SHAPE AND DATA DRIVEN TEXTURE SEGMENTATION USING LOCAL BINARY PATTERNS

by

ERKİN TEKELİ

Submitted to the Graduate School of Engineering and Natural Sciences

in partial fulfillment of

the requirements for the degree of

Master of Science

Sabancı University

February 2007

SHAPE AND DATA DRIVEN TEXTURE SEGMENTATION USING LOCAL BINARY PATTERNS

Assistant Prof. Müjdat Çetin ……..
(Thesis Supervisor)

Prof. Aytül ERÇİL ……..
(Thesis Co-Advisor)

Associate Prof. Mustafa Ünel ……..

Associate Prof. Berrin Yanıkoğlu ……..

Assistant Prof. Hakan ERDOĞAN ……..

DATE OF APPROVAL: ……

SHAPE AND DATA DRIVEN TEXTURE SEGMENTATION USING LOCAL BINARY PATTERNS

Erkin TEKELİ

Electronics Engineering and Computer Sciences, MSc. Thesis, 2007

Thesis Supervisor: Assistant Prof. Müjdat Çetin
Thesis Co-Advisor: Prof. Aytül ERÇİL

Keywords: Shape priors, texture segmentation, Local Binary Pattern (Partition) (LBP), active contours, nonparametric density estimation

ABSTRACT

Image segmentation is a fundamental step in image analysis. Segmentation can be done by isolating homogeneous regions within an image or finding the boundaries between such regions. There are several cases that intensity values, color, mean or variance of image intensity distributions and edge information cannot play a discriminative role in image segmentation. While other features are not sufficient to discriminate regions, texture might be a good feature to handle the segmentation problem. Texture analysis and texture segmentation are still challenging problems; there is no method which can clearly identify and discriminate all kind of textures. Especially identifying nonuniform textures and discriminating from other textures are still difficult problems in texture analysis. For this reason texture segmentation approaches may give unsatisfactory segmentation results with

missing data or with corrupted boundaries of regions. Using prior information about the shape of the object can aid segmentation which can also obtain a solution to occlusion problems.

In this thesis, we propose a shape and data driven texture segmentation method using local binary pattern (LBP). In particular, we train our LBP based texture filter with the texture which belongs to the region that we want to segment. We input the textured image into our filter to produce a "filtered image" which has been eluded from the structural properties of texture. Then by an energy functional, which combines the data term produced from the filtered image and shape prior term under a Bayesian framework, we evolve our level set based active contour for segmentation.

YEREL İKİLİ ÖRÜNTÜ(LBP) KULLANARAK ŞEKİL VE VERİYE DAYALI DESEN BÖLÜTLEME

Erkin TEKELİ

Elektronik Mühendisliği ve Bilgisayar Bilimleri, Yüksek Lisans Tezi, 2007

Tez Danışmanı: Yard. Doç. Dr. Müjdat Çetin
Yardımcı Tez Danışmanı: Prof. Dr. Aytül ERÇİL,

Keywords: Önsel şekil bilgisi, desen bölütleme, Yerel ikili örüntü (LBP), aktif konturlar, parametrik olmayan yoğunluk kestirimi

ÖZET

İmge bölütleme, imge analizinde temel bir adımdır. İmge bölütlemenin amacı basit olarak imgeyi anlamlı (homojen) bölümlere ayırmak olarak tanımlanabilir. Bölütleme imgedeki homojen bölgelerin izole edilmesi veya bu bölgeler arasındaki sınırların bulunması ile yapılabilir. İmge bölütleme problemlerinde piksel değerlerinin, rengin, piksel değerlerinin istatistiksel dağılımlarının ortalaması veya değişirliğinin ve kenar bilgisinin ayırt edici rol oynayamadığı çeşitli durumlar vardır. Diğer özniteliklerin ayırt edici rol oynayamadığı durumlarda desen iyi bir ayırt edici öznitelik olabilir. Fakat desen analizi ve desen bölütleme hala çözülmesi zor olan problemlerdir. Henüz bütün desen türlerini birbirinden ayırt edebilen ve tanımlayabilen bir yöntem bulunamamıştır. Özellikle düzensiz yapılı desenlerin tanımlanması ve ayırt edilmesi hala çözülmesi zor desen analizi problemlerindendir. Bundan dolayı desen bölütleme yaklaşımları tatmin edici bölütleme sonuçları vermeyebilir. Örneğin hatalı desen tanımlanması sonucunda objelerin sınırlarında

bozulmalar veya kayıplar gözlenebilir. Fakat bölütleyeceğimiz desen yapılı bölgenin şekil bilgisi hakkından bir ön bilgimiz olması bize daha iyi bölütleme sonuçlarına ulaşmamızda yardımcı olabilir. Aynı zamanda şekil bilgisi kapatılma problemlerinde bölütlemede bize çözüm sağlayabilir.

Bu tezde, yerel ikili örüntü (LBP) kullanarak şekil ve veriye dayalı desen bölütleme metodu sunuyoruz. Kısaca, LBP tabanlı desen filtremizi bölütlemek istediğimiz bölgeye ait desen ile eğittikten sonra test imgemizi desensel yapıdan kurtarmak için desen filtremiz ile filtreliyoruz. Daha sonra bölütleme için veri ve önsel şekil bilgisi terimlerini Bayes metodu bünyesinde birleştiren bir enerji denklemi kullanarak aktif konturumuzu hareket ettiriyoruz.

