_{\Omega} e_i(x, y) \chi_i(x, y) dx dy + \nu \sum_{j=1}^J \int_{\Omega} |\nabla H_j| dx dy, \end{aligned} \quad (7.8)$$

and

$$\frac{\partial \phi_j(x, y)}{\partial t} = \delta_j \left[\nu \kappa_j - \sum_{i=1}^{2^J} e_i(x, y) \frac{\partial \chi_i}{\partial H_j} \right], \quad \forall j, \quad (7.9)$$

where $\delta_j \equiv \delta_{\epsilon}(\phi_j(x, y))$ denotes the first derivatives of H_j with respect to ϕ_j , and $\kappa_j \equiv \kappa(\phi_j(x, y))$ denotes the mean curvature of $\phi_j(x, y)$.

7.2 Proposed Active Contour Model

The proposed active contour model is obtained by substituting the two proposed objective functions $e(x, y|p_i)$ in equation 4.9 and $e_i(x, y)$ in equation 4.13 into the generalized multi-phase active contour model shown in equation 7.9, such that

$$\frac{\partial \phi_j(x, y)}{\partial t} = \delta_j \left[\nu \kappa_j + \sum_{i=1}^{2^J} \log p_i(\mathbf{I}(x, y)) \frac{\partial \chi_i}{\partial H_j} \right], \quad \forall j, \quad (7.10)$$

where $p_i(\mathbf{I}(x, y)) \equiv p(\mathbf{I}(x, y)|(x, y) \in \Omega_i)$ for unsupervised segmentation model, and $p_i(\mathbf{I}(x, y)) \equiv p(\mathbf{I}(x, y)|\mathbf{I}(x, y) \in \mathcal{I}_i)$ for supervised segmentation model. The conditional PDF $p_i(\cdot)$ could be either a parametric expression or a non-parametric expression. We propose to use a mixture of multivariate density functions as shown in equation 6.3 for supervised segmentation model. More detail will be discussed in chapter 9 and 10. We also propose to use a histogram density

function as shown in equation 6.24 and 6.25 for unsupervised segmentation model. More detail will be discussed in chapter 8. The actual level set evolution equations for the case of two subsets and four subsets are respectively given by

$$\begin{aligned}\frac{\partial\phi(x,y)}{\partial t} &= \delta_\epsilon(\phi(x,y))[\nu\kappa(\phi(x,y)) \\ &\quad + \{\log p_1(\mathbf{I}(x,y)) - \log p_0(\mathbf{I}(x,y))\}],\end{aligned}\tag{7.11}$$

and

$$\begin{aligned}\frac{\partial\phi_1(x,y)}{\partial t} &= \delta_1 \left[\begin{array}{c} \nu\kappa_1 + \\ \left\{ \begin{array}{l} (\log p_3 - \log p_2)H_2 + \\ (\log p_1 - \log p_0)(1 - H_2) \end{array} \right\} \end{array} \right] \\ \frac{\partial\phi_2(x,y)}{\partial t} &= \delta_2 \left[\begin{array}{c} \nu\kappa_2 + \\ \left\{ \begin{array}{l} (\log p_3 - \log p_1)H_1 + \\ (\log p_2 - \log p_0)(1 - H_1) \end{array} \right\} \end{array} \right],\end{aligned}\tag{7.12}$$

where $\log p_i \equiv \log p_i(\mathbf{I}(x,y))$. $\delta_j \equiv \delta_\epsilon(\phi_j(x,y))$ is not necessarily a Dirac delta function, but it performs a role of window like a bandpass filter which controls the width ε to update $\phi_j(x,y)$. The level set function $\phi_j(x,y)$ only where $|\phi(x,y)| < \varepsilon$, i.e. around corresponding contours C_j , are updated and the rest of area is ignored. The curvature term κ_j keeps C_j in a smooth shape. During the contour evolution, $p_i(\mathbf{I}(x,y))$ for supervised segmentation methods returns the probability of $\mathbf{I}(x,y)$ under the condition that $\mathbf{I}(x,y)$ is an element of given training samples \mathcal{I}_i , while $p_i(\mathbf{I}(x,y))$ for unsupervised segmentation methods returns the probability of $\mathbf{I}(x,y)$ under the condition that $\mathbf{I}(x,y)$ is an element of the subset Ω_i . These conditional probabilities provide a force which changes the corresponding level set functions, i.e. propagates the corresponding active contours, and consequently partitions the given image satisfying the desired property, which is either uniformity of \mathbf{I} within Ω_i or similarity between the statistics of \mathbf{I} and \mathcal{I}_i . The terms inside of $\{\cdot\}$ in above expressions provide segmentation force, while the curvature terms provide a regularity force. Constant ν controls the balance between these two forces.

Chapter 8

Unsupervised Active Contour Model using Multi-dimensional Histograms

In this chapter, we propose an unsupervised active contour model using the non-parametric density estimation method introduced in section 6.2.1. The proposed active contour model measures a non-parametric PDF expression of each subset at each iteration, and updates level set functions based on the measured PDF expressions. Section 8.1 discuss the proposed unsupervised image segmentation model. We propose to use a multi-dimensional histogram density function as a discrete non-parametric density function of image intensity within each subset. The detail of this approach is discussed in section 8.2. Section 8.3 discusses the implementation of active contours using level set. Section 8.4 shows the numerical algorithm of the proposed method, and section 8.5 shows the experimental results applied on synthetic and real images.

8.1 Unsupervised Image Segmentation

The proposed method follows the classical role of image segmentation, which partitions an image without any prior knowledge or training samples, using a global energy minimization. The global energy function is given by

$$E(\{p_i\}, C) \equiv \sum_i \int_{\Omega_i} e(x, y|p_i) dx dy + \nu |C|, \quad (8.1)$$

and the global energy E is minimized with respect to two expressions: $\{p_i\}$, i.e. the statistical representation of the image within a subset Ω_i , and the variational contours C . First, the

minimum E with respect to $\{p_i\}$ is achieved while C is fixed. Then, the minimum E with respect to C is achieved while $\{p_i\}$ are fixed. An objective function

$$e(x, y|p_i) = -\log p_i, \quad \forall (x, y) \in \Omega, \quad \forall i, \quad (8.2)$$

determines the region-based segmentation criteria based on the statistical distribution of image intensity within each subset Ω_i . As $e(\cdot)$ has the minimum value where p_i has the maximum value, the minimum E with respect to $\{p_i\}$ is achieved where every p_i has the maximum. This is done by a similar way to the simplified Mumford-Shah segmentation model introduced in section 4.2 except that p_i is a statistical distribution of image intensity here, instead of a simple expression f_i . The best p_i satisfying the segmentation criteria is a conditional PDF of image intensity on the condition that the image pixel is an element of Ω_i , given by

$$p_i \equiv p(\mathbf{I}(x, y)|(x, y) \in \Omega_i), \quad \forall i, \quad (8.3)$$

where $\mathbf{I}(x, y) : \Omega \rightarrow \Re^B$ denotes a vector-valued image pixel. The estimated p_i provides a force to propagate the variational contours C to the proper position where C divides the entire domain Ω of the image $\mathbf{I}(x, y)$ into multiple subsets $\{\Omega_i\}$. This performs an unsupervised image segmentation. Then, the minimization of E with respect to C smoothes the variational boundaries C minimizing $|C|$.

As the proposed method does not require any prior knowledge or training samples, the use of proposed method is convenient and flexible, so it can be used in any kinds of images as the first choice. However, the performance of segmentation tends to be limited depending on the character of the given images. It is particularly limited if the image has significant non-uniform subsets. Supervised segmentation methods introduced in the next two chapters may show better performance in those non-uniform images.

8.2 Multi-dimensional Histogram Density Function

We have proposed to use a multivariate mixture density function as the statistical distribution of image intensity within a subset. In the proposed method, we propose to use a normalized multi-dimensional histogram, which is also introduced as a multi-dimensional histogram density function in section 6.2.1, as a non-parametric and discrete expression of the

multivariate mixture density function.

Let $\mathcal{I} = \{\mathbf{I}(1), \mathbf{I}(2), \dots, \mathbf{I}(N)\}$ be the N observed data samples, and $\mathbf{I} \in \Re^B$ be the vector-valued image intensity. The histogram density function of given data samples is given by

$$h(\mathcal{I}) \equiv \left\{ \frac{1}{N} \frac{\text{hist}[\mathcal{I}]}{\Delta \mathbf{I}} \right\} : \Re^B \rightarrow \Re, \quad (8.4)$$

where $\text{hist}[\mathcal{I}]$ denotes a multi-dimensional histogram of \mathcal{I} , and $\Delta \mathbf{I}$ denotes the volume of a histogram bin. If the histogram density function $h(\mathcal{I})$ is normalized as

$$\sum_{\mathbf{I}} h(\mathcal{I}) = 1 \quad (8.5)$$

with unlimited number of data samples, i.e. $N \rightarrow \infty$, and infinitely narrow histogram bins, i.e. $\Delta \mathbf{I} \rightarrow 0$, the discrete $h(\mathcal{I})$ ideally approximates a continuous PDF, given by

$$p(\mathbf{I}) \approx \lim_{\Delta \mathbf{I} \rightarrow 0} \lim_{N \rightarrow \infty} h(\mathcal{I}). \quad (8.6)$$

In the proposed active contour model, $h(\mathcal{I})$ substitutes the p_i in equation 8.1 and 8.2 as a discrete non-parametric form of a multivariate mixture density function.

In practical image processing problems, the number of data samples is limited and the width of a histogram bin cannot be zero. However, the native data is already in a sampled discrete form in digital image processing, so the quantization error caused by $h(\mathbf{I})$ is reasonably low if $\Delta \mathbf{I} = 1$. The simple computation of a histogram is another advantage. Although there are many non-parametric density estimation methods, e.g. k -nearest neighborhood, Parzen window, histogram provides fairly reasonable information with a short computation time. However, the required amount of memory significantly increases depending on the number of bands B and the resolution of image intensity $\Delta \mathbf{I}$. For example, a 24bit RGB image requires $2^{24} \times 8 = 128$ Mbytes, while a gray image requires only $2^8 \times 8 = 2$ Kbytes.

The statistics of image intensity in an image is not always representable as a simple parametric expression, such as a Gaussian distribution. Since the proposed method does not rely on a particular stochastic model, the use of the proposed method is more flexible and robust than other methods using simple parametric expressions. Figure 8.1 shows an example of non-uniform images whose statistical distribution is unrepresentable with a simple parametric expression. The wood pattern shown in figure 8.1(a) shows a non-uniform texture, and

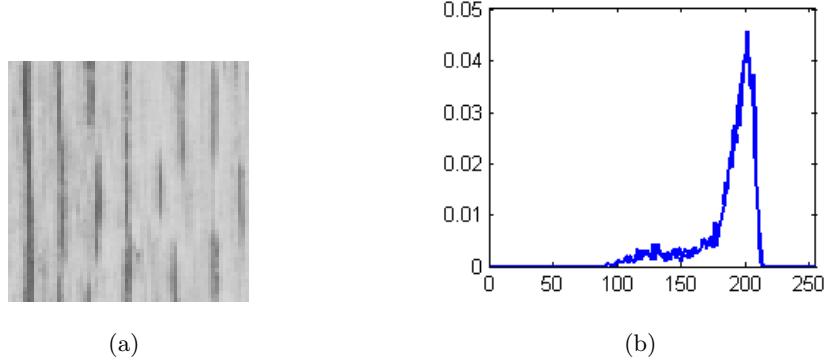


Figure 8.1: Statistical distribution of image intensity: (a) a wood pattern, (b) the histogram

the statistical distribution, i.e. a histogram, of image intensity shown in figure 8.1(b) shows a high peak in the range of high intensity and a low and wide distribution along the range of low intensity. The statistics of these data samples are not easily representable with a simple Gaussian or other parametric distributions though it is trivial for a non-parametric method.

8.3 Contour Evolution

The implementation of proposed active contour model is done using the multi-phase active contour model introduced in section 7.2. We define multiple level set functions $\Phi = \{\phi_1, \dots, \phi_j, \dots, \phi_J\}$ on the spatial domain, and each level set function represents a set of contours C_j . J level set functions can partition an image up to 2^J subsets as shown in chapter 7.

The Euler-Lagrange equation obtained by minimizing the energy function E with respect to level set functions $\Phi = \{\phi_1, \dots, \phi_j, \dots, \phi_J\}$ is given by

$$\frac{\partial \phi_j(x, y)}{\partial t} = \delta_j \left[\nu \kappa_j + \sum_{i=1}^{2^J} \log h(\mathbf{I}(x, y)|(x, y) \in \Omega_i) \frac{\partial \chi_i}{\partial H_j} \right], \quad \forall j, \quad (8.7)$$

where $\chi_i(x, y) : \Omega \rightarrow \{0, 1\}$ denotes the binary identity functions introduced in equation 7.4 and 7.5, and $H_j \equiv H_\epsilon(\phi_j(x, y)) : \Omega \rightarrow \mathbb{R}$ denotes the regularized unit step function of $\phi_j(x, y)$, introduced in equation 7.3. $h(\mathbf{I}(x, y)|(x, y) \in \Omega_i)$ denotes the histogram density function measured within Ω_i , and it determines the amount of update in $\phi_j(x, y)$. $\delta_j \equiv \delta_\epsilon(\phi_j(x, y)) : \Omega \rightarrow \mathbb{R}$ performs a function of a window, which updates $\phi_j(x, y)$ only around the contours where

$|\phi_j(x, y)| < \varepsilon$. The curvature term $\kappa_j \equiv \kappa(\phi_j(x, y)) : \Omega \rightarrow \mathbb{R}$, introduced in equation 3.10, keeps the regularity of contours, and ν controls the relative weight between the segmentation force and regularity force.

The update of level set functions is performed as an iterative processing. At each iteration, we measure $h(\mathbf{I}(x, y)|(x, y) \in \Omega_i)$ of each subset, and update each level set function $\phi_j(x, y)$ according to the update function shown above until the active contours finally converge to the position of desired boundaries.

8.4 Algorithm

Table 8.1: The input and output variables used in the unsupervised multi-dimensional histogram method

<ul style="list-style-type: none"> • Input
<ul style="list-style-type: none"> – $\mathbf{I}(x, y) : \Omega \rightarrow \mathbb{R}^B$, a multispectral image.
<ul style="list-style-type: none"> – J, the number of level set functions $\{\phi_j(x, y)\} =$ the number of active contours $\{C_j\}$.
<ul style="list-style-type: none"> – ν, a parameter to control the regularity of contours.
<ul style="list-style-type: none"> – Δt, the time step of iterative processing.
<ul style="list-style-type: none"> • Output
<ul style="list-style-type: none"> – $S(x, y) : \Omega \rightarrow \{1, \dots, 2^J\}$, the segmented image

Table 8.1 shows the input and output variables used in the proposed method. A multispectral image $\mathbf{I}(x, y)$ is given as the input data. The number of level set functions, which is equivalent to the number of contours, is set as J . ν controls the regularity of contour evolution by controlling the relative balance between segmentation force and regularity force. Δt determines the convergence speed. After the proposed active contours converge, the algorithm produces the segmented image $S(x, y)$ indicating each subset with different numbers.

As shown in algorithm 1, the proposed active contour algorithm consists of two stages: initialization and contour evolution. During the initialization stage, J sets of initial contours $\{C_j\}$

are given either manually or from a defined set, and corresponding level set functions are initialized as the signed distance from each (x, y) to the closest contours. During the iterative contour evolution, the mean curvature $\kappa_j(x, y)$ and the unit step function $H_j(x, y)$ are computed from each level set function $\phi_j(x, y)$. The mathematical description of function `meanCurvature()` and `unitStep()` are described respectively in equations 3.10 and 7.3. Then, binary identity functions $\chi_i(x, y)$ and the histogram density functions $h_i(\mathbf{I})$ for each subset Ω_i are measured. The mathematical description of functions `identityFunction()` and `histogramDensity()` are introduced respectively in equations 7.4 and 8.4. The measured $h_i(\mathbf{I})$ updates objective functions $e_i(x, y)$ for Ω_i , and finally level set functions $\{\phi_j(x, y)\}$ are updated.

Algorithm 1: Unsupervised Active Contour Model using Multi-dimensional Histogram Density Functions

Initialization

$t = 0$.

for $j = 1 \cdots J$ **do**

$C_j(x(s), y(s), t = 0)$: Initialize contours.

$\phi_j(x, y, t = 0) = \pm D((x, y), \mathbb{N}_{(x,y)}(C_j))$: Initialize level set functions.

Contour evolution

repeat

for $j = 1 \cdots J$ **do**

$\kappa_j(x, y) = \text{meanCurvature}(\phi_j(x, y))$

$H_j(x, y) = \text{unitStep}(\phi_j(x, y))$

Unsupervised region-based segmentation

for $i = 1 \cdots 2^J$ **do**

$\chi_i(x, y) = \text{identityFunction}(\{H_j(x, y), \forall j\})$

$h_i(\mathbf{I}) = \text{histogramDensity}(\mathbf{I}(x, y)\chi_i(x, y))$

$e_i(x, y) = -\log h_i(\mathbf{I}(x, y))$

Update level set function

for $j = 1 \cdots J$ **do**

$\phi_j(x, y) = \phi_j(x, y) + \Delta t \delta(\phi_j(x, y)) \left[\nu \kappa(\phi_j(x, y)) - \sum_{i=1}^{2^J} e_i(x, y) \frac{\partial \chi_i}{\partial H_j} \right]$

$t = t + \Delta t$

until *until the evolution of contour converges*

Segmentation result

$$S(x, y) = \sum_{i=1}^{2^J} \chi_i(x, y)$$

Table 8.2 shows the input and output variables used in the histogram density function `histogramDensity(I)`. The density function takes three inputs: data samples \mathcal{I} , the dynamic range of image intensity, and the size of histogram bins $\Delta\mathbf{I}$. The dynamic range and $\Delta\mathbf{I}$ determines the resolution of the histogram. After computation, the density function produces a normalized multi-dimensional histogram as a non-parametric and discrete density function. As shown in algorithm 2, the estimation of a histogram density function consists of two stages: initialization and computation. The mathematical model of this method was introduced in equation 8.4. First, the initial histogram is defined based on the dimension of the vector-valued

Table 8.2: The input and output variables used in the multi-dimensional histogram density function

<ul style="list-style-type: none"> • Input <ul style="list-style-type: none"> – $\mathcal{I} = \{\mathbf{I}(1), \mathbf{I}(2), \dots, \mathbf{I}(n), \dots, \mathbf{I}(N)\}$, $\mathbf{I} \in \Re^B$, data samples. – $[(\min_1 \max_1)^T, \dots, (\min_b \max_b)^T, \dots, (\min_B \max_B)^T]$, the dynamic range of image intensity \mathbf{I} at each band. – $\Delta\mathbf{I}$, the volume of a histogram bin. • Output <ul style="list-style-type: none"> – $h(\mathbf{I}) : \Re^B \rightarrow \Re$, histogram density function.
--

image intensity $\mathbf{I} \in \Re^B$ and the given dynamic range. Then, the corresponding histogram bin for each data sample is accumulated, and the whole histogram is normalized in the final stage.

Algorithm 2: Histogram density function

Initialization

`hist[\mathbf{I}]`: Initiate a B -dimensional histogram

Computing density function

`for` $n = 1 \dots N$ `do` $\text{hist}(\mathbf{I}(n)) = \text{hist}(\mathbf{I}(n)) + 1$

$h(\mathbf{I}) = \text{hist}(\mathbf{I}) / (N \Delta\mathbf{I})$: Normalization

Figure 8.2 shows the iterative contour evolution of the proposed active contour model. The figures on the top show the contour evolution, and the figures on the bottom show the corresponding segments. At the initial stage, a set of initial contours are given from a predefined shape as shown in the left figures. The contours move separating the given image, and finally converge to the boundary of the small rectangle.

8.5 Experiments

In this section, we apply the proposed segmentation method to three different images, and compare the results of the proposed method to the results of other active contour models. Two active contour models, which share the same region-based segmentation principle, were used as benchmarks: a piecewise-constant active contour model proposed by Sandberg and

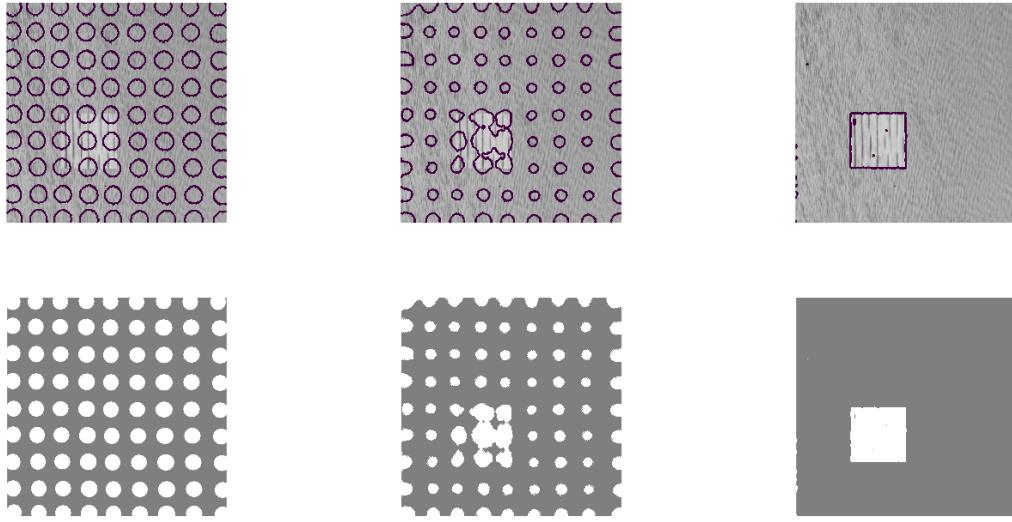


Figure 8.2: The iterative procedure of proposed active contour model: (top) contour evolution, (bottom) corresponding segments

Chan [6, 125, 142] and another active contour model proposed by Rousson and Deriche [11, 119]. Sandberg's method will be referred to method 1, and Rousson's method will be referred to method 2. All three methods including the proposed method determine the segmentation criteria based on the statistical property of image intensity measured within a subset. Method 1 represents each subset as a vector-valued constant number. Method 2 represents each subset as a multivariate Gaussian density function. Both methods 1 and 2 use parameters to represent each subset. Method 1 uses the mean of image intensity measured within the subset, while method 2 uses the mean and covariance of image intensity measured within the subset. The detail of mathematical and analytical descriptions of both methods are presented in section 3.4.

Figure 8.3(a) shows an example of synthetic gray images with textures. There is a small rectangle in the image, and both the rectangle and the background have their own texture pattern. The object of this experiment is to separate the rectangle from the background. Figure 8.3(b) shows the *ground truth*, the ideal result of segmentation, of the given image. The dark part presents the background, class 1, while the bright part presents the rectangle, class 2.

Figure 8.4 shows the result of method 1 applied on figure 8.3(a). Figure 8.4(a) shows the

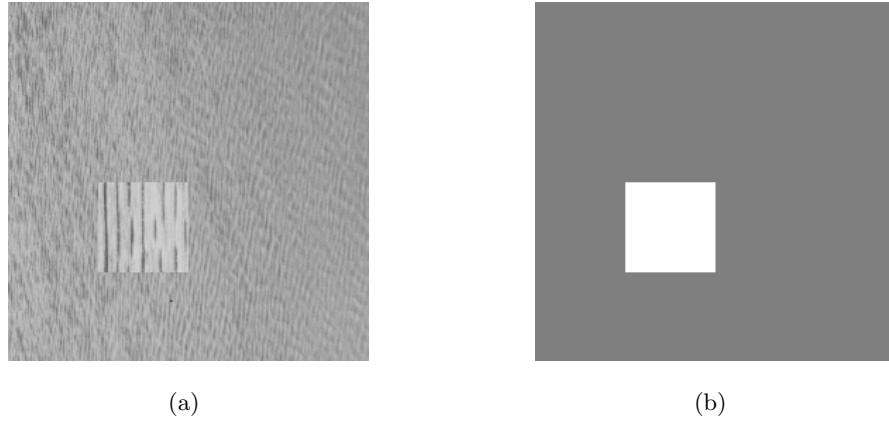


Figure 8.3: Synthetic textured image: (a) a textured gray image, (b) the ground truth image

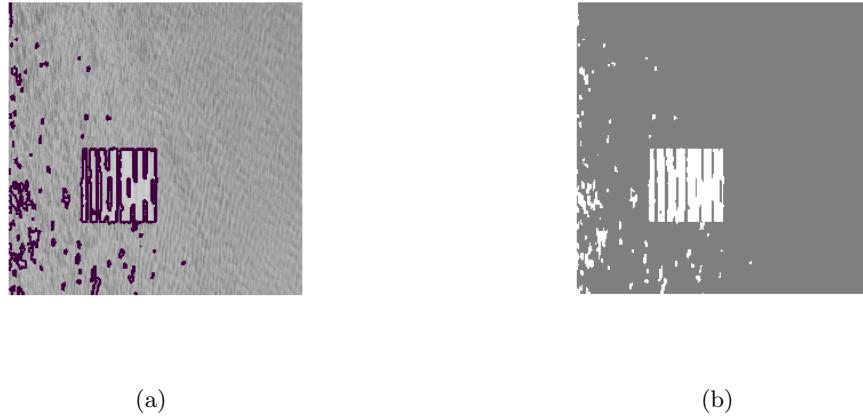


Figure 8.4: Method 1 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets

final stage of contour evolution, and figure 8.4(b) shows the corresponding segmentation result. Method 1 cannot partition the rectangle as a single object as it separates the rectangle as dark sub-regions and bright sub-regions, and the same effect also appears on the background. Method 1 fails in this problem because the dark intensity and the bright intensity of the rectangle stay too far away in the image intensity space, so it cannot represent the subset with the average value of the intensity.

Figure 8.5 shows the result of method 2 applied on figure 8.3(a). Figure 8.5(a) shows the final stage of contour evolution, and figure 8.5(b) shows the corresponding segmentation result. Method 2 shows a better segmentation result than method 1. It successfully separates the



Figure 8.5: Method 2 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets

rectangle from the background, but there are still a few artifacts in the background.

Figure 8.6 shows the result of the proposed method applied on the same image. Figure 8.6(a)

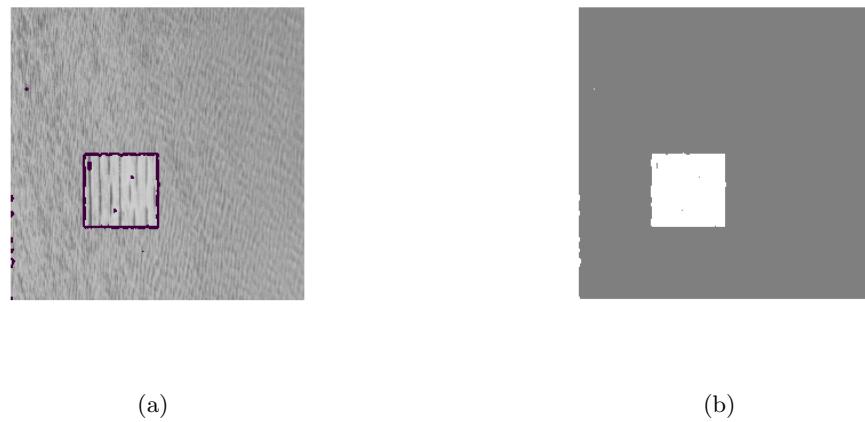


Figure 8.6: Proposed method applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets

shows the final stage of contour evolution, and figure 8.6(b) shows the corresponding segmentation result. The proposed method successfully separates the rectangle from the background without significant artifacts, which were observed in the cases of other two methods.

For more precise comparison, we measure the statistical distribution of image intensity

within class 2, which is supposed to be the small rectangle in the segmentation result. Figure 8.7(a), 8.7(b),

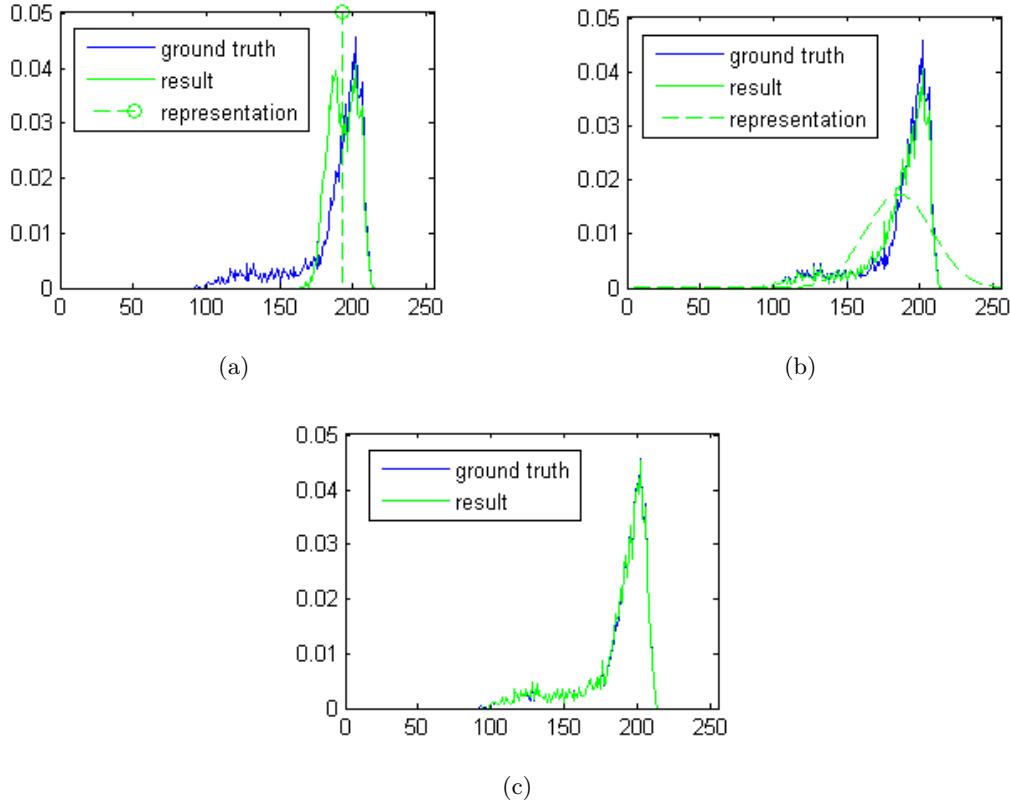


Figure 8.7: Statistical distribution of image intensity within class 2, the small rectangle: (a) method 1, (b) method 2, (c) proposed method

and 8.7(c) respectively present the results of method 1, method 2, and the proposed method. The blue solid line of all three graphs presents the statistical distribution of image intensity within the rectangle in the ground truth image. Because of the wood pattern, the statistical distribution of the rectangle is not simple. There is a high peak at the region of high intensity in the graph, and there are multiple low peaks at the region of dark intensity in the graph. The green solid line of all three graphs presents the same statistics measured within class 2 in the segmentation result. As the green solid line exists closer to the blue solid line, it presents better result. In figure 8.7(a), the dotted green line with a circle presents the representation of the subset used in method 1, i.e. the mean of green solid line. The green line is very different from the ground truth because the constant representation is not sufficient to represent the complicated ground truth. In figure 8.7(b), the dotted green line presents the representation of the subset used in method 2, i.e. the Gaussian density function of image intensity measured.

Although the green solid line stays close to the blue line, the dotted line seems quite different from the actual distribution, and it consequently results in the artifacts shown in figure 8.5. In figure 8.7(c), the result of the proposed method and the ground truth data is almost identical.

Figure 8.8(a) shows another example of gray images with textures. There are five objects

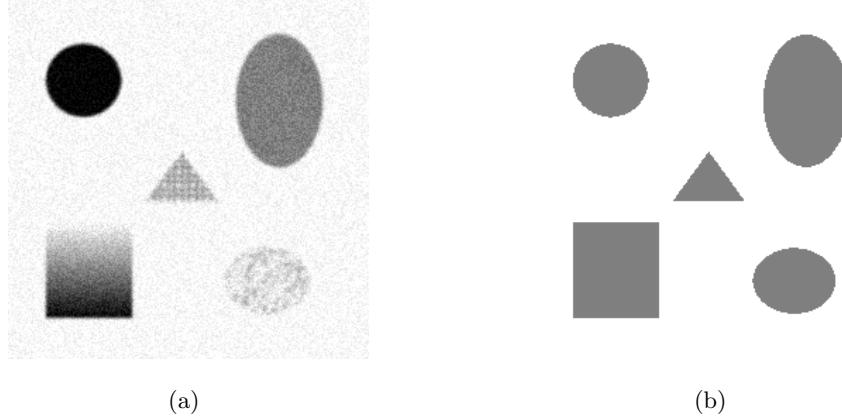


Figure 8.8: Synthetic textured image: (a) a textured gray image, (b) the ground truth image

in a synthetic gray image: a circle, a rectangle, a horizontal ellipse, a vertical ellipse, and a triangle. The circle and vertical ellipse have uniform image intensity, but the triangle and the horizontal ellipse have textured pattern. The rectangle has a gradient intensity. The overall image has some noise, so the edges between each object and the background are not clear. The object of this experiment is to separate all objects from the background forming two classes, i.e. background and objects. Figure 8.8(b) shows the ground truth of this experiment. Class 1, i.e. background, appears as a bright sub-region, while class 2, i.e. objects, appear as dark sub-regions.

Figure 8.9 shows the result of method 1 applied on figure 8.8(a). Figure 8.9(a) shows the final stage of contour evolution, and figure 8.9(b) shows the corresponding segmentation result. Although method 1 successfully partitions the circle and vertical ellipse as class 2, it completely fails to separate the horizontal ellipse from the background. Also, only some portions of the rectangle and triangle appear as class 2 in the final result due to their textured pattern and gradient intensity. Since method 1 represents two classes as two constant numbers, it cannot recognize an object if the image intensity of the object is not uniform.

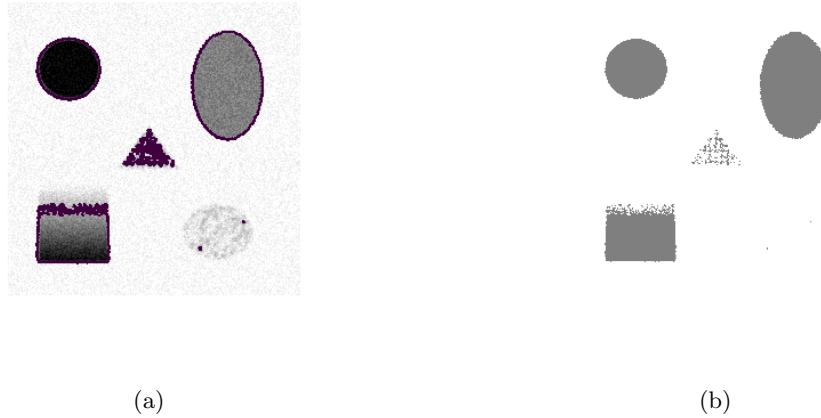


Figure 8.9: Method 1 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets

Figure 8.10 shows the result of method 2 applied on figure 8.8(a). Figure 8.10(a) shows

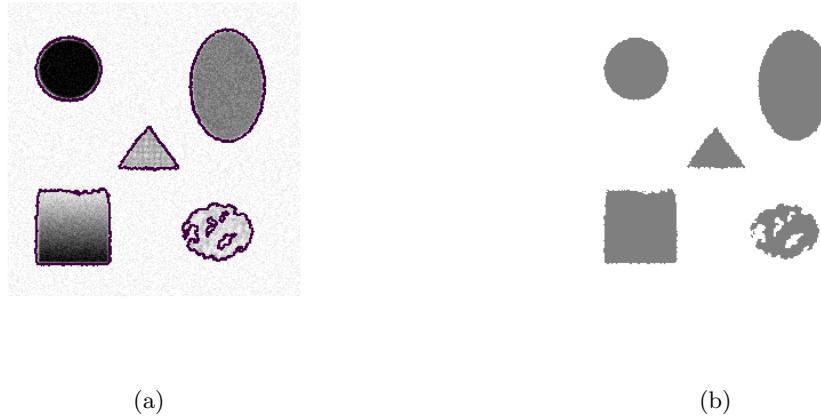


Figure 8.10: Method 2 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets

the final stage of contour evolution, and figure 8.10(b) shows the corresponding segmentation result. Method 2 shows a better segmentation result than method 1 successfully separating the circle, vertical ellipse, triangle, and rectangle from the background. However, there are still a few artifacts on the horizontal ellipse due to the textured pattern because the statistical distribution of the texture pattern stays out of the Gaussian density function of class 2.

Figure 8.11 shows the result of the proposed method applied on the same image. Figure 8.11(a)

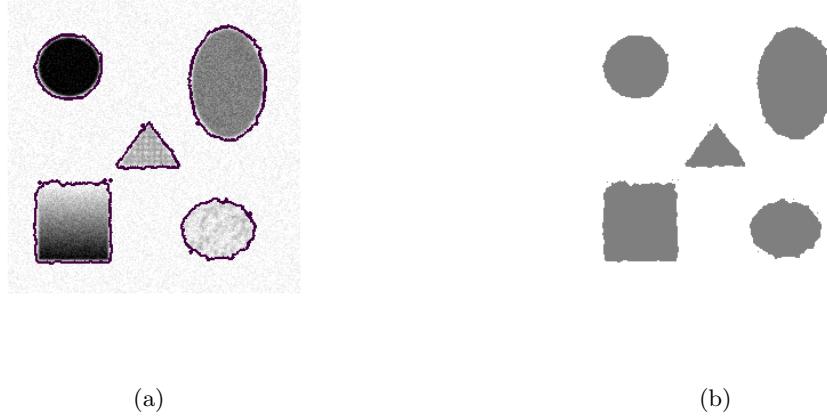


Figure 8.11: Proposed method applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets

shows the final stage of contour evolution, and figure 8.11(b) shows the corresponding segmentation result. The proposed method successfully separates all five objects, even the horizontal ellipse with texture pattern, from the background without significant artifacts. This experiment shows how robust the proposed method is in the segmentation problems of non-uniform images.

Figure 8.12 shows an example of outdoor RGB images. This color image consists of three



Figure 8.12: An outdoor RGB image

bands: red, green, blue. In the picture, there is a gray squirrel in the green grass field. The back of the squirrel shows a mix of gray color, the stomach side of the squirrel shows a mix of pink, and the tail shows a mix of pink and brown. Due to the texture of fur, there are a significant number of narrow shadows and sharp edges along the whole body of the squirrel as well as the grass field. The object of this experiment is to separate the squirrel from the grass field,

i.e. background. This is a difficult segmentation problem because the color of neither squirrel or grass field is uniform. Although this non-uniform texture is a common problem found in any outdoor color images, it makes image segmentation very difficult.