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge my advisor Assistant Prof. Müjdat Çetin for his valuable guidance, encouragement, unconditional support and understanding throughout my studies. I am also very grateful to my co-advisor Prof. Aytül ERÇİL for her support, trust and the opportunity she has given to me to pursue graduate studies at Sabancı University.

I am also grateful to Associate Prof. Mustafa Ünel, Associate Prof. Berrin Yanıkoğlu, and Assistant Prof. Hakan Erdoğan for their participation in my thesis committee. It was a great pleasure to feel their kindly support at my presentation.

I sincerely thank my colleagues and friends in VPA Laboratory for their cooperation and many fruitful interactions. I also would like to thank all my friends for their support during all my studies.

I am grateful deeply to my family for their endless love, support and patience that made everything about me possible. Without their belief and trust nothing would be possible in my life. I dedicate this thesis to my dear family.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Image segmentation has an important role in image processing. It is a fundamental step which can be defined as isolating homogeneous regions within an image or finding the boundaries between such regions. Image segmentation has been approached from a wide variety of perspectives but still it is a challenging problem and becomes more complicated in case of textured images. There are several thoughts about the definition of texture. Even though texture is an intuitive concept, a formal definition of texture has proven elusive in the literature over the years. Maria Petrou and Pedro Garcia Sevilla[21] defined texture as the variation of data at scales smaller than the scales of interest. There are many application areas of image processing in which texture plays important roles. The most important application areas are biomedical image analysis, industrial inspection, analysis of satellite imagery, content-based retrieval from image databases, document analysis, biometric person authentication, scene analysis for robot navigation, texture synthesis for computer graphics and animation, and image coding. In several cases of image segmentation problems; intensity values, color, mean or variance of image intensity distributions and edge information cannot play a discriminative role. While other features are not sufficient to discriminate regions, texture might be a good feature to handle the segmentation problem. We can define texture segmentation as the partitioning of an image into regions, each of which contains a single texture distinct from its neighbors.

Although texture analysis has been a topic of intensive research for over three decades, texture analysis and texture segmentation are still challenging problems. There is no method which can clearly discriminate all kind of textures. Especially defining nonuniform textures' structures and discriminating from other textures are still difficult problems in texture analysis. For this reason texture segmentation approaches may give unsatisfactory segmentation results with missing data or with corrupted boundaries of

regions because of the inaccurate texture identification results of texture analysis methods. Using prior information about the shape of the object can aid segmentation which can also obtain a solution to occlusion problems.

In this thesis, our aim is to use a robust texture filtering approach which can distinguish different textured regions in an image with similar first order pixel intensity probability density functions (pdfs) which cannot be distinguished by pixel intensity based methods. By combining filtered data with shape prior information we aim to reach satisfactory segmentation results. According to this purpose we propose a shape and data driven texture segmentation method using local binary pattern. In particular, we train our LBP based texture filter with the texture which belongs to the region that we want to segment. Then we enter the textured image into our filter to produce a "filtered image" which means the image has been eluded from texture structural properties. In other words we aim to produce a filtered image where the regions can be discriminated according to structurally independent pixel values. Then we produce a data term for our energy equation which involves maximizing mutual information between the region labels and the image pixel intensities. Also we produce a nonparametric shape prior by estimating underlying shape distribution by using Parzen density estimator in space of shapes. Then we combine the data term and shape prior within a Bayesian framework to form the energy functional which we minimize using active contours for segmentation.

1.1) Contributions

Texture Filter: In this thesis our first contribution is our texture filter. We propose a Local Binary Pattern (LBP) based texture filter to elude textured images from textural structure. We apply a windowing process to textured images in LBP domain to produce an image with distinct regions according to structurally independent pixel intensity values.

Using Shape Priors in Texture Segmentation: Our second contribution in this thesis is using nonparametric density estimation based shape prior information in texture segmentation problems.

1.2) Organization:

Chapter 2:

In this chapter we provide a theoretical background which will be necessary to follow the work done in this thesis. It begins with the background material on image segmentation and texture analysis. Respectively we present the methods used in the literature and emphasize the tools used in this thesis. Then we provide background on texture segmentation and discuss how image segmentation and texture analysis tools are used in solving the segmentation problem. Then we review previous work on shape analysis and shape-based segmentation, focusing on the tools used in this thesis. Finally we present some background on nonparametric density estimation and entropy estimation.

Chapter 3:

This chapter presents our shape and data driven texture segmentation method using local binary pattern. It begins with the detailed explanation of our texture filtering approach. Then we discuss the data term we use in our energy equation. Then we discuss the shape priors and how we combine our shape and data term under the energy equation we use. Finally we provide our experimental results.

Chapter 4:

We conclude by summarizing the contributions of this thesis and we discuss the future work to extend our approach.