Figure 8.13 shows the result of method 1 applied on figure 8.12. Figure 8.13(a) shows



Figure 8.13: Method 1 applied to an outdoor RGB image: (a) the final stage of contour evolution, (b) the segmented subsets

the final stage of contour evolution, and figure 8.13(b) shows the corresponding segmentation result. Because of the strong sharp texture pattern on the squirrel and the grass field, method 1 produces a lot of noise-like artifacts but cannot properly separate the squirrel from the background. Both the grass field and the squirrel are too complicated to be represented as a constant number due to the presence of many small edges and shadows.

Figure 8.14 shows the result of method 2 applied on figure 8.12. Figure 8.14(a) shows the final stage of contour evolution, and figure 8.14(b) shows the corresponding segmentation result. Method 2 shows a better segmentation result than method 1 successfully separating the squirrel from the grass field. However, there are still a few artifacts on the background due to the small edges and shadows of grass field.

Figure 8.15 shows the result of the proposed method applied on the same image. Figure 8.15(a) shows the final stage of contour evolution, and figure 8.15(b) shows the corresponding segmentation result. The proposed method successfully separates the squirrel from the grass field, and the number of noise-like artifacts is much lower than method 2 as well as method 1. This experiment shows that the proposed method is robust in the segmentation problems of multispectral



Figure 8.14: Method 2 applied to an outdoor RGB image: (a) the final stage of contour evolution, (b) the segmented subsets



Figure 8.15: Proposed method applied to an outdoor RGB image: (a) the final stage of contour evolution, (b) the segmented subsets

images with non-uniform subsets, which is common in outdoor color images.

Chapter 9

Half-supervised Active Contour Model using Multivariate Gaussian Mixture Density Functions

In this chapter, we propose a half-supervised¹ active contour model using a parametric density estimation method introduced in section 6.1.1. The proposed method follows the same region-based segmentation principle introduced in chapter 8 except for two major modifications. First, a supervised estimation method, instead of an unsupervised method, estimates the conditional PDF p_i from prior knowledge. Second, p_i is estimated as a mixture of parametric continuous functions instead of a non-parametric discrete function. Section 9.1 discusses the parametric density estimation method, and section 9.2 discusses the half-supervised image segmentation model. Section 9.3 presents the numerical algorithms of the proposed method, and section 9.4 presents the experimental results applied on synthetic and real images.

9.1 Multivariate Gaussian Mixture Density Function Estimated by EM

In chapter 8, we proposed to estimate the conditional PDF p_i of image intensity within a subset Ω_i as a non-parametric discrete function. In this chapter, we propose to estimate p_i as

¹Although there is no such term as *half-supervised*, we refer to the proposed method as a half-supervised method because it requires relatively lower level of prior knowledge compared to the supervised method introduced in the next chapter.

a mixture of continuous parametric density functions

$$p_i \equiv \sum_{k=1}^K \alpha_k p(\mathbf{I}|\boldsymbol{\theta}_k), \quad (9.1)$$

where the number of sub-classes K and the weight of each sub-class $\{\alpha_k\}$ determine the global statistical property of a mixture density function p_i , and the parameter set $\boldsymbol{\theta}_k$ determines the local statistical property of a sub-class k . The PDF of each sub-class is represented as a multivariate Gaussian density function

$$p(\mathbf{I}|\boldsymbol{\theta}_k) = p(\mathbf{I}|\boldsymbol{\mu}_k, \Sigma_k), \quad \forall k, \quad (9.2)$$

where $\boldsymbol{\mu}_k$ and Σ_k respectively denote the mean vector and the covariance matrix of data belong to the sub-class k . A global PDF can be estimated as a mixture of K Gaussian density functions by estimating these four sets of parameters: K , $\{\alpha_k\}$, $\{\boldsymbol{\mu}_k\}$, and $\{\Sigma_k\}$.

The proposed method uses the classic EM method, introduced in section 6.1.1, to estimate the mixture density function p_i from given data samples \mathcal{I} , such as

$$p_i \approx \sum_{k=1}^K \hat{\alpha}_k p(\mathcal{I}|\hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k). \quad (9.3)$$

As mentioned in section 6.1, there are two problems to estimate a mixture density function using the classic EM method. First, classic EM method cannot estimate K . Second, the result of classic EM method highly depends on the initial values of the parameters to be estimated: $\{\alpha_k(t=0)\}$, $\{\boldsymbol{\mu}_k(t=0)\}$, and $\{\Sigma_k(t=0)\}$. In the proposed method, we solve these problems by providing partial information of p_i as a prior knowledge, and then estimate the rest of information from the given data samples and the prior knowledge. Thus, the parameters are estimated by a supervised method instead of an unsupervised method as chapter 8.

A total of four sets of parameters are required to present p_i as a mixture of Gaussian density functions: the number of sub-classes K , the weight of each sub-class $\{\alpha_k\}$, the mean vector of each sub-class $\{\boldsymbol{\mu}_k\}$, and the covariance matrix of each sub-class $\{\Sigma_k\}$. Let us assume that some training samples are available and the statistical properties of both the training samples and test data are similar. We can find the expected values of K and $\{\boldsymbol{\mu}_k\}$ from training samples and provide them as the prior knowledge of supervised density estimation. Then, the proposed EM method needs to estimate only $\{\alpha_k\}$ and $\{\Sigma_k\}$ from the test data. Figure 9.1

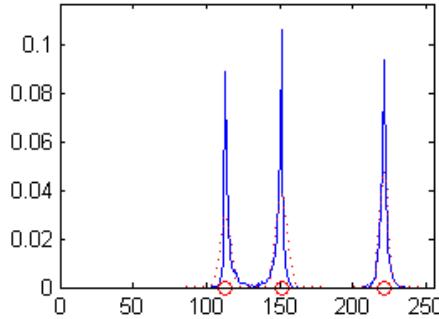


Figure 9.1: Density estimation using a supervised EM method

shows an example of mixture density estimation using the proposed method. The blue solid line presents the histogram of data samples, i.e. the true statistical distribution of the given data. There are three peaks in the histogram, thus the PDF of the given data should be represented as a mixture of at least three Gaussian density functions. Let us assume that two information are available: the expected number of sub-classes $K = 3$ and the expected values of the mean of each sub-class $\{\mu_1, \mu_2, \mu_3\}$ shown as the three red circles. Then, the classic EM method can estimate the weights of each sub-class $\{\hat{\alpha}_1, \hat{\alpha}_2, \hat{\alpha}_3\}$ and the covariance matrix of each sub-class $\{\hat{\sigma}_1, \hat{\sigma}_2, \hat{\sigma}_3\}$ from the given data. The red dotted line presents the estimated parametric mixture density function. The mathematic description of the general EM method was described in section 6.1.1, and the algorithm of the proposed method is presented in section 9.3.

9.2 Half-supervised Image Segmentation

The proposed segmentation method consists of two stages: training and test. During the training stage, a low level of prior knowledge is provided. The test stage consists of two procedures. First, we estimate the statistical distribution of image intensity within each subset as a mixture of continuous parametric density functions. Then, we partition an image using the same energy minimization based image segmentation principle used in the unsupervised multi-dimensional histogram method.

The same global energy function introduced in section 8.1

$$E(\{p_i\}, C) \equiv \sum_i \int_{\Omega_i} e(x, y|p_i) dx dy + \nu |C| \quad (9.4)$$

is minimized with respect to two expressions: $\{p_i\}$, i.e. the representation of the image within a subset Ω_i , and the variational contours C . The objective function $e(x, y|p_i)$ determines the region-based segmentation criteria based on p_i , and the best p_i satisfying the segmentation criteria is a conditional PDF of image intensity on the condition that the image pixel is an element of Ω_i , i.e. $p(\mathbf{I}(x, y)|(x, y) \in \Omega_i)$.

In the proposed method, p_i is represented by four sets of parameters: the number of sub-classes K , the weight of each sub-class $\{\alpha_k\}$, the mean vector of each sub-class $\{\boldsymbol{\mu}_k\}$, and the covariance matrix of each sub-class $\{\Sigma_k\}$. Before the image segmentation, i.e. during the training stage, the expected values of K and $\{\boldsymbol{\mu}_k\}$ are provided from training samples. During image segmentation, i.e. the test stage, $\{\alpha_k\}$ and $\{\Sigma_k\}$ of image intensity within Ω_i are estimated by the classic EM method as described in the previous section. The given parameters, \hat{K} and $\{\hat{\boldsymbol{\mu}}_k\}$, and the estimated parameters, $\{\hat{\alpha}_k\}$ and $\{\hat{\Sigma}_k\}$, represent the conditional PDF p_i of image intensity at each subset, and the estimated p_i provides a force to propagate the variational contours C to the proper position where C divides the entire domain Ω of the image $\mathbf{I}(x, y)$ into multiple subsets $\{\Omega_i\}$. Figure 9.2 and 9.3 show an example of the proposed half-supervised segmentation. Figure 9.2(a) shows an RGB image consisting of two military



Figure 9.2: A complicated synthetic RGB image: (a) an RGB image with two camouflage patterns, (b) the ground truth image

camouflage patterns where each of the pattern consists of at least three different paints, and figure 9.2(b) shows the two subsets of the given image as $\{\Omega_1, \Omega_2\}$. Since both camouflage patterns consist of three different paints, the PDF of image intensity for each subset should be a mixture of at least three sub-classes. Figure 9.3 shows the example of training stage of the

proposed method. The vector-valued image intensities at six points are measured from training



Figure 9.3: Training stage of the proposed method: (a) 6 samples measured for background, (b) 3 samples measured for the rectangle

samples, and assigned as $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \boldsymbol{\mu}_4, \boldsymbol{\mu}_5, \boldsymbol{\mu}_6\}$ for the background Ω_1 in figure 9.3(a). Three samples are measured from training samples, and assigned as $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3\}$ for the rectangle Ω_2 in figure 9.3(b). During the image segmentation, $\{\hat{\alpha}_k\}$ and $\{\hat{\Sigma}_k\}$ are estimated for both subsets $\{\Omega_1, \Omega_2\}$ from the given expected values of \hat{K} and $\{\hat{\mu}_k\}$ and the image intensity of each subset $\{\mathbf{I}(x, y) | \forall (x, y) \in \Omega_i\}$.

The minimization of E with respect to C smoothes the variational boundaries C minimizing the length $|C|$. We use the same level set theory introduced in section 8.3 to implement the proposed active contour model.

The proposed half-supervised image segmentation method performs better than the unsupervised image segmentation method introduced in the last chapter particularly if the image has distinctive non-uniform sub-regions as shown in figure 9.3. Since the statistical property of each subset must be represented as a mixture density function, the segmentation of the images like figure 9.3 is extremely difficult for unsupervised segmentation methods. The supervised density estimation improves the classic EM method providing more robust segmentation capability, and the parametric mixture density function provides more flexibility than the histogram density function.

9.3 Algorithm

Table 9.1 shows the input and output variables used in the proposed segmentation method. A multispectral image $\mathbf{I}(x, y)$ is given as the input data. The number of level set functions, J , the number of level set functions $\{\phi_j(x, y)\}$ = the number of active contours $\{C_j\}$. Table 9.1: The input and output variables used in the half-supervised multivariate Gaussian mixture density method

<ul style="list-style-type: none"> • Input
<ul style="list-style-type: none"> – $\mathbf{I}(x, y) : \Omega \rightarrow \Re^B$, a multispectral image.
<ul style="list-style-type: none"> – J, the number of level set functions $\{\phi_j(x, y)\} =$ the number of active contours $\{C_j\}$.
<ul style="list-style-type: none"> – ν, a parameter to control the regularity of contours.
<ul style="list-style-type: none"> – Δt, the time step of iterative processing.
<ul style="list-style-type: none"> – $\{\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_{2^J}\}$, training samples for each subset.
<ul style="list-style-type: none"> • Output
<ul style="list-style-type: none"> – $S(x, y) : \Omega \rightarrow \{1, \dots, 2^J\}$, the segmented image

which is equivalent to the number of contours, is set as J . ν controls the regularity of contour evolution by controlling the relative balance between segmentation force and regularity force. Δt determines the convergence speed. The expected number \hat{K} of sub-classes and the expected mean vectors $\{\hat{\mu}_k\}$ of sub-classes for each subset is measured from the training samples \mathcal{I}_i for each subset Ω_i , and they provide prior information on supervised density estimation for $\{\hat{\alpha}_k\}$ and $\{\hat{\Sigma}_k\}$. After the proposed active contours converge, the algorithm produces the segmented image $S(x, y)$ indicating each subset with different numbers.

The proposed active contour algorithm consists of three stages: training, initialization, and contour evolution (test). During the training stage, the expected number \hat{K} of sub-classes and the expected mean vectors $\{\hat{\mu}_k\}$ of sub-classes are measured from the training samples \mathcal{I}_i and assigned for each subset Ω_i . The description of function `priorInfo()` is presented in section 9.2. During the initialization stage, J sets of initial contours $\{C_j\}$ are given either manually or from a defined set, and corresponding level set functions are initialized as the signed distance from each (x, y) to the closest contours. During the iterative contour evolution, the mean curvature $\kappa_j(x, y)$ and the unit step function $H_j(x, y)$ are computed from each $\phi_j(x, y)$.

The mathematical description of function `meanCurvature(·)` and `unitStep(·)` are introduced respectively in equation 3.10 and 7.3. Then, binary identity functions $\chi_i(x, y)$ for each subset Ω_i are computed. The mathematical description of function `identityFunction(·)` is introduced in equation 7.4. The weights $\{\hat{\alpha}_k\}$ and covariance matrix $\{\hat{\Sigma}_k\}$ of sub-classes are estimated from the given $\{\hat{\mu}_k\}$ and the image intensity measured within each subset $\mathbf{I}(x, y)\chi_i(x, y)$ using the classic EM method. The mathematical description of function `EM(·)` is introduced in section 6.1.1, and the numerical algorithm is presented in algorithm 4. The estimated $\{\hat{\alpha}_k\}$ and $\{\hat{\Sigma}_k\}$ and the given $\{\hat{\mu}_k\}$ determine the conditional PDF of image intensity p_i , and p_i updates the objective function $e_i(x, y)$ for Ω_i . The level set functions $\{\phi_j(x, y)\}$ are finally updated based on $\chi_i(x, y)$ and $e_i(x, y)$.

Algorithm 3: Half-supervised Active Contour Model using Multivariate Gaussian Mixture Density Functions

Training

for $i = 1 \cdots 2^J$ **do**
 | $[\hat{K}, \{\hat{\mu}_k\}]_i = \text{priorInfo}(\mathcal{I}_i)$: The expected number and the means of sub-classes

Initialization

$t = 0$.

for $j = 1 \cdots J$ **do**

$C_j(x(s), y(s), t = 0)$: Initialize contours.

$\phi_j(x, y, t = 0) = \pm D((x, y), \mathbb{N}_{(x,y)}(C_j))$: Initialize level set functions.

Contour evolution

repeat

for $j = 1 \cdots J$ **do**

$\kappa_j(x, y) = \text{meanCurvature}(\phi_j(x, y))$

$H_j(x, y) = \text{unitStep}(\phi_j(x, y))$

Half-supervised region-based segmentation

for $i = 1 \cdots 2^J$ **do**

$\chi_i(x, y) = \text{identityFunction}(\{H_j(x, y), \forall j\})$

$[\{\hat{\alpha}_k\}, \{\hat{\Sigma}_k\}]_i = \text{EM}(\mathbf{I}(x, y)\chi_i(x, y)|[\hat{K}, \{\hat{\mu}_k\}]_i)$

$e_i(x, y) = -\log p(\mathbf{I}(x, y)|[\hat{K}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}]_i)$

Update level set function

for $j = 1 \cdots J$ **do**

$\phi_j(x, y) = \phi_j(x, y) + \Delta t \delta(\phi_j(x, y)) \left[\nu \kappa(\phi_j(x, y)) - \sum_{i=1}^{2^J} e_i(x, y) \frac{\partial \chi_i}{\partial H_j} \right]$

$t = t + \Delta t$

until until the evolution of contour converges

Segmentation result

$$S(x, y) = \sum_{i=1}^{2^J} \chi_i(x, y)$$

Table 9.2 shows the input and output variables used in the function $\text{EM}(\mathcal{I}|K, \{\mu_k\})$. The supervised EM method takes three inputs, data samples \mathcal{I} , the number of sub-classes K , and the mean vector of each sub-class $\{\mu_k\}$, and produces two outputs, the estimated weight $\{\hat{\alpha}_k\}$ and covariance matrix $\{\hat{\Sigma}_k\}$ of each sub-class.

Table 9.2: The input and output variables used in the estimation of a mixture of multivariate Gaussian density functions using a classic EM method

<ul style="list-style-type: none"> • Input <ul style="list-style-type: none"> – $\mathcal{I} = \{\mathbf{I}(1), \mathbf{I}(2), \dots, \mathbf{I}(n), \dots, \mathbf{I}(N)\}$, $\mathbf{I} \in \Re^B$, data samples. – K: the number of sub-classes. – $\{\boldsymbol{\mu}_k\}$: the mean vector of each sub-class. • Output <ul style="list-style-type: none"> – $\{\hat{\alpha}_k\}$: the estimated weight of each sub-class. – $\{\hat{\Sigma}_k\}$: the estimated covariance matrix of each sub-class.
--

The proposed EM method estimates the output values through an iterative process consisting of three steps: expectation (E-step), maximization (M-step), and evaluation. During E-step, the expected posterior probabilities of the given data samples based on the current $\hat{\Sigma}_k$ and given $\boldsymbol{\mu}_k$ for each sub-class k are estimated and updated. During M-step, two output parameters, $\hat{\alpha}_k$ and $\hat{\Sigma}_k$, which maximize the likelihood function $Q(t)$, are estimated for each sub-class k . During the evaluation step, we check the value of $Q(t)$, and stop the iteration if $Q(t)$ has converged.

Algorithm 4: Half-supervised classic EM Algorithm

Initialization

```

for  $k = 1 \dots K$  do
     $\alpha_k = 1/K$ 
     $\Sigma_k$ : identity matrix
     $Q(0) = \infty, \Delta Q = \infty$ 

```

Estimation

```

while  $\Delta Q \geq \epsilon$  do
    E-step: update posterior probabilities.
        for  $k = 1 \dots K$  do
            for  $n = 1 \dots N$  do  $u_k(n) = p(\mathbf{I}(n)|\boldsymbol{\mu}_k, \hat{\Sigma}_k)$ 
            for  $k = 1 \dots K$  do
                for  $n = 1 \dots N$  do  $w_k(n) = \hat{\alpha}_k u_k(n) / \sum_{j=1}^K \hat{\alpha}_j u_j(n)$ 
    M-step: update parameter set  $\Theta = [\{\hat{\alpha}_k\}, \{\hat{\Sigma}_k\}]$ .
        for  $k = 1 \dots K$  do  $\hat{\alpha}_k = \frac{1}{N} \sum_{n=1}^N w_k(n)$ 
         $\{\hat{\alpha}_k\} = \hat{\alpha}_k / \sum_{j=1}^K \hat{\alpha}_j, \forall k$ 
        for  $k = 1 \dots K$  do
             $\hat{\Sigma}_k = \frac{\sum_{n=1}^N w_k(n)(\mathbf{I}(n)-\boldsymbol{\mu}_k)(\mathbf{I}(n)-\boldsymbol{\mu}_k)^T}{\sum_{n=1}^N w_k(n)}$ 
    Evaluation: check the criterion to stop the iteration.
     $Q(t) = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K w_k(n) \log \hat{\alpha}_k u_k(n)$ 
     $\Delta Q = |Q(t) - Q(t-1)|$ 
     $t = t + 1$ 

```

9.4 Experiments

In this section, we apply the proposed segmentation method to four different images, and compare the results of the proposed method to the results of other active contour models. The same region-based active contour models used in the last chapter were used as benchmarks. All three methods including the proposed method determine the segmentation criteria based on the statistical property of image intensity measured within a subset. Method 1 and 2 are unsupervised methods, while the proposed method uses prior knowledge from the given training samples. The experiments of all three methods use the same pre-defined initial contours. The detail of mathematical and analytical descriptions of both benchmarks methods are presented

in section 3.4.

Figure 9.4 shows an example of synthetic RGB images with complicated non-uniform patterns. There is a rectangle in the middle of the image in figure 9.4(a). Figure 9.4(b) shows the



Figure 9.4: A complicated synthetic RGB image: (a) an RGB image with two camouflage patterns, (b) the ground truth image

ground truth information of the given image. The object of this experiment is to separate the rectangle, class 2, from the background, class 1. This is a difficult segmentation problem because the statistical distribution of both the rectangle and the background are complicated due to the military camouflage patterns consisting of multiple paints. Also, the image is manipulated, so the mean vectors of both interior and exterior are identical as $[R, G, B] = [163, 164, 77]$.

Figure 9.5 shows the result of method 1 applied on figure 9.4. Figure 9.5(a) shows the final stage of contour evolution, and figure 9.5(b) shows the corresponding segmentation result. Method 1 cannot separate the rectangle as a single object because it separates the rectangle as two sub-regions, bright and dark, and the same effect also appears on the background. The dark sub-class and the bright sub-class of the rectangle stay too far sway in the image intensity space, so the average value of the intensity cannot represent the subset. The same problem appears on the background.

Figure 9.6 shows the result of method 2 applied on figure 9.4. Figure 9.6(a) shows the final stage of contour evolution, and figure 9.6(b) shows the corresponding segmentation result. Method 2 cannot separate the rectangle as a single object because of the same reason that



Figure 9.5: Method 1 applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets



Figure 9.6: Method 2 applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets

method 1 fails. Experiments shown in figure 9.5 and 9.6 show the limitation of unsupervised segmentation methods on the images with non-uniform sub-regions.

We applied the proposed half-supervised segmentation method on the same image. Figure 9.7 shows the training stage of the proposed method. Training samples for both the rectangle and the background are provided. Six vector-valued image intensities are measured from the training samples and assigned as $\{\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_6\}$ for the background in figure 9.7(a). Three vector-valued image intensities are measured from the training samples and assigned as $\{\hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3\}$ for the rectangle in figure 9.7(b). Figure 9.8 shows the the result of the proposed

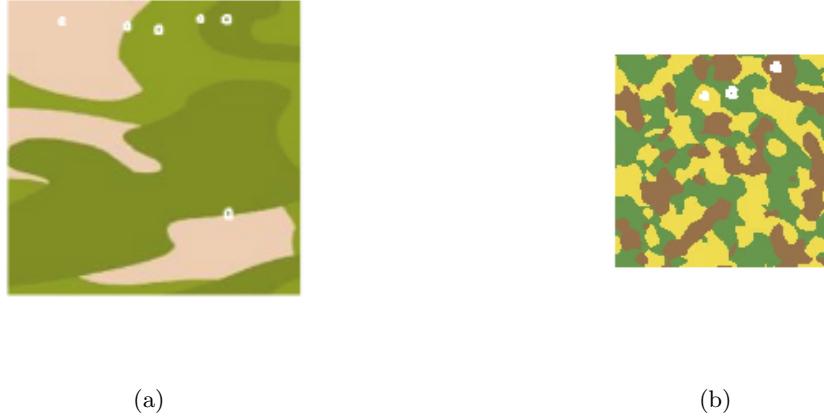


Figure 9.7: Training stage of the proposed method: (a) 6 samples measured for background, (b) 3 samples measured for the core

method. Figure 9.8(a) shows the final stage of contour evolution, and figure 9.8(b) shows the

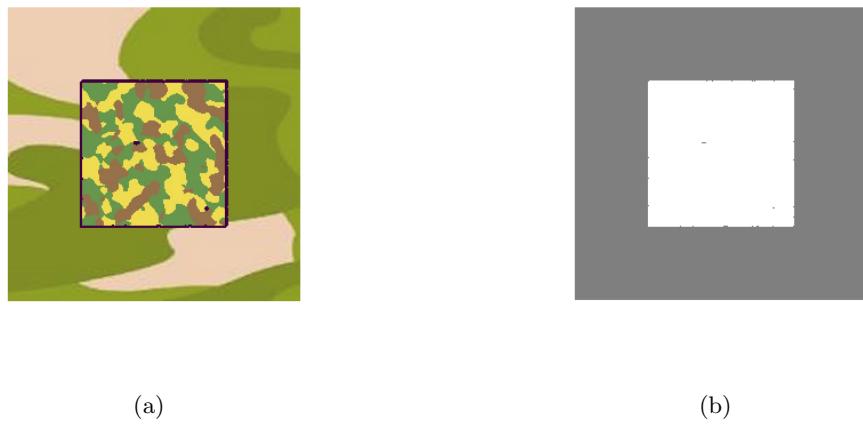


Figure 9.8: Proposed method applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets

corresponding segmentation result. The proposed method successfully separates the rectangle from the background without any significant artifact.

For more precise comparison, the statistical distribution of image intensity within both class 1 and 2 are measured. Figure 9.9 shows the statistical distribution of image intensity measured at green channel within class 1, which is supposed to be the background in the segmentation result. Figure 9.9(a), 9.9(b), and 9.9(c) respectively present the results of method

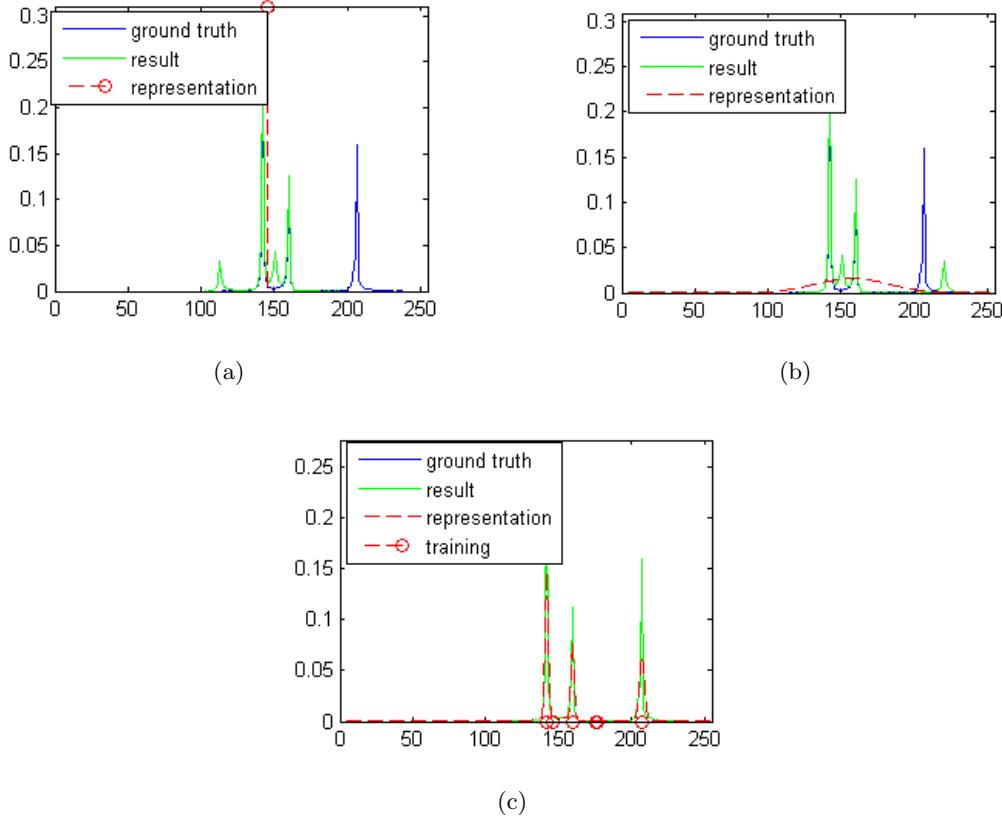


Figure 9.9: Statistical distribution of image intensity within the background measured at green channel: (a) method 1, (b) method 2, (c) proposed method

1, method 2, and the proposed method. The blue solid lines of all three graphs present the statistical distribution of image intensity measured at the background, and it shows that the camouflage pattern of background consists of at least three different paints. The green solid lines of all three graphs present the same statistics measured within class 1 in the segmentation results. As the green solid line exists closer to the blue solid line, it presents better result. In figure 9.9(a), the dotted red line with a circle presents the representation of the subset used in method 1, i.e. the mean of green solid line. The measured statistics are very different from the ground truth because the constant representation is not sufficient to represent the non-uniform sub-region consisting of three different paints. In figure 9.9(b), the dotted red line presents the representation of the subset used in method 2, i.e. the Gaussian density function of image intensity measured. The measured statistics is very different from the ground truth because of the same reason as figure 9.9(a). The red circles in figure 9.9(c) present $\{\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_6\}$ measured from training samples. They are the centers of each sub-class. The dotted red line

presents the estimated parametric mixture density function at the final stage of contour evolution. Although the estimated mixture density function is not identical to the true statistical distribution, i.e. the blue solid line, it successfully finds the position of peaks producing a good segmentation result. The estimated weights $\{\hat{\alpha}_1, \hat{\alpha}_2, \dots, \hat{\alpha}_6\}$ help to ignore insignificant sub-classes and to emphasize significant sub-classes. The measured statistics and the ground truth are almost identical. Figure 9.10 shows the same measure for class 2, the rectangle. Both

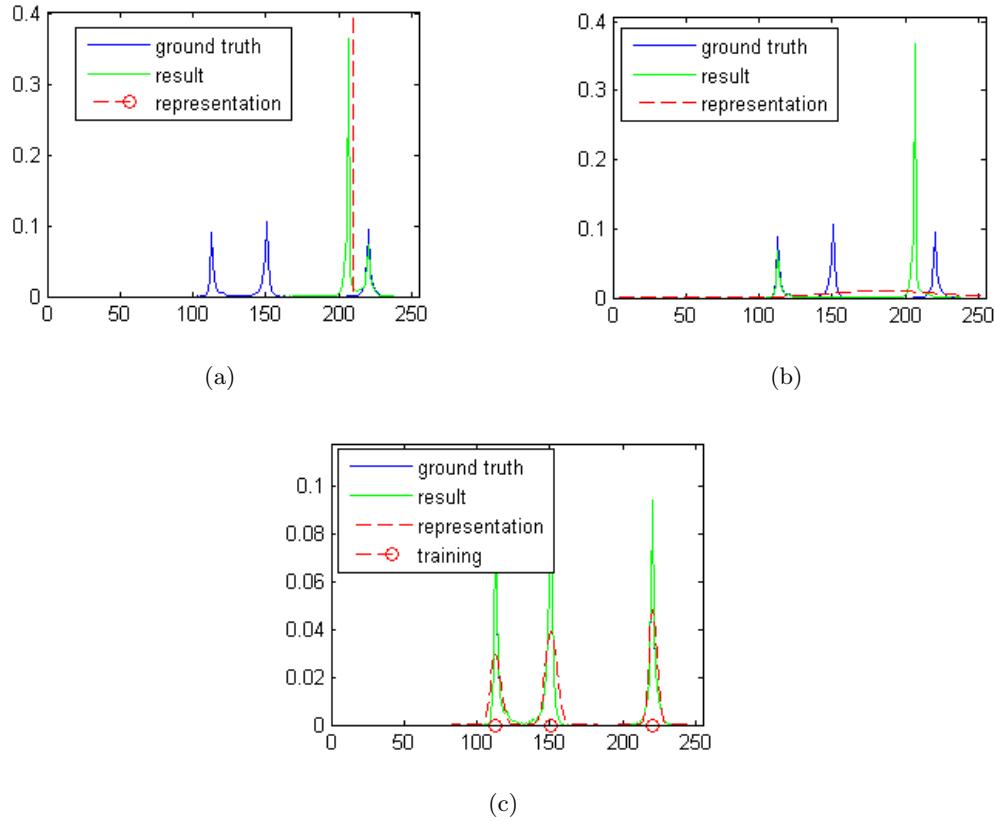


Figure 9.10: Statistical distribution of image intensity within the core measured at green channel: (a) method 1, (b) method 2, (c) proposed method

method 1 and 2 cannot produce satisfying results, while the proposed method successfully estimates a mixture density function, which is close to the true statistical distribution of image intensity, from $\{\hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3\}$. The measured statistics and the ground truth are almost identical.

Figure 9.11 shows an RGB image captured from a virtual 3D scene generated by an ATR simulator. The background is reconstructed from an air-borne range data, and the corresponding color picture is put on the reconstructed surface as texture. We also reconstruct a tank

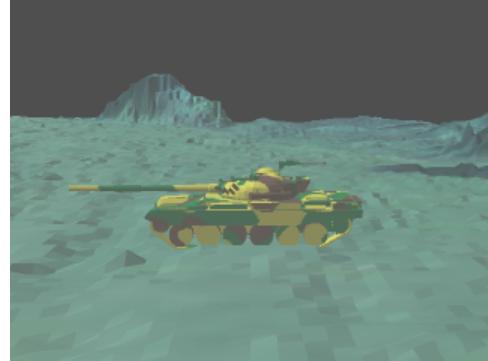


Figure 9.11: A complicated synthetic RGB image

model, and put the camouflage pattern shown in figure 9.7(b) on its surface. The object of this experiment is to extract the shape of the tank from the scene as a target. This image is another difficult problem because there are a lot of shades on the surface of both the tank and the background as well as the complicated camouflage pattern.

Figure 9.12 shows the result of method 1 applied on figure 9.11. Figure 9.12(a) shows the



Figure 9.12: Method 1 applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets

final stage of contour evolution, and figure 9.12(b) shows the corresponding segmentation result. Method 1 identifies the sky as the target, the white subset. Also, method 1 cannot extract the whole body of the tank from the scene because of the complicated camouflage pattern. Method 2 also produces an undesirable result as shown in figure 9.13. The mountains and other areas in the background are identified as target, the dark subset.



Figure 9.13: Method 2 applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets

We applied the proposed half-supervised segmentation method on the same image. Figure 9.14 shows the training stage of the proposed method. We used the same image as the

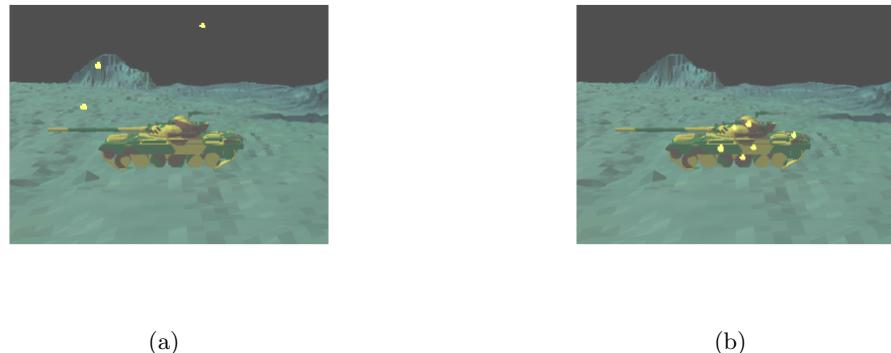


Figure 9.14: Training stage of the proposed method: (a) 3 samples measured for background, (b) 6 samples measured for the tank

training samples for this experiment. Three vector-valued image intensities are measured from a mountain, sky and the field, and assigned for the background. Six vector-valued image intensities are measured from the tank, and assigned for the tank. After applying the proposed method on the same image, we successfully extract the almost perfect shape of the tank from the background as shown in figure 9.15. Figure 9.15(a) shows the final stage of contour evolution, and figure 9.15(b) shows the corresponding segmentation result. In figure 9.15(b), we can see very sophisticated detail of the tank. This experiment shows the robustness of the proposed method in ATR problems.



Figure 9.15: Proposed method applied to a complicated synthetic RGB image: (a) the final stage of contour evolution, (b) the segmented subsets

Figure 9.16 shows an example of outdoor gray scale images with texture. There are two

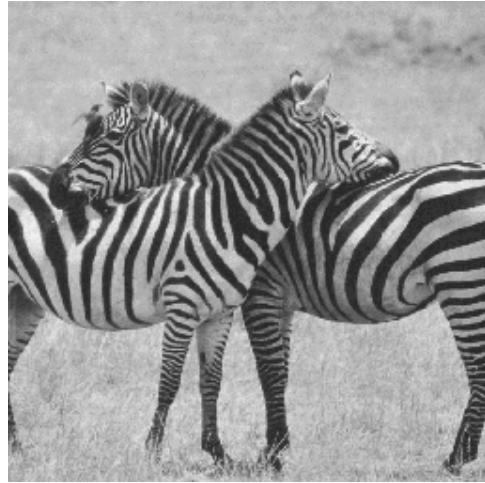


Figure 9.16: A complicated outdoor gray image; two zebras

zebras standing close on the field. The zebras consist of white and black stripes, while the background is textured gray. The object of this experiment is to extract the shape of zebras from the background. This is a difficult segmentation problem because of the complicated patterns of zebras. The white stripes of zebras are brighter than the background, while the black stripes of zebras are darker than the background. Also, some portions of zebras, e.g. white stripes under the shade, have similar intensity to the background. The statistical distribution of image intensity within the zebras must be represented as a mixture of more than two sub-classes.

Figure 9.17 shows the result of method 1 applied on figure 9.16. Figure 9.17(a) shows



Figure 9.17: Method 1 applied to the zebra image: (a) the final stage of contour evolution, (b) the segmented subsets

the final stage of contour evolution, and figure 9.17(b) shows the corresponding segmentation result. Although method 1 identifies the whole background as a subset, it separates the zebras into two parts: bright sub-regions and dark sub-regions. Also, method 1 identifies the bright sub-regions of zebras as the background.

Figure 9.18 shows the result of method 2 applied on figure 9.16. Figure 9.18(a) shows



Figure 9.18: Method 2 applied to the zebra image: (a) the final stage of contour evolution, (b) the segmented subsets

the final stage of contour evolution, and figure 9.18(b) shows the corresponding segmentation

result. Although method 2 produces better segmentation result than method 1, there are many artifacts on both zebras and background. The sub-regions of zebras whose intensity is similar to the background are identified as background, the white subset, while the sub-regions of background whose intensity is darker than the other parts of background are identified as zebras, the black subset.