CHAPTER 2

BACKGROUND

In this chapter we provide a theoretical background which will be necessary to follow the work done in this thesis. In Section 2.1 and Section 2.2 we provide background material on image segmentation and texture analysis. Respectively we present the methods used in the literature and emphasize the tools used in this thesis. In Section 2.3 we provide background on texture segmentation and discuss how image segmentation and texture analysis tools are used in solving the segmentation problem. In Section 2.4 we review previous work on shape analysis and shape-based segmentation, focusing on the tools used in this thesis. In Section 2.5 we present some background on nonparametric density estimation and entropy estimation.

2.1) Image segmentation:

Image segmentation is a fundamental step in image analysis. The aim of image segmentation can be simply defined as extracting the region we are interested in from an image. Segmentation can be done by isolating homogeneous regions of different objects within an image or finding the boundaries between such regions. Segmentation stage is not concerned with the identity of the objects that can be labeled later.

Choosing a suitable approach for isolating different objects from the background is the main problem of image segmentation. When the gray levels of different objects are quite similar in an image, the segmentation problem is challenging. For this reason the aim of image segmentation techniques is to distinguish the region of interest from the whole image by utilizing the salient features.

Image segmentation has been approached from a wide variety of perspectives. Some of the gray level image segmentation approaches are histogram thresholding; tree/graph based approaches, region growing, clustering, probabilistic or Bayesian approaches, neural networks, active contour or curve evolution methods. In this thesis we have used a region based active contour approach where the level set method has been used to implement. Active contours are evolved by minimizing an energy functional $E(C)$. In this section, first we provide an overview of active contour models and then provide some mathematical tools used for curve evolution equations for the energy functionals.

2.1.1) Active Contours:

In recent years there has been great interest in active contours. The earlier development of active contour models is concerned with "boundary-based" models and more recent work about active contours is concerned with "region-based" models. In this section we provide some key pieces of active contours that are most relevant for this thesis.

The main idea of active contours is to evolve a curve such that it converges to the boundary of an object by minimizing an energy functional $E(C)$. In order to minimize the energy functionals $E(C)$, a variational analysis is used to compute the velocity field $\partial C/\partial t = V(C)$, by which the curve is evolved iteratively over time t.

The velocity field $V(C)$ is often the best deformation δC in that it maximizes:

$$\lim_{h \to 0} - \frac{E(\vec{C} + h\delta\vec{C}) - E(\vec{C})}{h} \qquad (2.1)$$

In this sense, such a velocity field is called a gradient flow. Difference of boundary-based models and region based models is; in boundary based models, the data driven part of the energy functional depends solely on the information in the local neighborhood of the active contour. In region-based models the energy functional depends on the information obtained from the region inside or outside the curve, or from the entire image. In this thesis we use region-based active contours. For this reason, we focus on region-based active contours from this point on, and refer the reader to [11] for details of boundary based approaches.

- Region Based Active Contour Approaches

Region based methods are recent approaches to active contours. In particular, a number of methods, which are inspired by the Mumford-Shah functional [15, 16] will be overviewed in this part. The original Mumford-Shah functional is given in terms of a general set of discontinuities (boundary) Γ as follows:

$$E(f,\Gamma) = \beta\int_{\Omega}(f-g)^2 dx + \alpha\int_{\Omega-\Gamma}\left|\nabla f\right|^2 dx + \gamma\left|\Gamma\right| \qquad (2.2)$$

|Γ| stands for the total length of the arcs making up Γ. Here g is the image intensity observed, f is a piecewise smooth approximation of the observed image and Ω is the image domain. The first term stands for the estimate f being as close as possible to the data g and the second term is a smoothness criterion in that the estimate f should be smooth at all points except those on the boundary. The third term says that the boundary should be as short as possible. Tsai et al. [31] and Vese et al. [35] used Mumford-Shah functional for a given curve C.

$$E(f,\underset{\rho}{C}) = \beta\int_{\Omega}(f-g)^2 dx + \alpha\int_{\Omega-\underset{\rho}{C}}\left|\nabla f\right|^2 dx + \gamma\oint_{\underset{\rho}{C}} ds \qquad (2.3)$$

They also developed a curve evolution solution for minimizing this functional.

Chan and Vese proposed an approach [3] called "Active contours without edges". They solve the following modified Mumford Shah functional,

$$E(\underset{\rho}{C}) = \int_{R}(g(x)-c_1)^2 dx + \int_{R_C}(g(x)-c_2)^2 dx + \gamma\oint_{\underset{\rho}{C}} ds \qquad (2.4)$$

where R is the region inside the curve, R_c is the region outside the curve, c_1 is the average intensity value inside the curve C, and c_2 is the average intensity value outside the curve C in the image g. So, c_1 and c_2 constants depend on the curve C. Restricting the Mumford-Shah functional by approximating the observed image g by a piecewise constant image ($f(x) = c_1$ if $x \in R$ and $f(x) = c_2$ if $x \in R^c$) makes this functional a reduced form of the Mumford-Shah functional.