Figure 9.19 shows the training stage of the proposed segmentation method. We used the

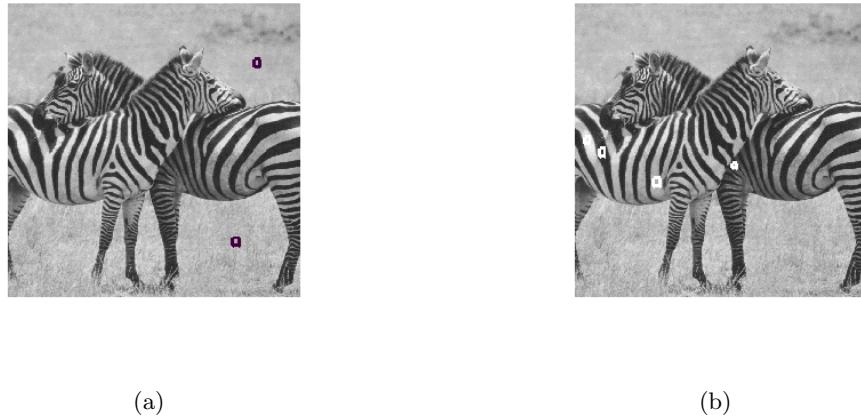


Figure 9.19: Training stage of the proposed method: (a) 2 samples measured for background, (b) 4 samples measured for zebras

same image as the training samples for this experiment. The image intensities of two samples are measured and assigned as $\{\hat{\mu}_1, \hat{\mu}_2\}$ for the background. Four image intensities from bright white stripes, dark white stripes, bright black stripes, and dark black stripes are measured and assigned as $\{\hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3, \hat{\mu}_4\}$ for the zebras. After applying the proposed method on the same image, the proposed method successfully extracts the shape of two zebras from the background as shown in figure 9.20. Figure 9.20(a) shows the final stage of contour evolution, and figure 9.20(b) shows the corresponding segmentation result. Although there are still a few artifacts on the zebras, the number of artifacts has been significantly reduced, and there are no artifacts in the background. The artifacts found within zebras appear because the image intensity of those sub-regions is identical to the background. This experiment shows the robustness and the limitation of the proposed segmentation method.

Figure 9.21 shows an example of indoor RGB images with a strong texture. There is a human hand and a donut-shaped object on the texture background. The object of this experi-



Figure 9.20: Proposed method applied to the zebra image: (a) the final stage of contour evolution, (b) the segmented subsets



Figure 9.21: A complicated indoor RGB image; hand

ment is to extract the hand and the donut-shaped object from the background and to identify them as different subsets. Thus, the ideal segmentation result should have three subsets as $\{\Omega_{\text{hand}}, \Omega_{\text{object}}, \Omega_{\text{background}}\}$. This is a difficult segmentation problem because it requires not only segmentation but also pattern classification. The textured background is another difficult problem. In order to partition the given image into three subsets, all three active contour models use two level set functions $\{\phi_1(x, y), \phi_2(x, y)\}$. The mathematical description of 4-phase active contour is presented in chapter 7.

Figure 9.22 shows the result of method 1 applied on figure 9.21. Figure 9.22(a) shows the final stage of contour evolution, and figure 9.22(b) shows the corresponding segmentation

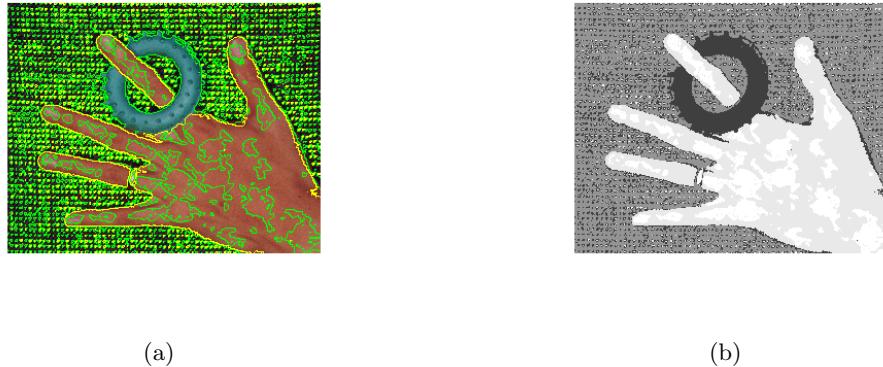


Figure 9.22: Method 1 applied to the hand image: (a) the final stage of contour evolution, (b) the segmented subsets

result. Method 1 produces a very poor segmentation result in this case. Because of the texture, a lot of artifacts appear on the background. Also, the use of two level set functions produces undesired local minima resulting in a lot of artifacts on the hand.

Figure 9.23 shows the result of method 2 applied on figure 9.21. Figure 9.23(a) shows the

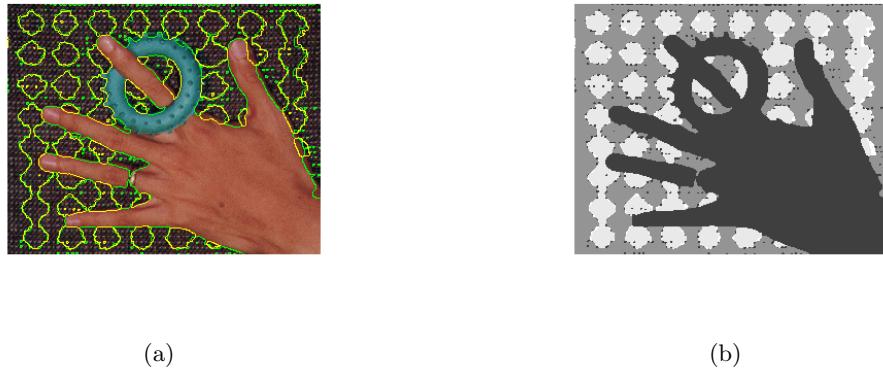


Figure 9.23: Method 2 applied to the hand image: (a) the final stage of contour evolution, (b) the segmented subsets

final stage of contour evolution, and figure 9.23(b) shows the corresponding segmentation result. Although method 2 produces a relatively cleaner segmentation on the textured background, it generates a lot of artifacts on the background, which is the local minima of two level set functions. Method 2 also identifies the hand and the donut-shaped object as the same subset. A smart choice of initial contours may avoid the artifacts caused by local minima of two level set

functions, but it requires prior knowledge of given image.

Figure 9.24 shows the training step of the proposed segmentation method. We used the

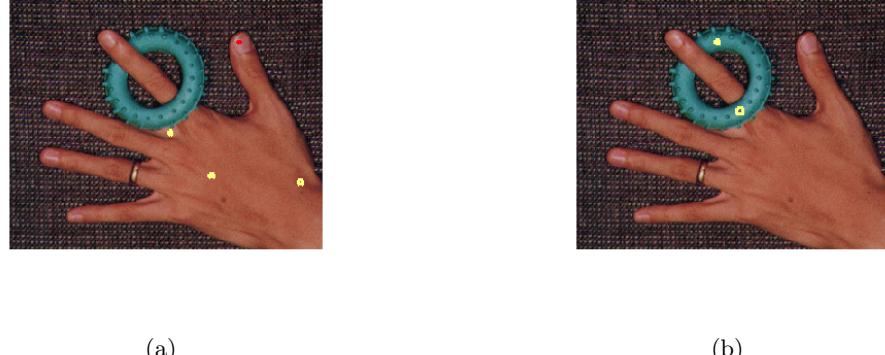


Figure 9.24: Training stage of the proposed method: (a) 3 samples measured for hand, (b) 2 samples measured for the donut shaped object

same image as the training samples for this experiment. The vector-valued image intensities of three samples are measured from hand and assigned as $\{\hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3\}$ for the hand. Two samples are measured from the donut-shaped object, and assigned as $\{\hat{\mu}_1, \hat{\mu}_2\}$ for the object. For the background, we did not provide any information, but assumed the background can be represented as a multivariate Gaussian density function. After applying the proposed method on the same image, the proposed method successfully extracts the shape of hand and the donut-shaped object from the background, and identifies them as three different subsets $\{\Omega_{hand}, \Omega_{object}, \Omega_{background}\}$ as shown in figure 9.25. Although there are still a few artifacts, identified as $\Omega_{unknown}$, the proposed method shows the best result compared to the benchmarks. This experiment shows that the proposed method can be used for not only image segmentation but also pattern classification.

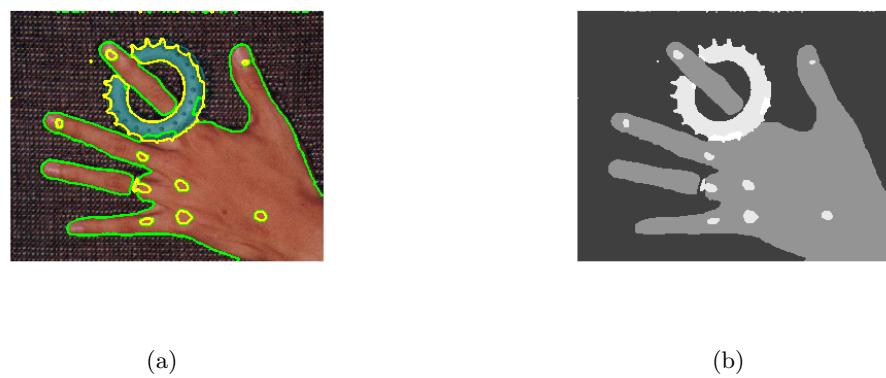


Figure 9.25: Proposed method applied to the hand image: (a) the final stage of contour evolution, (b) the segmented subsets

Chapter 10

Supervised Active Contour Model using Multivariate Gaussian Mixture Density Functions

In this chapter, we propose a supervised active contour model using the parametric density estimation method introduced in section 6.1.2. The proposed active contour model estimates the statistical information of training samples for each subset, and applies the pre-estimated information into image segmentation without iterative density estimation during contour evolution. Section 10.1 discusses the supervised image segmentation model, and section 10.2 discusses the parametric density estimation method. Section 10.3 presents the numerical algorithms of the proposed method, and section 10.4 presents the experimental results applied on synthetic and real images.

10.1 Supervised Image Segmentation

The proposed segmentation method consists of two stages: training and test stages. During the training stage, an advanced EM method with MML estimates the PDF of training samples as a mixture of multivariate Gaussian density functions. During the test stage, the proposed segmentation method assigns the pre-estimated density functions as the conditional PDF of image intensity for each subset, and partitions the given image without any density estimation process.

Let us assume that training samples for each subset are available and the statistical distribution of training samples is similar to the statistical distribution of image intensity within each subset in the image. Then, we can estimate the conditional PDF of image intensity from the training samples, such as

$$p(\mathbf{I}(x, y) | (x, y) \in \Omega_i) \approx p(\mathbf{I} | \mathbf{I} \in \mathcal{I}_i), \quad \forall i \quad (10.1)$$

where $\mathcal{I}_i = \{\mathbf{I}_1, \dots, \mathbf{I}_n, \dots, \mathbf{I}_N\}$ denotes the ideally sampled training samples or a statistical template of the object, which is supposed to be isolated as a subset Ω_i in the result of the segmentation. $p(\mathbf{I} | \mathbf{I} \in \mathcal{I}_i) : \Re^B \rightarrow \Re$ denotes a multivariate conditional PDF of a vector-valued image intensity \mathbf{I} on the condition that the multispectral image pixel is an element of training samples \mathcal{I}_i . The detail on the density estimation of $p(\mathbf{I} | \mathbf{I} \in \mathcal{I}_i)$ will be discussed in the next section.

The proposed segmentation method uses the same global energy minimization scheme as the other two methods introduced in chapter 8 and 9. However, the minimization of the global energy function E with respect to the statistical representation p_i in equation 9.4 and 8.1 of each subset is not necessary in the proposed method because the best fitting p_i is already given as $p(\mathbf{I} | \mathbf{I} \in \mathcal{I}_i)$ from training samples instead of the given image. The global energy function can be simplified as a function of variational contours C only, such as

$$E(C) \equiv \sum_i \int_{\Omega_i} -\log p(\mathbf{I}(x, y) | \mathbf{I}(x, y) \in \mathcal{I}_i) dx dy + \nu |C|, \quad (10.2)$$

where the pre-estimated $p(\mathbf{I}(x, y) | \mathbf{I}(x, y) \in \mathcal{I}_i)$ determines the segmentation criteria and provides a force to propagate the variational contours C to the proper position where C divides the entire domain Ω of the image $\mathbf{I}(x, y)$ into multiple subsets $\{\Omega_i\}$. This is a supervised image segmentation. Then, the minimization of E with respect to C smoothes the variational boundaries C minimizing the length $|C|$.

The Euler-Lagrange equation obtained by minimizing the energy function E with respect to level set functions $\Phi = \{\phi_1, \dots, \phi_j, \dots, \phi_J\}$ is given by

$$\frac{\partial \phi_j(x, y)}{\partial t} = \delta_j \left[\nu \kappa_j + \sum_{i=1}^{2^J} \log p(\mathbf{I}(x, y) | \mathbf{I}(x, y) \in \mathcal{I}_i) \frac{\partial \chi_i}{\partial H_j} \right], \quad \forall j, \quad (10.3)$$

where $\delta_j \equiv \delta_\epsilon(\phi_j(x, y))$ controls the width of update $\partial\phi_j/\partial t$, $p(\mathbf{I}(x, y)|\mathbf{I}(x, y) \in \mathcal{I}_i)$ provides a force to propagate contours C_j , and $\partial\chi_i/\partial H_j$ determines the region where ϕ_j updates. The implementation of the proposed active contour model is identical to the two other proposed active contour models except that the proposed method does not need to compute $p(\mathbf{I}(x, y)|\mathbf{I}(x, y) \in \mathcal{I}_i)$ at each iteration.

The proposed active contour model estimates the PDF of training samples prior to the contour evolution, and does not estimate the PDF from the image but only apply the pre-estimated PDF for image segmentation. This supervised approach simplifies the segmentation procedure reducing the convergence time of active contours. As long as the training samples are reliable, the segmentation results will be also more robust than the results of the two methods introduced in chapter 8 and 9 as well as traditional segmentation methods. However, the results will not be consistent or robust if the training samples are unreliable.

10.2 Multivariate Gaussian Mixture Density Function Estimated by EM with MML

In this chapter, we represent the conditional PDF of vector-valued image intensity for being an element of a subset $p(\mathbf{I}(x, y)|(x, y) \in \Omega_i)$ in the same way that we estimate p_i in chapter 9, i.e. a mixture of multivariate Gaussian density functions

$$p(\mathbf{I}(x, y)|(x, y) \in \Omega_i) \equiv \sum_{k=1}^K \alpha_k p(\mathbf{I}(x, y)|\boldsymbol{\mu}_k, \Sigma_k), \quad \forall i, \quad (10.4)$$

except that the mixture density function is estimated from training samples \mathcal{I}_i given by

$$p(\mathbf{I}(x, y)|(x, y) \in \Omega_i) \approx \sum_{k=1}^{\hat{K}} \hat{\alpha}_k p(\mathcal{I}_i|\hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k), \quad \forall i. \quad (10.5)$$

The estimation above is done by estimating four sets of parameters: the number of sub-classes K , the weight $\{\alpha_k\}$, the mean vector $\{\boldsymbol{\mu}_k\}$, and the covariance matrix $\{\Sigma_k\}$.

Since classic EM method cannot estimate the number of sub-classes K , the proposed method uses an advanced EM method proposed by Figueiredo and Jain [146] to estimate a mixture density function from given data samples, i.e. training samples. The advanced EM

method based on the minimum encoding length criteria initiates a high number of sub-classes, and eliminates the weakest sub-class one by one, so it can automatically select the number of sub-classes K . Therefore, all four sets of parameters, \hat{K} , $\{\hat{\alpha}_k\}$, $\{\hat{\mu}_k\}$, $\{\hat{\Sigma}_k\}$, can be estimated from given data samples without any prior knowledge on the data. That is an unsupervised density estimation method estimates parameters, and a supervised segmentation method partitions an image based on the estimated parameters. The detail of mathematic description of this density estimation method is presented in section 6.1.2, and the numerical algorithm is presented in the next section.

10.3 Algorithm

Table 10.1: The input and output variables used in the supervised active contour model using multivariate Gaussian mixture densities

<ul style="list-style-type: none"> • Input <ul style="list-style-type: none"> – $\mathbf{I}(x, y) : \Omega \rightarrow \Re^B$, a multispectral image. – J, the number of level set functions $\{\phi_j(x, y)\} =$ the number of active contours $\{C_j\}$. – ν, a parameter to control the regularity of contours. – Δt, the time step of iterative processing. – $\{\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_{2^J}\}$, training samples for each subset. • Output <ul style="list-style-type: none"> – $S(x, y) : \Omega \rightarrow \{1, \dots, 2^J\}$, the segmented image

Table 10.1 shows the input and output variables used in the proposed segmentation method. A multispectral image $\mathbf{I}(x, y)$ is given as the input data. The number of level set functions, which is equivalent to the number of contours, is set as J . ν controls the regularity of contour evolution by controlling the relative balance between segmentation force and regularity force. Δt determines the convergence speed. The parameters representing each subset Ω_i are estimated from the corresponding training samples \mathcal{I}_i . After the proposed active contours converge, the algorithm produces the segmented image $S(x, y)$ indicating each subset with different numbers.

The proposed active contour algorithm consists of three stages: training, initialization, and contour evolution (test). During the training stage, the four sets of parameters representing the PDF of training samples \mathcal{I}_i are estimated for each subset Ω_i : the mean vector $\{\hat{\mu}_k\}$, the weight $\{\hat{\alpha}_k\}$, and the covariance matrix $\{\hat{\Sigma}_k\}$ of each sub-class, and the number \hat{K} of sub-classes. The mathematical description of density estimation method **EMwMML**(\cdot) is presented in section 6.1.2, and the algorithm is presented in algorithm 6. During the initialization stage, J sets of initial contours $\{C_j\}$ are given either manually or from a defined set, and corresponding level set functions are initialized as the signed distance from each (x, y) to the closest contours. During the iterative contour evolution, the mean curvature $\kappa_j(x, y)$ and the unit step function $H_j(x, y)$ are computed from each level set function $\phi_j(x, y)$. The mathematical description of function **meanCurvature**(\cdot) and **unitStep**(\cdot) are introduced respectively in equation 3.10 and 7.3. Then, binary identity functions $\chi_i(x, y)$ for each subset Ω_i are computed. The mathematical description of function **identityFunction**(\cdot) is given in equation 7.4. There is no density estimation procedure during the contour evolution as the required parameters are already estimated from training samples \mathcal{I}_i . Instead, the proposed method just assigns the pre-estimated parametric mixture density functions $p(\mathbf{I}(x, y)|[\hat{K}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}]_i)$ to each subset Ω_i on the image, and the density functions perform the role of conditional PDF of image intensity. $p(\mathbf{I}(x, y)|[\hat{K}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}]_i)$ updates the objective function $e_i(x, y)$ for Ω_i , consequently updates the level set functions $\{\phi_j(x, y)\}$ based on $\chi_i(x, y)$ and $e_i(x, y)$.

Algorithm 5: Supervised Active Contour Model using Multivariate Gaussian Mixture Density Functions

Training

```

for  $i = 1 \dots 2^J$  do
     $[\hat{K}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}]_i = \text{EMwMML}(\mathcal{I}_i)$ 

```

Initialization

$t = 0.$

```

for  $j = 1 \dots J$  do

```

$C_j(x(s), y(s), t = 0)$: Initialize contours.

```

     $\phi_j(x, y, t = 0) = \pm D((x, y), \mathbb{N}_{(x,y)}(C_j))$ : Initialize level set functions.