Yezzi et al. [38] proposed another region-based approach, whose energy functional is given by:

$$E(\vec{C}) = -\frac{1}{2}(u-v)^2 + \alpha \oint_{\vec{C}} ds \tag{2.5}$$

where u and v are respectively the mean intensities of the region inside and outside the curve. This energy functional takes the mean intensity of each region as the statistical feature. The curve evolution based on this energy functional basically tries to maximally separate the mean intensities of the region inside the curve and region outside the curve. Also they proposed an energy functional which is given in terms of difference of the intensity variances:

$$E(\vec{C}) = -\frac{1}{2}(\sigma_u^2 - \sigma_v^2)^2 + \alpha \oint_{\vec{C}} ds \tag{2.6}$$

σ_u^2 and σ_v^2 are respectively the sample variance inside and outside the curve *C*. By using the intensity variance as a discriminative feature they can segment a different class of images on which the original Mumford Shah functional does not work.

Recently Kim et al. [11,12] proposed a nonparametric statistical method for image segmentation. Their mutual information based energy functional is as follows:

$$E(\vec{C}) = -|\Omega| \hat{I}(G(X); L_{\vec{C}}(X)) + \alpha \oint_{\vec{C}} ds \tag{2.7}$$

They approach the image segmentation problem as maximization of mutual information between region labels and image pixel intensities. In other energy functionals we have mentioned before they use statistical features of the pixel intensity distributions (e.g. mean or variance). Kim et al.'s energy functional provides a solution to the segmentation problems with regions which have same first order pdfs. Whether the homogeneous regions have same means or variances in the image, mutual information aid the segmentation problem. In equation 2.7 $\hat{I}(.)$ is the estimated mutual information between intensity values ($G(X)$) and binary label ($L_{\vec{C}}(X)$). $|\Omega|$ is the area of the image domain and $\oint_{\vec{C}} ds$ is the length of the curve $\vec{C}$ which is a regularization term used as curve length penalty. α is a scalar constant and *X* is the pixel index. Suppose that we have an evolving curve $\vec{C}$ and let us define the region inside curve as R_+ and region

outside the curve as R_-.We can define $L_{\vec{C}}(.)$ as a binary label which is a mapping from the image domain Ω to a set of two labeling symbols which are L_+ and L_-. So, if the pixel X is in R_+, then $L_{\vec{C}}(X)=L_+$ else if the pixel X is in R_-, then $L_{\vec{C}}(X)=L_-$. As a result of this we can say that $L_{\vec{C}}(X)$ is a random variable which depends on the curve $\vec{C}$. Suppose that we have two regions in our image. R_1 is the region of our object and R_2 is the background. So, the probability of the pixel intensity value of a randomly chosen pixel X is as follows:

$$p_{G(X)}(z)=\frac{|R_1|}{|\Omega|}p_1(z)+\frac{|R_2|}{|\Omega|}p_2(z) \tag{2.8}$$

Which means that $G(X)$ is a random variable that depends on the true regions R_1 and R_2. According to these properties of $G(X)$ and $L_{\vec{C}}(X)$ they write the dependencies of these two random variables by using mutual information. The more accurate the label $L_{\vec{C}}(X)$, the less uncertainty has $G(X)$. For instance when the contour fits exactly to the boundary of region R_1, the uncertainty of G(X) will be zero.

Since the distributions of the regions are unknown they use $\hat{I}(.)$ which is the nonparametric Parzen density estimate of mutual information. To estimate mutual information it is necessary to estimate the differential entropies of following terms:

$$\hat{h}(G(X)\mid L_{\vec{C}}(X)=L_+) \tag{2.9}$$

$$\hat{h}(G(X)\mid L_{\vec{C}}(X)=L_-) \tag{2.10}$$

The estimate of equation 2.9 and 2.10 using continuous version of Parzen density estimation are as follows:

$$-\frac{1}{|R_+|}\int_{R_+}\log\left(\frac{1}{|R_+|}\int_{R_+}K(G(x)-G(\hat{x}))d\hat{x}\right)dx \tag{2.11}$$

$$-\frac{1}{|R_-|}\int_{R_-}\log\left(\frac{1}{|R_-|}\int_{R_-}K(G(x)-G(\hat{x}))d\hat{x}\right)dx \tag{2.12}$$

$K(.)$ is the Gaussian kernel $K(z)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{z^2}{2\sigma^2}}$ where σ is a scalar parameter. For details of the derivations we refer reader to [11].

- Gradient Flows for Region Integrals

Computing a gradient flow for a general energy functional $E(C)$ is a difficult task. However, in most cases the energy functional is given in the form of region integrals.

$$E(\vec{C}) = \int_R f(x)dx \tag{2.13}$$

where $f(.)$ does not depend on the curve C . The gradient flow that decreases this type of region integral most rapidly is given by:

$$\frac{\partial \vec{C}}{\partial t} = -f\vec{N} \tag{2.14}$$

For detailed derivations we refer reader to [11].

Similarly for a region integral

$$E(\vec{C}) = \int_{R_C} f(x)dx \tag{2.15}$$

The gradient flow is obtained by flipping the sign of outward normal vector N,

$$\frac{\partial \vec{C}}{\partial t} = f\vec{N} \tag{2.16}$$

In the region integrals (2.13) and (2.15), the integrand does not depend on the curve. Computing gradient flow for nested integrals is more complicated. Consider the following nested integral of the form:

$$\int_R f(\varepsilon(x,t))dx \quad \text{where } \varepsilon(x,t) = \int_R f(g(x,\hat{x}))d\hat{x} \tag{2.17}$$

where $g(\cdot,\cdot)$ does not depend on $\vec{C}$ given its arguments, R is the region inside the curve $\vec{C}$, and t is a time index for the evolution of $\vec{C}$. Note that $\varepsilon(x, t)$, which is a region integral, depends on $\vec{C}$ for an arbitrary fixed value of the argument x, whereas $g(x, \hat{x})$ is a function of only x and $\hat{x}$ and does not depend on $\vec{C}$ given x and $\hat{x}$. For nested region integrals of the form (2.17), the gradient flow is as follows:

$$\frac{\partial \vec{C}}{\partial t} = -\left[f(\varepsilon(\vec{C})) + \int_R f'(\varepsilon(x))g(x,\vec{C})dx\right]\vec{N} \tag{2.18}$$

All the curve evolution equations mentioned in this section involve motion in the normal direction. For this reason a tangential component of the velocity does not have anything to do with the geometry of the evolving curve and only reparameterizes the curve.

2.1.2) Level Set Methods:

[1] The level set method is a numerical technique for tracking interfaces and shapes. The advantage of the level set method is that one can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize; which is called the Eulerian approach; these objects. Also, the level set method makes it very easy to follow shapes which change topology, for example when a shape splits in two, develops holes, or the reverse of these operations.
The Lagrangian approach first discretizes the boundary into segments (2D) or triangles (3D) and evolves the endpoints (marker points) of these segments or triangles. Lagrangian approach requires very small time steps for stable evolution of the boundary [26] and in the case of topological changes it requires an ad-hoc procedure such as removal of the marker points on the portions of the boundary that have disappeared during the merging. Since level set methods are Eulerian approaches they are more stable and also even more powerful when dealing with the evolution of 3D boundaries than the Lagrangian approach.

In this section, we provide some background on level set methods for curve evolution. In particular, evolution equations for a level set function, properties of such evolution equations for level set functions, and some issues on numerical implementation will be discussed.

- Implicit Representation of Boundary by Level Set Function

The concept of implicit representation of boundaries is based on a closed curve C in R_2 which divides the image domain Ω into three parts:

- the region inside the curve R,
- the region outside the curve R_c,
- the boundary C .

The idea of Osher and Sethian's work [19] is to define a smooth function $\varphi(\mathbf{x})$ such that the set where $\varphi(\mathbf{x}) = 0$ represents the boundary C. Such a function φ is said to be a level set function for the boundary C, if φ has the following property:

$$\begin{aligned} \varphi(\mathbf{x}) &< 0 \text{ for } \mathbf{x} \in R \\ \varphi(\mathbf{x}) &> 0 \text{ for } \mathbf{x} \in R_c \\ \varphi(\mathbf{x}) &= 0 \text{ for } \mathbf{x} \in C \end{aligned} \tag{2.19}$$

[1] Background material about level sets is taken from Junmo Kim's PhD. thesis [11].

There are many such level set functions given a boundary C, but given a level set function the boundary is uniquely determined. In this sense, the level set function has redundancy in representing a boundary.

Since the gradient $\nabla\varphi$ is normal to equipotential lines $\{x|\varphi(x) = c\}$, ∇c, $\nabla\varphi$ evaluated at a point on the zero level set $\{x|\varphi(x) = 0\}$ is normal to the curve. Thus the outward unit normal to C is given by

$$N = \nabla\varphi / |\nabla\varphi| \tag{2.20}$$

The curvature κ of C is defined as the divergence of the normal N as follows:

$$\kappa = \nabla \cdot (\nabla\varphi / |\nabla\varphi|) \tag{2.21}$$

The characteristic function χ of a region inside the curve is given by;

$$\chi(x) = H(-\varphi(x)) \tag{2.22}$$

where the Heaviside function $H(z)$ is

$$H(z)=1 \text{ if } z \geq 0$$

$$H(z)=0 \text{ if } z < 0$$

Similarly, the characteristic function of a region outside the curve is $H(\varphi(x))$.

Signed Distance Function

A level set function φ is said to be a *signed distance function* if it not only satisfies the condition (2.19) but also its magnitude is the distance to the boundary,

i.e. $|\varphi(x)| = \min_{x_I \in C} d(x, x_I)$. If φ is a signed distance function, it satisfies the Eikonal equation [30]

$$|\nabla\varphi| = 1 \tag{2.23}$$

For signed distance functions, geometric quantities such as the outward normal vector or curvature are much simpler to compute.

The outward normal vector is given by,

$$N = \nabla\varphi \tag{2.24}$$

and the curvature is given by

$$\kappa = \Delta\varphi \tag{2.25}$$

where φ is the Laplacian of $\Delta\varphi$ defined as

$$\Delta\varphi = \varphi_{xx} + \varphi_{yy} \tag{2.26}$$

Using signed distance functions not only simplifies computations of several quantities, but also makes these computations stable. When we implement an evolution of a curve by evolving the corresponding level set function, a numerical error can

accumulate and the level set function can develop noisy features. For this reason, it is always advisable to reinitialize a level set function to a signed distance function occasionally during the evolution of the curve. For such a reinitialization, Tsitsiklis [33] developed a fast algorithm, the so-called fast marching method, and we refer the readers to [20, 33] for details.