```

Contour evolution

```

repeat

```

```

    for  $j = 1 \dots J$  do

```

$\kappa_j(x, y) = \text{meanCurvature}(\phi_j(x, y))$

```

         $H_j(x, y) = \text{unitStep}(\phi_j(x, y))$ 

```

Supervised region-based segmentation

```

        for  $i = 1 \dots 2^J$  do

```

$\chi_i(x, y) = \text{identityFunction}(\{H_j(x, y), \forall j\})$

```

             $e_i(x, y) = -\log p(\mathbf{I}(x, y) | [\hat{K}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}]_i)$ 

```

Update level set function

```

            for  $j = 1 \dots J$  do

```

$\phi_j(x, y) = \phi_j(x, y) + \Delta t \delta(\phi_j(x, y)) \left[\nu \kappa(\phi_j(x, y)) - \sum_{i=1}^{2^J} e_i(x, y) \frac{\partial \chi_i}{\partial H_j} \right]$

$t = t + \Delta t$

until *until the evolution of contour converges*

Segmentation result

$$S(x, y) = \sum_{i=1}^{2^J} \chi_i(x, y)$$

Table 10.2 shows the input and output variables used in the function $\text{EMwMML}(\cdot)$ [146]. The advanced unsupervised EM method takes data samples \mathcal{I} and the maximum and minimum possible number of sub-classes $\{K_{max}, K_{min}\}$ as input, and produces the four sets of parameters representing a mixture of multivariate Gaussian density functions: the mean vectors $\{\hat{\mu}_k\}$, the weights $\{\hat{\alpha}_k\}$, and the covariance matrix $\{\hat{\Sigma}_k\}$ of each sub-class, and the number \hat{K} of sub-classes.

Table 10.2: The input and output variables used in the estimation of a mixture of multivariate Gaussian density functions using an advanced EM method with MML

<ul style="list-style-type: none"> • Input <ul style="list-style-type: none"> – $\mathcal{I} = \{\mathbf{I}(1), \mathbf{I}(2), \dots, \mathbf{I}(n), \dots, \mathbf{I}(N)\}$, $\mathbf{I} \in \Re^B$, data samples. – K_{max}: the maximum possible number of sub-classes. – K_{min}: the minimum possible number of sub-classes. • Output <ul style="list-style-type: none"> – \hat{K}: the estimated number of sub-classes. – $\{\hat{\mu}_k\}$: the estimated mean vector of each sub-class. – $\{\hat{\alpha}_k\}$: the estimated weight of each sub-class. – $\{\hat{\Sigma}_k\}$: the estimated covariance matrix of each sub-class.

The advanced EM method used in the proposed active contour model estimates the output values through an iterative process consisting of four steps: expectation (E-step), maximization (M-step), evaluation, and annihilation. At the initialization stage, the number of sub-classes is initialized as K_{max} . The three other parameters, $\{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}$, are also initialized as appropriate values. During E-step, the expected posterior probabilities of the given data samples based on the current status of parameters $[\hat{\alpha}_k, \hat{\mu}_k, \hat{\Sigma}_k]$ are estimated and updated for each sub-class k . During M-step, four output parameters, $[\hat{K}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}]$, are estimated. Function `eliminate()` eliminates the sub-class with zero weight $\hat{\alpha}_k = 0$, adjusts the number of sub-classes $\hat{K} = \hat{K} - 1$, and rearranges the parameters of remained sub-classes. During the evaluation step, we check the value of metric function \mathcal{L} , and stop the iteration if $\mathcal{L}(t)$ has converged. During the annihilation step, we eliminate the sub-class with the lowest weight.

Algorithm 6: Advanced EM Algorithm with MML

Initialization
 $\hat{K} = K_{max}$

for $k = 1 \dots K$ **do**

- Initialize $\hat{\alpha}_k, \hat{\mu}_k, \hat{\Sigma}_k$.
- for** $n = 1 \dots N$ **do** $u_k(n) = p(\mathbf{I}(n)|\mu_k, \Sigma_k)$
- $\mathcal{L}(0) = \infty, \mathcal{L}_{min} = \infty$

Estimation
 $\text{while } \hat{K} \geq K_{min} \text{ do}$

- while** $\Delta\mathcal{L} \geq \epsilon$ **do**
 - $k = 1$
 - while** $k \leq \hat{K}$ **do**
 - E-step: update posterior probabilities.
 - for** $n = 1 \dots N$ **do** $w_k(n) = \frac{\hat{\alpha}_k u_k(n)}{\sum_{j=1}^{\hat{K}} \hat{\alpha}_j u_j(n)}$
 - M-step: update parameter set $\Theta = [\hat{K}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}]$.
 - $\hat{\alpha}_k = \frac{\max(0, \sum_{n=1}^N w_k(n) - N/2)}{\sum_{j=1}^K \{\max(0, \sum_{n=1}^N w_j(n) - N/2)\}}$
 - $\{\hat{\alpha}_k\} = \hat{\alpha}_k / \sum_{j=1}^K \hat{\alpha}_j, \forall k$
 - if** $\hat{\alpha}_k = 0$ **then**
 - $[\hat{K}, \{u_k\}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}] = \text{eliminate}(\hat{K}, \{u_k\}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\})$
 - else**
 - for** $k = 1 \dots \hat{K}$ **do**
 - $\hat{\mu}_k = \frac{\sum_{n=1}^N \mathbf{I}(n) w_k(n)}{\sum_{n=1}^N w_k(n)}$
 - $\hat{\Sigma}_k = \frac{\sum_{n=1}^N w_k(n)(\mathbf{I}(n) - \hat{\mu}_k)(\mathbf{I}(n) - \hat{\mu}_k)^T}{\sum_{n=1}^N w_k(n)}$
 - for** $n = 1 \dots N$ **do** $u_k(n) = p(\mathbf{I}(n)|\theta_k) = p(\mathbf{I}(n)|\hat{\mu}_k, \hat{\Sigma}_k)$
 - $k = k + 1$
 - Evaluation: check the criterion to stop the iteration.
 - $\mathcal{L}(t) = \frac{M}{2} \sum_{k=1}^K \log \frac{N \hat{\alpha}_k}{12} + \frac{k}{2} \log \frac{N}{12} + \frac{\hat{K}(M+1)}{2} - \sum_{n=1}^N \log \left(\sum_{k=1}^{\hat{K}} \hat{\alpha}_k u_k(n) \right)$
 - $\Delta\mathcal{L} = |\{\mathcal{L}(t) - \mathcal{L}(t-1)\}/\mathcal{L}(t-1)|$
 - if** $\mathcal{L}_{min} > \mathcal{L}(t)$ **then** $\mathcal{L}_{min} = \mathcal{L}(t)$ and $\Theta_{best} = \Theta(t)$

Annihilation: eliminate the weakest sub-class.

- if** $\hat{K} \neq K_{min}$ **then**
 - $\hat{\alpha}_{k^*} = 0$ where $k^* = \arg \min_k \hat{\alpha}_k$
 - $\{\hat{\alpha}_k\} = \hat{\alpha}_k / \sum_{j=1}^K \hat{\alpha}_j, \forall k$
 - $[\hat{K}, \{u_k\}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\}] = \text{eliminate}(\hat{K}, \{u_k\}, \{\hat{\alpha}_k\}, \{\hat{\mu}_k\}, \{\hat{\Sigma}_k\})$

10.4 Experiments

In this section, we apply the proposed segmentation method to three different images, and compare the results of the proposed method to the results of other active contour models. The same region-based active contour models used in the last two chapters were used as benchmarks. All three methods including the proposed method perform the segmentation based on the statistical property of image intensity measured within a subset. Method 1 and 2 are unsupervised methods, while the proposed method determines the statistical information from training samples. The experiments of all three methods use the same pre-defined initial contours. The detail of mathematical and analytical descriptions of both benchmarks methods are presented in section 3.4.

Figure 10.1 shows an example of synthetic gray images with textures. There is a small



Figure 10.1: Synthetic textured image: (a) a textured gray image, (b) the ground truth image

rectangle in figure 10.1(a), and both the rectangle and the background have their own texture pattern. The object of this experiment is to separate the rectangle from the background. Figure 10.1(b) shows the ground truth of the given image. The dark part presents the background, class 1, while the bright part presents the rectangle, class 2.

Figure 10.2 shows the result of method 1 applied on figure 10.1. Figure 10.2(a) shows the final stage of contour evolution, and figure 10.2(b) shows the corresponding segmentation result.

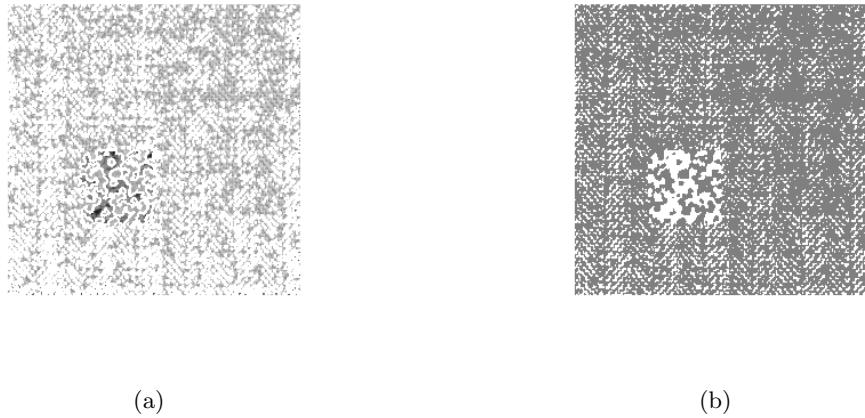


Figure 10.2: Method 1 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets

Method 1 produces extremely many artifacts on both the background and the rectangle because of the complicated texture patterns. Method 1 partitions the image into dark sub-regions and bright sub-regions, but they are not equivalent to the two texture patterns.

Figure 10.3 shows the result of method 2 applied on figure 10.1. Figure 10.3(a) shows the

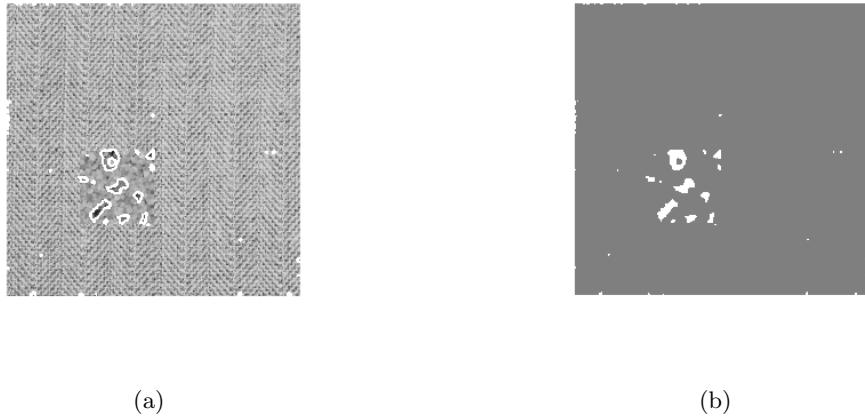


Figure 10.3: Method 2 applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets

final stage of contour evolution, and figure 10.3(b) shows the corresponding segmentation result. Method 2 does not show many artifacts on the background, but cannot extract the shape of the rectangle. Figure 10.2 and 10.3 show that the performance of unsupervised segmentation

methods is limited if the image has non-uniform subsets and different subsets show similar image intensities.

We applied the proposed supervised segmentation method on the same image. Figure 9.7 shows the training samples used for the proposed method. Two reference images of the tex-

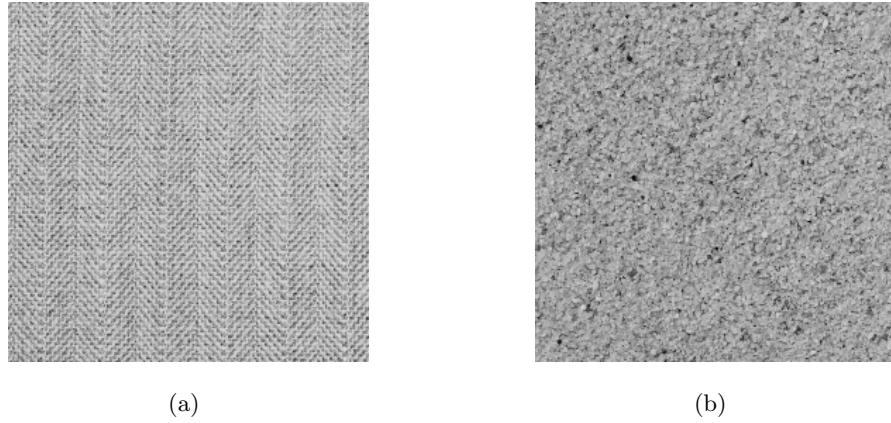


Figure 10.4: Training stage of the proposed method: (a) reference image for background, (b) reference image for the rectangle

ture patterns are given as training samples. Figure 10.4(a) shows the training samples for the background, and figure 10.4(b) shows the training samples for the rectangle. The proposed method estimates the PDF of both images as a mixture of Gaussian density functions, and assign them as the conditional PDF of image intensity for each subset. Figure 10.5 shows the the result of the proposed method. Figure 10.5(a) shows the final stage of contour evolution, and figure 10.5(b) shows the corresponding segmentation result. The proposed method successfully separates the rectangle from the background without any significant artifact.

For more precise comparison, the statistical distribution of image intensity within both the background and the rectangle are measured. Figure 10.6 shows the statistical distribution of image intensity within class 1, which is supposed to be the background in the segmentation result. Figure 10.6(a), 10.6(b), and 10.6(c) respectively present the results of method 1, method 2, and the proposed method. The blue solid lines of all three graphs present the statistical distribution of image intensity measured at the background, which is a mixture of at least three sub-classes. The green solid lines of all three graphs present the same statistics measured within class 1 in the segmentation results. As the green solid line exists closer to the blue solid line, it



Figure 10.5: Proposed method applied to a synthetic textured image: (a) the final stage of contour evolution, (b) the segmented subsets

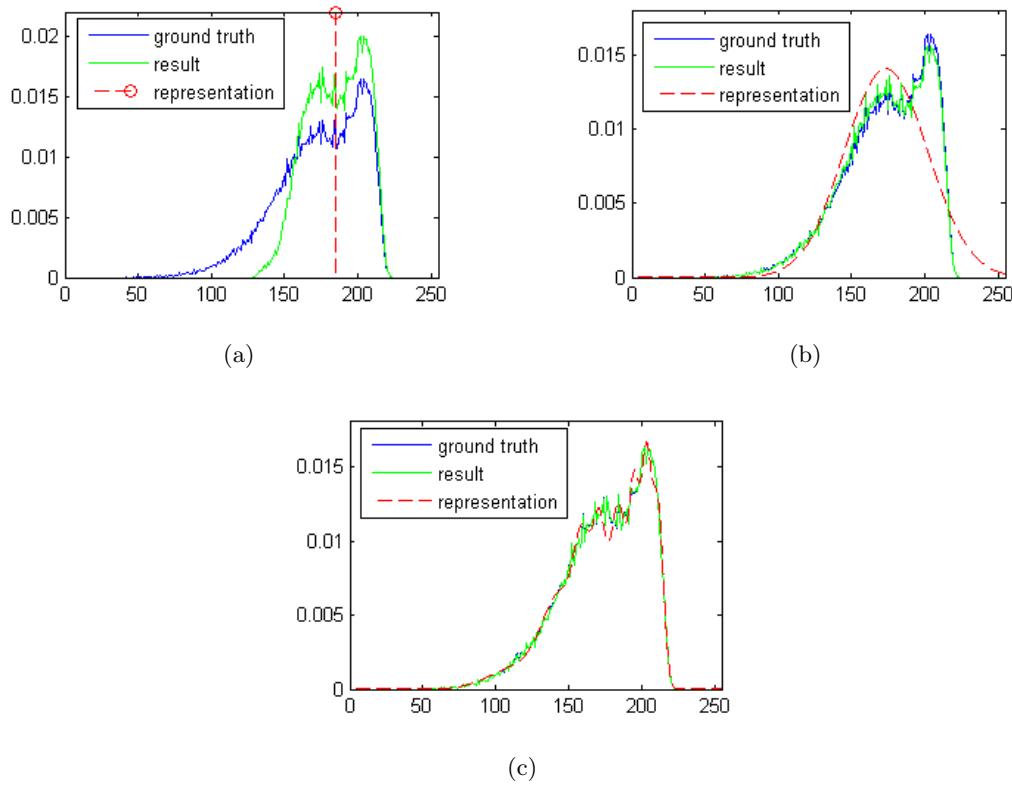


Figure 10.6: Statistical distribution of image intensity within background: (a) method 1, (b) method 2, (c) proposed method

presents better result. In figure 10.6(a), the dotted red line with a circle presents the represen-

tation of the subset used in method 1, i.e. the mean of green solid line. The measured statistics is very different from the ground truth because the constant representation is not sufficient to represent the non-uniform sub-region consisting of many sub-classes. In figure 10.6(b), the dotted red line presents the representation of the subset used in method 2, i.e. the Gaussian density function of image intensity measured. The measured statistics is similar to the ground truth though the representation is quite different from the true statistics. In figure 10.6(c), the dotted red line presents the estimated parametric mixture density function from training samples. The estimated density function is very similar to the true statistics, and the measured statistics is almost identical to the ground truth. Figure 9.10 shows the same measure for class 2, the rectangle. Although the ground truth looks like a simple Gaussian distribution, neither

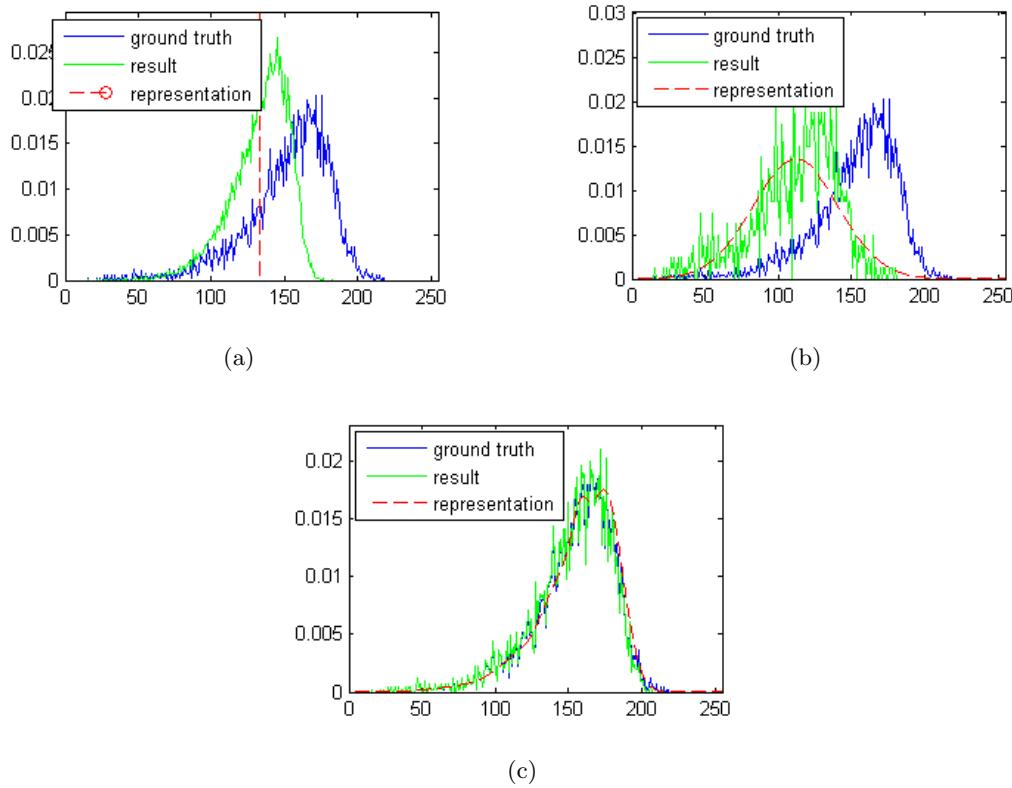
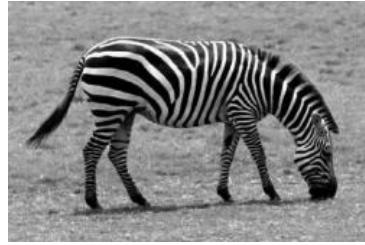


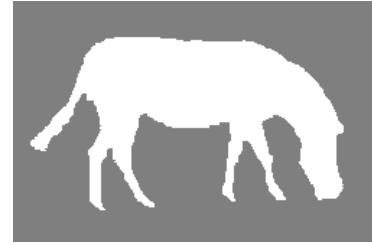
Figure 10.7: Statistical distribution of image intensity within the small rectangle: (a) method 1, (b) method 2, (c) proposed method

method 1 or 2 can produce reasonable results, as shown in figure 10.7(a) and 10.7(b), because the non-uniform statistics in the other subset was too complicated for them. In figure 10.7(c), the proposed method estimates very accurate statistics from training samples, and it leads to a very successful segmentation result.