□

- Evolution Equations for Level Set Functions

Suppose that the curve evolution is specified by a velocity field $V(x)$, whose values are defined on every point x on the evolving curve $C(t)$. With the velocity V, we can describe the motion of each point on the curve $C(p, t)$ as follows:

$$\partial C(p, t)/\partial t = V(C(p, t)), \ \forall\, p \in [0, 1] \tag{2.27}$$

We can rewrite the above equation in vector form:

$$\partial C/\partial t = V(C) \tag{2.28}$$

These equations are the *Lagrangian* formulations of the curve evolution equation.

When the curve $C(t)$ evolves according to (2.28), the zero level set $\{x|\varphi(x, t) = 0\}$ will evolve in exactly the same way. Now we derive the update equation for the level set function. The level set function $\varphi(\mathbf{x}, t)$ and the boundary $C(t)$ are related by the following equation:

$$\varphi(C(p, t), t) = 0 \text{ for all } p, t. \tag{2.29}$$

$$\varphi(x, t) < 0 \text{ if } x \text{ is inside } C(t) \tag{2.30}$$

$$\varphi(x, t) > 0 \text{ if } x \text{ is outside } C(t) \tag{2.31}$$

By differentiating (2.29) w.r.t. t, we obtain

$$\varphi_t(C(p, t), t) + \nabla \varphi(C(p, t), t) C_t(p, t) = 0 \tag{2.32}$$

Substituting the velocity $V(C(p, t))$ for $C_t(p, t)$ gives the following PDE for the update of level set function.

$$\varphi_t(C(p, t), t) + V(C(p, t))\ \nabla \varphi(C(p, t), t) = 0, \text{ for all } p, t \tag{2.33}$$

We can rewrite (2.33) in vector form

$$\varphi_t(C) + V(C)\ \nabla \varphi(C) = 0. \tag{2.34}$$

or simply

$$\varphi_t + V \cdot \nabla \varphi = 0 \qquad (2.35)$$

Since N and $\nabla \varphi$ point in the same direction, $T \cdot \nabla \varphi$ is zero for any tangent vector T. Hence, if we write $V = V_n N + V_t T$, we have

$$\varphi_t(C) + V_n(C)N \nabla \varphi(C, t) = 0 \quad \text{for all } t \qquad (2.36)$$

indicating that only the normal component of V is relevant for the evolution of the level set function. Since $N = \nabla \varphi / |\nabla \varphi|$, we have

$$\varphi_t + V_n |\nabla \varphi| = 0 \qquad (2.37)$$

In equation (2.28) the velocity V is defined only on the boundary, but for (2.35) we need the velocity everywhere on the field. Hence, we need to choose the velocity value off the boundary. When we decide to use a signed distance function, and if we want to constrain the evolving level set function remains a signed distance function, then the velocity on the boundary is sufficient to specify the velocity at every point $x \in \Omega$. Such an evolving level set function remains a signed distance function if and only if the velocity of the boundary remains constant along the normal direction to the boundary [39]. The speed of the level set implementation of the curve evolution can be increased by the narrow band method. We refer the reader to [1, 4, 20, 26] for the details of this method.

2.2) TEXTURE ANALYSIS

There are many application areas of image processing in which texture plays important roles. The most important application areas are biomedical image analysis, industrial inspection, analysis of satellite imagery, content-based retrieval from image databases, document analysis, biometric person authentication, scene analysis for robot navigation, texture synthesis for computer graphics and animation, and image coding. Image segmentation is related to the classification problem; when other features, like shape, tone or color, are not sufficient to discriminate between regions, one criterion of image segmentation might be texture.

This section aims to build the background knowledge on texture analysis. First, we discuss the concept of texture. Then, we describe some texture analysis methods and make a taxonomy of them.

2.2.1) What is Texture?

There are several thoughts about the definition of texture. Even though texture is an intuitive concept, a formal definition of texture has proven elusive in the literature over the years. Leopards have spots but tigers have stripes, curly hair looks different from straight hair, etc. but in all of these examples there are variations of intensity and color which form certain repeated patterns called visual texture. The patterns can be the result of physical surface properties such as roughness or oriented strands which often have a tactile quality or they could be the result of reflectance differences such as the color on a surface. We can recognize texture when we see it but, even though the concept of texture is intuitive, a precise definition of texture has proven difficult to formulate. Some scientists noted this difficulty. Bovik, Clarke and Geisler said "An exact definition of texture either as a surface property or as an image property has never been adequately formulated." Jain and Karu noted "Texture eludes a formal definition."

In the last thirty years many researcher tried to define texture. It is not possible to mention all of them here but, in 1979 Haralick [9] proposed the following definition of texture, "Image texture is described by the number and types of its primitives and the spatial organization or layout of its primitives. The spatial organization may be random, may have a pair-wise dependence of one primitive on a neighboring primitive, or may have a dependence of *n* primitives at a time. The dependence may be structural, probabilistic, or functional."

Some researchers described texture in terms of the human visual system. They claim textures do not have uniform intensity, but are none-the-less perceived as homogeneous regions by a human observer. For example, Bovik, Clarke, and Geisler write, "an image texture may be defined as a local arrangement of image irradiances projected from a surface patch of perceptually homogeneous irradiances." Also, Chaudhuri, Sarkar, and Kundu write, "Texture regions give different interpretations at different distances and at different degrees of visual attention. At a standard distance with normal attention, it gives the notion of macro-regularity that is characteristic of the

particular texture. When viewed closely and attentively, homogeneous regions and edges, sometimes constituting texels are noticeable." However, a definition based on human acuity poses problems when used as the theoretical basis for a quantitative texture analysis algorithm. Faugeras and Pratt note, "The basic pattern and repetition frequency of a texture sample could be perceptually invisible, although quantitatively present."

There are several trials to define texture in the literature. Apparently no one has succeeded in finding a formal or commonly accepted definition of texture. There are two accepted thoughts about texture in the literature.

-There is significant variation in intensity levels between nearby pixels; that is, at the limit of resolution, there is non-homogeneity.

-Texture is a homogeneous property at some spatial scale larger than the resolution of the image.

Maria Petrou and Pedro Garcia Sevilla defined texture in their book [7] based on these two accepted thoughts as "Texture is the variation of data at scales smaller than the scales of interest." For example, think about the following lady wearing a dress in figure [2.1].

Figure (2.1): Lady with flower textured dress

If we are interested in identifying the lady in the image, the pattern on the dress is considered as texture but, if we are interested in identifying a flower on the dress, a flower in the pattern is not a textured object. The reason is, it is impossible to recognize texture structure of a flower at this resolution of image. Haralick provides the following

claim about this situation: "A fundamental characteristic of texture is it cannot be analyzed without a frame of reference of tonal primitive being stated or implied. For any smooth gray tone surface, there exists a scale such that when the surface is examined, it has no texture. Then as resolution increases, it takes on a fine texture and then a coarse texture."

Identifying the perceived qualities in an image is an important first step towards building mathematical models for texture. The intensity variations in an image which characterize texture are generally due to some underlying physical variation in the scene such as pebbles on a beach or waves in water. Modeling this physical variation is very difficult, so texture is usually characterized by the two-dimensional variations in the intensities present in the image. The definitions of texture in the literature lead to a variety of different ways to analyze texture. In following section, an overview of various texture analysis approaches will be provided.

2.2.2) Texture Analysis Approaches:

a) Statistical Approaches

Statistical approaches define the texture as “a quantitative measure of the arrangement of intensities in a region”. Each texture is described by a feature vector of properties, which represents a point in a multi-dimensional feature space. The statistical approaches include spatial domain methods, frequency based methods, and model based methods.

-Spatial domain methods:

In spatial domain methods, features are derived from especially second order statistics of the texture gray level values because human beings are sensitive to second-order statistics. Examples of such statistics are the gray level co-occurrence matrix, autocorrelation function and the gray level variance matrix.

-Frequency based methods:

Frequency based methods like Gabor filters and Wavelet transformations take the visual system as its model and decomposes images into a number of band-pass filtered images, each of which contains intensity variations over a narrow range of frequency (size) and orientation, which will resemble certain textural properties. Psychophysical experiments have shown that methods based on spatial-frequency domain features are plausible with the human visual system. In recent years, by the

development of the wavelet theory this method has received special interest, and a number of algorithms based on wavelet transforms and Gabor filters have been proposed for classification and segmentation of textured images.

-Model based methods:

Model based texture analysis methods try to capture the process that generated the texture by determining the parameters of a predefined model. Model based methods assume textured images as realizations or samples from parametric probability distributions on the image space, and try to fit e.g. simultaneous autoregressive models, Markov random field models, and fractal models to the textured image. Model parameters are used as features in the classification and segmentation problems. Advantage of this method over other statistical methods is that the model parameters can be used not only to describe the texture but also to synthesize it. They have been used for classification, segmentation and compression of textured images.

b) Structural approaches:

Structural approaches (Haralick 1979, Levine 1985) represent texture by well defined primitives (*microtexture*) and a hierarchy of spatial arrangements (*macrotexture*) of those primitives. To describe the texture, one must define the primitives and the placement rules. The choice of a primitive (from a set of primitives) and the probability of the chosen primitive to be placed at a particular location can be a function of location or the primitives near the location. The advantage of the structural approach is that it provides a good symbolic description of the image; however, this feature is more useful for synthesis than analysis tasks. The abstract descriptions can be ill defined for natural textures because of the variability of both micro and macrostructure and no clear distinction between them. A powerful tool for structural texture analysis is provided by mathematical morphology (Serra 1982, Chen 1994).

In this thesis Local Binary Partition (LBP) method has been used for texture analysis which is a unifying approach to structural and statistical texture analysis. LBP method will be discussed in the following section.

2.3) TEXTURE SEGMENTATION:

The concept of texture segmentation can be defined as: “the partitioning of an image into regions, each of which contains a single texture distinct from its neighbors.” However, this definition does not exactly explain the main difficulties and practical limitations involved in the texture segmentation problem. To make an exact definition, we should consider these two terms, “texture” and “segmentation”, separately.