Figure 10.8 shows an example of outdoor gray images with texture. There is a zebra stand-



(a)

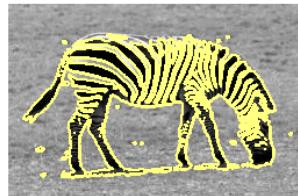


(b)

Figure 10.8: A complicated outdoor image: (a) a zebra, (b) the ground truth image

ing on the field. The zebra consists of white and black stripes, while the background is uniformly gray. The object of this experiment is to extract the shape of zebra from the background. This is a difficult segmentation problem because of the complicated patterns of the zebra and the shade under the zebra. The white stripes of zebra are brighter than the background, while the black stripes of zebra are darker than the background. Also, some portions of zebras, e.g. white stripes under the shade, have similar intensity to the background. The shade under the zebra is as dark as the black stripes though it is not part of the zebra. The statistical distribution of image intensity within the zebra must be represented as a mixture of more than at least two sub-classes.

Figure 10.9 shows the result of method 1 applied on figure 10.8. Figure 10.9(a) shows the



(a)



(b)

Figure 10.9: Method 1 applied to a complicated outdoor image: (a) the final stage of contour evolution, (b) the segmented subsets

final stage of contour evolution, and figure 10.9(b) shows the corresponding segmentation result. Method 1 separates the zebra into two parts, i.e. bright sub-regions and dark sub-regions, but identifies the bright sub-regions of the zebra as the background.

Figure 10.10 shows the result of method 2 applied on figure 10.8. Figure 10.10(a) shows



Figure 10.10: Method 2 applied to a complicated outdoor image: (a) the final stage of contour evolution, (b) the segmented subsets

the final stage of contour evolution, and figure 10.10(b) shows the corresponding segmentation result. Although method 2 produces better segmentation result than method 1, there are still many artifacts on both zebras and background. The sub-regions of zebras whose intensity is similar to the background are identified as background, i.e. the white subset, while the sub-regions of background whose intensity is darker than the other parts of background are identified as zebras, i.e. the black subset.

Figure 10.11 shows the training stage of the proposed segmentation method. We used the same image as the training samples for this experiment. The yellow rectangles indicate the regions used as training samples. The statistical distribution of image intensity within those sampled regions $\{p(I|I \in \mathcal{I}_i)\}$ is estimated as a mixture of Gaussian density functions, and assigned as the conditional PDF of image intensity as the background $p(I(x,y)|(x,y) \in \Omega_1)$ and the zebra $p(I(x,y)|(x,y) \in \Omega_2)$. Figure 10.12(a) shows the final stage of contour evolution, and figure 10.12(b) shows the corresponding segmentation result. Although there are still a few artifacts on the zebra, the number has been significantly reduced. There are no artifacts in the background. The artifacts found within the zebra appear because the image intensity of those sub-regions is identical to the background. This experiment shows how robust the proposed method using mixture density functions is compared to traditional methods using unimodal



Figure 10.11: Training stage of the proposed method: (a) training samples for background, (b) training samples for the zebra



Figure 10.12: Proposed method applied to a complicated outdoor image: (a) the final stage of contour evolution, (b) the segmented subsets

density functions, and it also shows the limitation of the image segmentation method based on image intensity as it cannot separate two objects with identical image intensities.

Figure 10.13 shows an example of applications using the proposed method on indoor RGB images. The object of this experiment is to simulate ATR problems with real RGB images instead of synthetic images. We measure training samples from a reference image, and apply the proposed method with the training samples into other images taken under different light condition and angle. We applied them into the images with even unknown objects. Figure 10.13(a) shows the reference image. There is a toy tank on the carpet, and the carpet has a wool pattern. This image simulates a miniature version of a tank on the grass field. Figure 10.13(b) and 10.13(c) show the two test images. The reference image figure 10.13(a) is taken from directly above the target simulating a view from satellite or high-altitude reconnaissance flight,

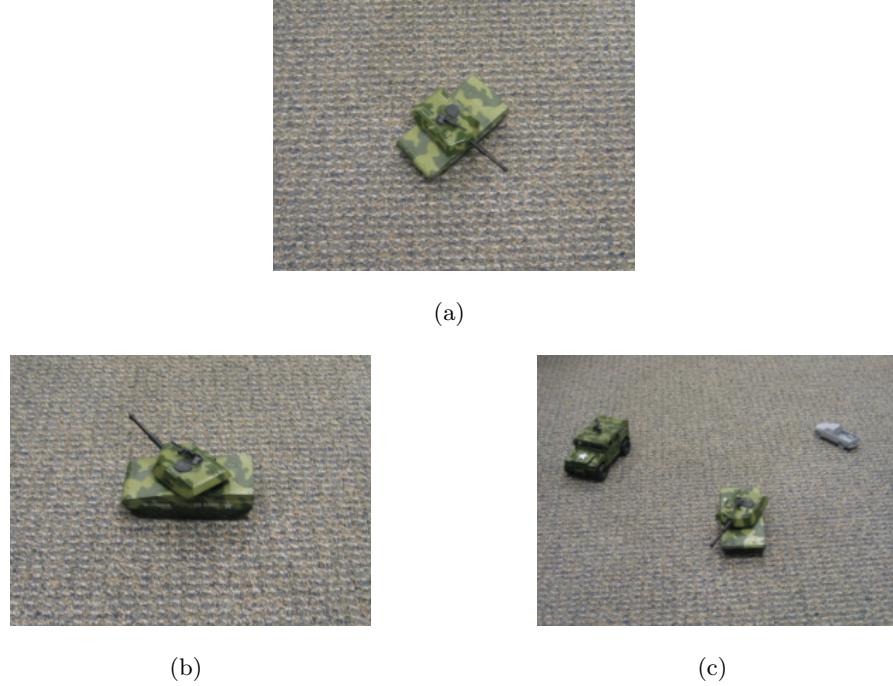


Figure 10.13: Complicated indoor RGB images: (a) reference image, (b) test 1, (c) test 2

while the first test image figure 10.13(b) is taken from a diagonal view and the focus is little bit blurred simulating a view from a missile or a jet fighter. The second test image figure 10.13(c) shows another target, a humvee, as well as the tank shown in the reference image. There is also a gray painted vehicle in the figure 10.13(c), which simulates a non-target object. This is the most difficult problem among the experiments shown in this document because the reference image and test images were taken from different views, and we do not have information for all objects appeared in the test images, i.e. the humvee and the gray painted vehicle.

Figure 10.11 shows the training stage of the proposed segmentation method. We obtain some training samples from the reference image. The yellow rectangles indicate the regions used as training samples. The statistical distribution of image intensity within those sampled regions $\{p(\mathbf{I}|\mathbf{I} \in \mathcal{I}_i)\}$ is estimated as a mixture of multivariate Gaussian density functions, and assigned as the conditional PDF of image intensity as the background $p(\mathbf{I}(x,y)|(x,y) \in \Omega_1)$ and the target $p(\mathbf{I}(x,y)|(x,y) \in \Omega_2)$.

Figure 10.15 shows the result of the proposed method applied on figure 10.13(b). Figure 10.15(a) shows the final stage of contour evolution, and figure 10.16(b) shows the corresponding seg-



Figure 10.14: Training stage of the proposed method: (a) training samples for background, (b) training samples for the tank



Figure 10.15: Proposed method applied to test 1 image: (a) the final stage of contour evolution, (b) the segmented subsets

mentation result. The object of this experiment is to extract the shape of the tank from the background, and the proposed method produces a reasonable result. Although there are a few small artifacts on the result, a smarter choice of training samples may solve the problem.

Figure 10.16 shows the result of the proposed method applied on figure 10.13(c). Figure 10.16(a) shows the final stage of contour evolution, and figure 10.16(b) shows the corresponding segmentation result. The object of this experiment is to extract the shape of the tank and the humvee, but to ignore the non-target object, i.e. the gray painted car. There was no such information on the non-target object in the reference image. The proposed method successfully extracts the shape of the tank and the humvee, while it ignores the non-target object. There is an artifact on the hood of the humvee because the white star did not appear in the reference image. The



Figure 10.16: Proposed method applied to test 2 image: (a) the final stage of contour evolution, (b) the segmented subsets

segmentation result is little bit affected by shading and different light condition. The shadows of the gray painted vehicle are recognized as targets, and a bright portion of the tank is recognized as the background. This experiment shows the potential use of the proposed method in real ATR problems and the possible problems that should be improved by future research.

Chapter 11

Conclusions

The proposed work has two main objects: robust image segmentation and the integration of image segmentation and pattern classification. Although image segmentation is a fundamental problem in image analysis, it has been difficult for traditional image segmentation methods to produce satisfying results on images with non-uniform sub-regions. We have developed robust image segmentation methods, which can produce good segmentation results on difficult image segmentation problems that traditional segmentation methods cannot produce reasonable results. In traditional image analysis, image segmentation performs a role of preprocessing for pattern classification providing the input data. We have integrated image segmentation and statistical pattern classification by using the statistical distribution of image intensity as the feature used in image segmentation. Thus, the proposed image segmentation methods produce the result of statistical pattern classification as well as image segmentation reducing the computation time and effort.

The proposed image segmentation methods use the framework of active contours. As active contours always provide continuous boundaries of sub-regions, they can produce more reasonable segmentation results than traditional segmentation methods, and consequently improve the final results of image analysis. The mathematical implementation of the proposed active contour models is accomplished using level set method. By presenting contours as a level of a topological function, we can merge multiple contours into one contour, or can split a contour into multiple contours, providing a good flexibility in the use of active contours.

Traditional image segmentation methods partition an image based on either discontinuity

between sub-regions or uniformity within a sub-region of a desired feature, e.g. image intensity, texture. Instead of a simple feature, the proposed image segmentation methods use the statistical distribution of image intensity within a sub-region, in a form of conditional probability density functions, as the feature to determine the segmentation criteria. Since the segmentation criteria are determined by the statistics of large samples rather than a single image intensity at a pixel, the proposed image segmentation methods show robust performance on images with noise. The proposed image segmentation methods partition an image in two ways: unsupervised or supervised. The unsupervised method partitions an image based on the uniformity of the feature within a sub-region. The supervised method partitions an image based on the similarity of the feature within a sub-region to the feature provided from corresponding training samples. The unique character of the proposed methods, which helps them to outperform traditional methods, is the use of a mixture density function to represent the statistical distribution of image intensity within a subset. As the segmentation criteria is determined by a mixture of sub-probability density functions instead of a probability density function, the proposed methods are able to recognize subsets with non-uniform (multimodal) image intensity, such as zebras. For multispectral images, we use a mixture of multivariate density functions instead of a product of marginal mixture density functions to represent the statistics of complicated patterns. These methods recognize extremely non-uniform vector-valued image intensities, such as military camouflage patterns, very well.

As the application of the proposed segmentation principle, three image segmentation methods are proposed. First, we represent the statistics of image intensities within each subset during the contour evolution in a form of multi-dimensional histograms. The use of a non-parametric discrete function provides an improved segmentation results compared to traditional segmentation methods based on particular stochastic model because the statistics of image intensity can be hardly represented as a simple stochastic model. Second, we represent the same statistics as a mixture of multivariate Gaussian density functions during the contour evolution, using a prior knowledge provided from training samples. The use of supervised mixture density functions provides significant improvements on segmentation and classification results because proposed method can recognize extremely non-uniform subsets. This method provides reasonable results even on the images that the previous method cannot segment well. Third, instead of computing the statistics of image intensity during contour evolution, we compute the statistics of corresponding training samples in advance before image segmentation, and assign the pre-computed

statistics as the conditional PDF of image intensities for each subset during image segmentation. This method provides reasonable results even on images that the two previous proposed methods cannot segment well.

In order to show the improved performance of the proposed methods, we have applied them to images that other segmentation methods cannot properly segment. As benchmarks, the segmentation results of proposed methods and the results of two more traditional active contour models, which share similar segmentation criteria, are compared. The proposed methods have shown very promising results outperforming the reference methods.

Although the proposed methods produce very robust and promising results, there are still a few aspects could be improved. Since the proposed methods represent the statistical distribution of image intensity in a complicated form, i.e. a mixture of multivariate density functions, the required computation is relatively larger than traditional methods increasing the convergence time. The fast advance of computer processor may solve this problem. Since supervised methods extract information from training samples, the performance may rely on the quality of training samples. A smart choice of training samples may reduce the effect of this problem.

Bibliography

- [1] R. Gonzales and R. Woods, *Digital Image Processing*. Prentice Hall, 2nd ed., 2001.
- [2] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama, “Theory of communication,” *ACM Transactions on Database Systems*, vol. 26, no. 2, pp. 179–213, 2001.
- [3] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes, active contour model,” *International Journal of Computer Vision*, pp. 321–331, 1988.
- [4] S. Osher and J. Sethian, “Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations,” *Journal of Computationl Physics*, pp. 12–49, 1988.
- [5] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 8, p. 769, 1986.
- [6] T. Chan, B. Sandberg, and L. Vese, “Active contours without edges for vector-valued images,” *Journal of Visual Communication and Image Representation*, pp. 130–141, 2000.
- [7] G. Sapiro and D. L. Ringach, “Anisotropic diffusion of multivalued images with applications to color filtering,” *IEEE Transactions on Image Processing*, vol. 5, pp. 1582–1586, 1996.
- [8] M. Dallwitz, “Introduction to computer images.” <http://biodiversity.uno.edu/delta/www/images.htm>, April 1998.
- [9] “Wikipedia, the free encyclopedia.” <http://www.wikipedia.org>, 2004.
- [10] T. Chan and L. Vese, “Active contours without edges,” *IEEE Transactions on Image Processing*, no. 2, pp. 266–277, 2001.

- [11] M. Rousson and R. Deriche, "A variational framework for active and adaptive segmentation of vector valued images," in *Proc. of IEEE Workshop on Motion and Video Computing*, 2003.
- [12] R. Gonzales and R. Woods, *Digital Image Processing*. Addison-Wesley Publishing, 1st ed., 1993.
- [13] B. Fisher, S. Perkins, A. Walker, and E. Wolfart, "Hypermedia image processing reference: Digital filters." <http://www.cee.hw.ac.uk/hipr/html/filtops.html>, 1994.
- [14] D. Marr and E. Hildreth, "Theory of edge detection," in *Proc. of Roy. Soc.*, (London), pp. 187–217, 1980.
- [15] B. Green, "Edge detection tutorial." <http://www.pages.drexel.edu/~weg22/edge.html>, 2002.
- [16] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.
- [17] B. Green, "Canny edge detection tutorial." http://www.pages.drexel.edu/~weg22/can_tut.html, 2002.
- [18] A. Jain, *Fundamentals of Digital Image Processing*. Prentice Hall Information and System Sciences Series, Prentice Hall, 1989.
- [19] C. Brice and C. Fennema, *Computer Method in Image Analysis*, ch. Scene Analysis using Regions. Los Angeles: IEEE Computer Society, 1977.
- [20] S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 9, pp. 884–200, 1996.
- [21] S. Zhu and A. Yuille, "Region competition and its analysis: a unified theory for image segmentation," Tech. Rep. 7, Robotics Lab, Havard University, 1995.
- [22] K. Fu, R. Gonzales, and C. Lee, *Robotics: Control, Sensing, Vision, and Inteligence*. New York: McGraw-Hill, 1987.
- [23] W. Snyder and H. Qi, *Machine Vision*. Cambridge University Press, 2004.

- [24] J. Silverman and D. Cooper, "Bayesian clustering for unsupervised estimation of surface and texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 373–387, May 1993.
- [25] M. LaValle and S. Hutchinson, "A bayesian segmentation methodology for parametric image models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 211–217, February 1995.
- [26] Y. Han, W. Snyder, and G. Bilbro, "Discontinuity-preserving vector smoothing of multivariate mr images using vector mean field annealing," *Journal of Mathematical Imaging and Vision*, vol. 9, no. 3, pp. 199–212, 1998.
- [27] G. Bilbro and W. Snyder, "Range image restoration using mean field annealing," in *Advances in Neural Information Processing Systems* (Touretzky, ed.), vol. 2, pp. 594–601, 1989.
- [28] D. Geman and S. Geman, "Stochastic relaxation, gibbs distributions and bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, November 1984.
- [29] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, 1990.
- [30] G. Gerig, O. Kubler, R. Kikinis, and F. Jolesz, "Nonlinear anisotropic filtering of mri data," *IEEE Transactions on Medical Imaging*, vol. 11, pp. 221–232, June 1992.
- [31] W. Snyder, Y. Han, G. Bilbro, R. Whitaker, and S. Pizer, "Image relaxation: Restoration and feature extraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 620–624, June 1995.
- [32] W. Snyder, A. Logenthiran, P. Santiago, K. Link, G. Bilbro, , and S. Rajala, "Segmentation of magnetic resonance images using mean field annealing," *Image and Vision Computing*, vol. 10, pp. 362–368, July 1992.
- [33] W. Snyder, Y. Han, G. Billbro, R. Whitakar, and S. Pizer, "Unified theory of edge-preserving smoothing," in *Proc. of International Conference on Artificial Neural Networks*, (U.K.), September 1992.

- [34] M. Symonds, W. Snyder, and P. Santago, "Segmentation of phase-coded mr images using the mean field annealing," in *Proc. of Symposium for Computer Assisted Radiology*, (Baltimore), pp. 398–404, 1992.
- [35] I. Abdelqader, S. Rajala, W. Snyder, and G. Bilbro, "Energy minimization approach to motion estimation using mean field annealing," *Signal Processing*, vol. 28, pp. 291–309, September 1992.
- [36] G. Bilbro, W. Snyder, S. Garnier, and G. Gault, "Mean field annealing, a formalism for constructing gnc-like algorithms," *IEEE Transactions on Neural Networks*, vol. 3, pp. 131–138, January 1992.
- [37] G. Bilbro, W. Snyder, and R. Mann, "Mean field approximation minimizes relative entropy," *Journal of the Optical Society of America (A)*, vol. 8, February 1991.
- [38] J. Zerubia and R. Chellappa, "Mean field approximation using compound gauss-malkov random field for edge detection and image restoration," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2193–2196, 1990.
- [39] G. Bilbro, R. Mann, T. Miller, W. Snyder, D. Vout, and M. White, "Optimization by mean field annealing," in *Advances in Neural Information Processing Systems* (Touretzky, ed.), vol. 1, pp. 91–98, 1988.
- [40] Y. Han and W. Snyder, "Adaptive edge-preserving smoothing via adaptive mean field annealing," in *Proc. of SPIE International Symposium on Optical Applied Science and Engineering*, vol. 1766, 1992.
- [41] G. Bilbro and W. Snyder, "Optimization of functions with many minima," *IEEE Transactions on System, Man, and Cybernetics*, vol. 21, pp. 840–849, July 1991.
- [42] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, May 1983.
- [43] G. Bilbro and W. Snyder, "Mean field annealing, an application to image noise removal," *Journal of Neural Network Computing*, Fall 1990.
- [44] Hiriyanaiyah, G. Bilbro, and W. Snyder, "Restoration of locally homogeneous images using mean field annealing," *Journal of the Optical Society of America (A)*, vol. 6, pp. 1901–1912, December 1989.

- [45] A. Chakraborty, S. Lawrence, and J. Duncan, “Deformable boundary finding influenced by region homogeneity,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 624–627, 1994.
- [46] A. Chakraborty, S. Lawrence, and J. Duncan, “an integrated approach to boundary finding in medical images,” in *Proc. of IEEE Workshop on Biomedical Image Analysis*, pp. 13–22, 1994.
- [47] J. Roerdink and A. Meijster, “The watershed transform: Definitions, algorithms, and parallel strategies,” *Fundamenta Informaticae*, vol. 41, pp. 187–228, 2000.
- [48] S. Beucher and C. Lantuejoul, “Use of watershed in contour detection,” in *Proc. of International Workshop in Image Processing, Real-time Edge and Motion Detection/Estimation*, (Rennes, France), September 1979.
- [49] G. Ederra, “Mathematical morphology techniques applied to anti-personnel mine detection,” 1999, Dept. of Electronics and Information Processing, Vrije Universiteit Brussel, Belgium, 1999.
- [50] S. Beucher, “The watershed transformation applied to image segmentation,” in *Proc. of Conference on Signal and Image Processing in Microscopy and Microanalysis*, (Cambridge, UK), pp. 299–314, September 1991.
- [51] F. Meyer and S. Beucher, “Morphology segmentation,” *Journal of Visual Communication and Image Representation*, vol. 1, no. 1, pp. 21–46, 1990.
- [52] S. Beucher, “The watershed transformation.” <http://cmm.ensmp.fr/~beucher/wtshed.html>.
- [53] H. Heijmans, *Morphological Image Operators*. Boston: Academic Press, 1994.
- [54] J. Rivest, P. Soille, and S. Beucher, “Morphological gradients,” *Journal of Electronic Imaging*, vol. 2, December 1993.
- [55] E. Dougherty, *Mathematical Morphology in Image Processing*. Optical engineering, New York: Marcel Dekker, 1993.
- [56] J. Haddon and J. Boyce, “Image segmentation by unifying region and boundary information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 929–948, 1990.

- [57] T. Pavlidis and Y. Liow, "Integrating region growing and edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 3, pp. 225–233, 1990.
- [58] H. Nguyen, M. Worring, and R. Boomgaard, "Watersnakes: Energy-driven watershed segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 330–342, March 2003.
- [59] D. Gabor, "Theory of communication," *Journal of IEE*, vol. 93, pp. 429–457, 1946.
- [60] F. Campbell and J. Robson, "Application of fourier analysis to the visibility of gratings," *Journal of Physiology*, pp. 551–566, 1968.
- [61] T. Weldon and W. Higgins, "Designing multiple gabor filters for multitexture image segmentation," *SPIE Optical Engineering*, vol. 38, pp. 1478–1489, September 1999.
- [62] D. Dunn, W. Higgins, and J. Wakeley, "Texture segmentation using 2-d gabor elementary functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 130–149, February 1994.
- [63] A. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 23, pp. 1167–1186, 1991.
- [64] A. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, 1990.
- [65] K. Hammouda and E. Jernigan, "Texture segmentation using gabor filters," tech. rep., University of Waterloo.
- [66] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Transactions on Image Processing*, vol. 4, pp. 1549–1560, November 1995.
- [67] S. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$," *Transactions of the American Mathematical Society*, vol. 315, pp. 69–87, 1989.
- [68] J. Aujol, G. Aubert, and L. Blanc-Feraud, "Supervised classification for textured images," Tech. Rep. 4640, INRIA, 2002.
- [69] D. Young, "Active contour models (snakes)." <http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision7.html>, March 1995.

- [70] N. Ray, S. Acton, T. Altes, E. de Lange, and J. Brookeman, “Merging parametric active contours within homogeneous image regions for MRI-based lung segmentation,” *IEEE Transactions on Medical Imaging*, no. 2, pp. 189–199, 2003.
- [71] J. Sethian, “Tracking interfaces with level sets,” *American Scientist*, vol. 85, no. 3, 1997.
- [72] J. Sethian, *Level Set Methods*. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 1st ed., 1996.
- [73] T. Chan and L. Vese, “Active contours without edges,” Tech. Rep. 9853, Computational Applied Math Group, UCLA, 1998.
- [74] L. Evans and R. Gariepy, *Measure Theory and Fine Properties of Functions*. Boca Raton, Florida: CRC Press, 1992.
- [75] H. Zhao, T. Chan, B. Merriman, and S. Osher, “A variational level set approach to multiphase motion,” *Journal of Computational Physics*, vol. 127, pp. 179–195, 1996.
- [76] L. Vese and T. Chan, “A multiphase level set framework for image segmentation using the mumford and shah model,” *International Journal of Computer Vision*, no. 3, pp. 271–293, 2002.
- [77] J. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2nd ed., 1999.
- [78] R. Malladi and J. Sethian, “Image processing via level set curvature flow,” in *Proc. of National Academy of Science*, pp. 7046–7050, 1995.
- [79] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, vol. 153 of *Applied mathematical sciences*. New York: Springer, 2003.
- [80] S. Osher and R. Fedkiw, “Level set methods: an overview and some recent results,” *Journal of Computationl Physics*, no. 2, pp. 463–527, 2001.
- [81] V. Caselles, F. Catte, T. Coll, and F. Dibos, “A geometric model for active contours,” *Numerische Mathematik*, p. 19, 1993.
- [82] R. Malladi, J. Sethian, and C. Vemuri, “Shape modeling with front propagation: a level set approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 158–175, 1995.

- [83] F. Cohen and D. Cooper, "Maximum likelihood unsupervised textured image segmentation," *Computer Vision, Graphics, and Image Process*, vol. 54, pp. 239–251, May 1992.
- [84] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, no. 1, pp. 61–79, 1997.
- [85] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in *Proc. of IEEE International Conference on Computer Vision*, p. 694, 1995.
- [86] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient flows and geometric active contour models," in *Proc. of IEEE International Conference on Computer Vision*, pp. 810–815, 1995.
- [87] A. Yezzi, S. Kichenassamy, and A. Kumar, "A geometric snake model for segmentation of medical imagery," *IEEE Transactions on Medical Imaging*, no. 2, pp. 199–209, 1997.
- [88] G. Sapiro, "Color snakes," *Computer Vision and Image Understanding*, no. 2, pp. 247–253, 1997.
- [89] G. Sapiro, "Color snakes," Tech. Rep. 113, Hewlett-Packard Labs, 1995.
- [90] G. Aubert and L. Blanc-Feraud, "Some remarks on the equivalence between 2d and 3d classical snakes and geodesic active contours," *International Journal of Computer Vision*, no. 1, pp. 19–28, 1999.
- [91] G. Aubert and L. Blanc-Feraud, "An elementary proof of the equivalence between 2d and 3d classical snakes and geodesic active contours," Tech. Rep. 3340, INRIA, 1998.
- [92] C. Xu, A. Yezzi, and J. Prince, "A summary of geometric level set analogues for a general class of parametric active contour and surface models," in *Proc. of IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pp. 104–111, 2001.
- [93] C. Xu, A. Yezzi, and J. Prince, "on the relationship between parametric and geometric active contours," in *Proc. of IEEE Conference on Signals, Systems and Computers*, pp. 483–489, 2000.
- [94] G. Sapiro, "Vector (self) snakes: a geometric framework for color, texture, and multiscale image segmentation," in *Proc. of IEEE International Conference on Image Processing*, pp. 817–820, 1996.

- [95] G. Sapiro, “Vector-valued active contours,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 680–685, 1996.
- [96] E. Kreyszig, *Differential Geometry*. University of Toronto Press, 1959.
- [97] G. Sapiro, “from active contours to anisotropic diffusion: Connections between basic pde’s in image processing,” in *Proc. of IEEE International Conference on Image Processing*, pp. 477–480, 1996.
- [98] N. Paragios, O. Mellina-Gottardo, and V. Ramesh, “Gradient vector flow fast geometric active contours,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 3, pp. 402–407, 2004.
- [99] N. Paragios, O. Mellina-Gottardo, and V. Ramesh, “Gradient vector flow fast geodesic active contours,” in *Proc. of IEEE International Conference on Computer Vision*, 2001.
- [100] C. Xu and J. Prince, *Handbook of Medical Imaging: Processing and Analysis Management*, ch. Gradient Vector Flow Deformable Models. Academic Press, 2000.
- [101] C. Xu, D. Pham, and J. Prince, *Handbook of Medical Imaging: Processing and Analysis Management*, ch. Image Segmentation Using Deformable Models, pp. 129–174. Academic Press, 2000.
- [102] C. Xu and J. Prince, “Generalized gradient vector flow external forces for active contours,” *Signal Processing*, no. 3, pp. 131–139, 1998.
- [103] C. Xu and J. Prince, “Snakes, shapes, and gradient vector flow,” *IEEE Transactions on Image Processing*, no. 3, pp. 359–369, 1998.
- [104] C. Xu and J. Prince, “Gradient vector flow: a new external force for snakes,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 66–71, 1997.
- [105] T. Gevers, S. Ghebreab, and A. Smeulders, “Color invariant snakes,” in *Proc. of British Machine Vision Conference*, pp. 578–588, 1998.
- [106] G. Hamarneh, A. Chodorowski, and T. Gustavsson, “Active contour models: Application to oral lesion detection in color images,” in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 578–583, 2000.

- [107] D. Mumford and J. Shah, “Optimal approximation by piecewise smooth functions and associated variational problems,” *Communication Pure and Applied Mathematics*, p. 577, 1989.
- [108] D. Mumford and J. Shah, “Boundary detection by minimizing functionals,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1985.
- [109] B. Lee, B. Hamza, and H. Krim, “an active contour model for image segmentation: a variational perspective,” in *Proc. of IEEE International Conference on Acoustics Speech and Signal Processing*, 2002.
- [110] L. Vese, *Geometric Level Set Methods in Imaging, Vision, and Graphics*, ch. Multiphase Object Detection and Image Segmentation, pp. 175–194. New York: Springer-Verlog, 2003.
- [111] L. Vese, “Multiphase object detection and image segmentation,” Tech. Rep. 0236, Computational Applied Math Group, UCLA, 2002.
- [112] A. Tsai, A. Yezzi, and A. Willsky, “Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification,” *IEEE Transactions on Image Processing*, no. 8, pp. 1169–1186, 2001.
- [113] A. Tsai, A. Yezzi, and W. Wells, “A curve evolution approach to smoothing and segmentation using the mumford-shah functional,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2000.
- [114] T. Chan and L. Vese, “Image segmentation using level sets and the piecewise-constant mumford-shah model,” Tech. Rep. 0014, Computational Applied Math Group, UCLA, 2000.
- [115] T. Chan and L. Vese, “A level set algorithm for minimizing the mumford-shah functional in image processing,” in *Proc. of IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pp. 161–168, 2001.
- [116] L. Vese and T. Chan, “A multiphase level set framework for image segmentation using the mumford and shah model,” Tech. Rep. 0125, Computational Applied Math Group, UCLA, 2001.

- [117] T. Chan and L. Vese, “A level set algorithm for minimizing the mumford-shah functional in image processing,” Tech. Rep. 0013, Computational Applied Math Group, UCLA, 2000.
- [118] M. Rousson and R. Deriche, *Geometric Level Set Methods in Imaging, Vision, and Graphics*, ch. Adaptive Segmentation of Vector Valued Images, pp. 195–206. New York: Springer-Verlog, 2003.
- [119] M. Rousson and R. Deriche, “A variational framework for active and adaptive segmentation of vector valued images,” Tech. Rep. 4515, INRIA, 2002.
- [120] A. Yezzi, A. Tsai, and A. Willsky, “A fully global approach to image segmentation via coupled curve evolution equations,” *Journal of Visual Communication and Image Representation*, pp. 195–216, 2002.
- [121] A. Yezzi, A. Tsai, and A. Willsky, “Medical image segmentation via coupled curve evolution equations with global constraints,” in *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 104–111, 2000.
- [122] C. Samson, L. Blanc-Feraud, and G. Aubert, “A level set model for image classification,” *International Journal of Computer Vision*, no. 3, pp. 187–197, 2000.
- [123] C. Samson, L. Blanc-Feraud, and G. Aubert, “Multiphase evolution and variational image classification,” Tech. Rep. 3662, INRIA, 1999.
- [124] C. Samson, L. Blanc-Feraud, and G. Aubert, “Image classification using a variational approach,” Tech. Rep. 3523, INRIA, 1998.
- [125] T. Chan, B. Sandberg, and L. Vese, “Active contours without edges for vector-valued images,” Tech. Rep. 9935, Computational Applied Math Group, UCLA, 1999.
- [126] N. Paragios and R. Deriche, “Geodesic active regions: A new framework to deal with frame partition problems in computer vision,” *Journal of Visual Communication and Image Representation*, pp. 249–268, 2002.
- [127] N. Paragios, *Geodesic Active Regions and Level Set Methods: Contributions and Applications in Artificial Vision*. PhD thesis, Dept. of Electrical and Computer Engineering, INRIA, 2000.

- [128] N. Paragios and R. Deriche, "Unifying boundary and region-based information for geodesic active tracking," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 300–305, 1999.
- [129] N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation: a level set approach," in *Proc. of European Conference on Computer Vision*, 2000.
- [130] N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation," Tech. Rep. 3783, INRIA, 1999.
- [131] N. Paragios, "A variational approach for the segmentation of the left ventricle in cardiac image analysis," *International Journal of Computer Vision*, no. 3, pp. 345–362, 2002.
- [132] N. Paragios, "A variational approach for the segmentaion of the left ventricle in mr cardiac images," in *Proc. of IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, 2001.
- [133] S. Jehan-Besson, M. Barlaud, and G. Aubert, "Shape gradients for histogram segmen-tation using active contours," in *Proc. of IEEE International Conference on Computer Vision*, 2003.
- [134] S. Jehan-Besson, M. Barlaud, and G. Aubert, "Region-based active contours using geometrical and statistical features for image segmentation," in *Proc. of IEEE International Conference on Image Processing*, pp. 643–646, 2003.
- [135] A. Chakraborty, S. Lawrence, and J. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Transactions on Medical Imaging*, no. 6, pp. 859–870, 1996.
- [136] S. Osher and N. Paragios, eds., *Geometric Level Set Methods in Imaging, Vision, and Graphics*. New York: Springer-Verlog, 2003.
- [137] A. Blake and M. Isard, *Active Contours*. Springer-Verlag, 1998.
- [138] J. Morel and S. Solimini, *Variational Methods in Image Segmentation*, vol. 14 of *Progress in nonlinear differential equations and their applications*. Boston: Birkhauser, 1995.
- [139] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, no. 3, pp. 223–247, 2002.

- [140] N. Paragios and R. Deriche, "Geodesic active contours for supervised texture segmentation," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1999.
- [141] C. P. Lee and W. E. Snyder, "Multispectral image segmentation using active contours," in *Proc. of SPIE Defense and Security Symposium*, (Orlando, Florida), April 2004.
- [142] B. Sandberg, *Active Contour Segmentation of Multi-channel Images*. PhD thesis, Dept. of Mathematics, UCLA, 2002.
- [143] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. John Wiley & Sons, Inc., 2nd ed., 2000.
- [144] C. Archambeau and M. Verleysen, "Fully nonparametric probability density function estimation with finite gaussian mixture models," in *Proc. of International Conference on Advances in Pattern Recognition*, (Calcutta, India), pp. 81–84, December 2003.
- [145] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press, 1998.
- [146] M. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 381–396, March 2000.
- [147] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite mixture Distributions*. U.K.: John Wiely & Sons, 1985.
- [148] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. B, pp. 1–38, 1977.
- [149] R. Render and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM review*, vol. 26, pp. 195–239, April 1984.
- [150] G. McLachlan and D. Peel, *Finite Mixture Models*. New York: John Wiley, 2000.
- [151] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics, New York: John Wiley, 1997.
- [152] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," *Nueral Networks*, vol. 11, pp. 271–282, 1998.
- [153] T. Hastie and R. Tibshirani, "Discriminant analysis by gaussian mixtures," *Journal of Royal Statistical Society (B)*, vol. 58, pp. 155–176, 1996.

- [154] S. Roberts, D. Husmeier, I. Rezek, and W. Penny, “Bayesian approaches to gaussian mixture modelling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1133–1142, November 1998.
- [155] J. Hoffbeck and D. Landgrebe, “Estimating the number of components in a normal mixture density function,” in *Proc. of IEEE International Geoscience and Remote Sensing Symposium*, pp. 1674–1677, 2000.
- [156] J. Oliver, R. Baxter, and C. Wallace, “Unsupervised learning using MML,” in *Proc. of International Conference on Machine Learning*, pp. 364–372, 1996.
- [157] C. Wallace and D. Dowe, “Estimation and inference via compacy coding,” *Journal of Royal Statistical Society (B)*, vol. 49, no. 3, pp. 241–252, 1987.
- [158] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*. singapore: World Scientific, 1989.
- [159] C. Wallace and D. Dowe, “Minimum message length and kolmogorov complexity,” *The Computer Journal*, vol. 42, no. 4, pp. 270–283, 1999.
- [160] J. Bernado and A. Smith, *Bayseian Theory*. U.K.: John Wiley & Sons, 1994.
- [161] T. Cover and J. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, 1991.
- [162] G. Celeux, S. Chretien, F. Forbes, and A. Mkhadri, “A component-wise EM algorithm for mixtures,” Tech. Rep. 3746, INRIA, 1999.
- [163] A. McLeod, “Nonparametric probability density estimation.” Course Note, March 2004. Dept. of Statistical and Actuarial Sciences, University of Western Ontario.
- [164] R. Bellman, *Adaptive Control Processes: a Guided Tour*. Princeton University Press, 1961.
- [165] D. Donoho, “High-dimensional data analysis: The curse and blessings of dimensionality.” Lecture Note, August 2000. Dept. of Statistics, Stanford University.