Image segmentation can be defined as giving a label to each pixel of an image consisting of regions. A region is a group of connected pixels that share the same label. Ideally, each region that represents a different “object” in the image forms a “proper region”. However, what is an object? If a bookshelf is filled with books, do we want to consider each book as a separate object, or do we want the bookshelf and everything in it to be a single object? If we see a computer monitor sitting on a console with a keyboard attached to it, is there one object or three?

Even more slippery is the notion of texture. Whereas segmentation has at least a formal definition, as mentioned before a formal definition of texture has proven elusive in the literature over the years. It is not even clear whether texture is an inherent property of all things, or whether some objects or regions lack texture altogether.

There are two approaches to define texture, which may be thought of as ``top-down" and ``bottom-up." Top-down models claim that there is a basic element, called a texel or a texton, and a placement rule, and the rule defines how and where the elements are placed. This definition works well if the texture consists of e.g. bricks or piles of pennies. The bottom-up approach claims that texture is a property that can be derived from the statistics of small groups of pixels, such as mean and variance. This works better for textures like quartz and grass where it is difficult to see individual elements. The dividing line between the two approaches is by no means clear.

Putting the two terms back together, we see the difficulties of the task. We wish to partition an image into regions of homogeneous texture, but we cannot always agree on when two texture samples are similar to each other. Furthermore, two different objects with a common boundary may have the same texture and be lumped together, which may or may not be desired. Texture segmentation will invariably break up certain objects which contain multiple textures, and group together pieces of different objects

into one region.

Until this section we have mentioned active contour based image segmentation and texture analysis methods. In the recent years there have been texture segmentation approaches which unify active contour based image segmentation and texture analysis. Paragios and Deriche proposed a variational method for texture segmentation named "Geodesic active contours for supervised texture segmentation"[22] which is based on curve propagation theory. They have generated the texture space by filtering the input images using Gabor filters and they have analyzed their responses as multi-component conditional probability density functions. They obtained the texture segmentation by minimizing a geodesic active contour model objective function; where the boundary information is expressed by discontinuities in the multi modal textured feature space; by using gradient descent. Sagiv, Sochen, and Zeevi [25] also used a Gabor feature space and extracted a two dimensional Riemannian manifold of local features that are extracted by Beltrami framework. They have used this Beltrami-based diffusion mechanism in a geodesic active contours algorithm for texture segmentation. Unal, Yezzi and Krim [34] proposed a new active contour model which nicely ties the desirable polygonal representation of an object directly to the image segmentation process. This model captures texture boundaries by way of higher-order statistics of the data and using an information-theoretic measure and with its nature of the ordinary differential equations. As a different kind of approach Wang and Vemuri [36] presented a region-based active contour model for tensor field segmentation and show its application for texture segmentation. In [37] a shape-based approach has been proposed for texture segmentation based on Gabor filtering.

In this thesis we have used Local Binary Partition (LBP) method for texture analysis. We provide detailed background of LBP in the following subsection. Our approach is different from [37] both in terms of the texture filter used and in terms of the way shape information is incorporated.

2.3.1) Local Binary Pattern (LBP) Method:

The local binary pattern (LBP) is a non-parametric operator which is used for describing the local spatial structure of an image. Local binary pattern has been first introduced by Ojala et al.[17] as a high discriminative powered texture analysis technique. At a given pixel position, LBP is defined as an ordered set of binary comparisons of pixel intensities between the center pixel and its neighboring pixels. LBP operator labels the pixels of an image by using the value of the center pixel as a threshold value at the 3 by 3 neighborhood of every pixel. If the neighboring pixel value is greater than or equal to the center pixel value this pixel takes 1 otherwise 0. Then an 8-bit LBP code for a neighborhood is formed. The order of the digits in 8-bit binary code can be, provided that we are consistent in our choice throughout the process, arbitrary. The decimal value of this binary code gives the local structural information around the given pixel (Figure 2.2).

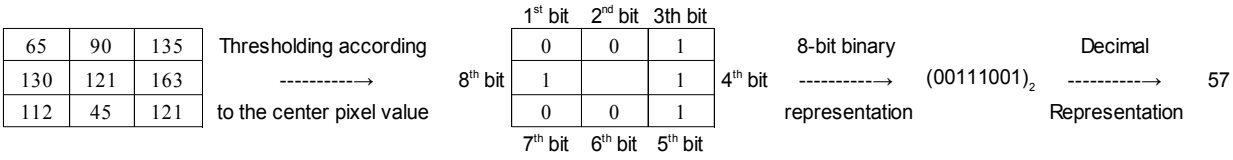


Figure 2.2: Graphical view of the LBP procedure. After the thresholding step, the top-left digit is assigned as the first digit and the 8-bit binary code is written in the clockwise order. For the last step the decimal value of the binary representation is found which is our decimal LBP code for the center pixel.

By applying this procedure we form our LBP image which has pixel values ranging between 0 and 255. If all the intensity values are equal or higher than the center pixel, the LBP value for this center pixel is $(11111111)_2 = 255$ otherwise, if all the values are lower than the center pixel, the LBP value is $(00000000)_2 = 0$. Every LBP value corresponds to a different pattern. When we take the histogram of the LBP image, it shows us, how often each of these 256 different patterns appears in a given texture. The distribution of these patterns in the 256-bin histogram defines the whole structure of the texture.

However, it is possible to decrease the number of patterns in the LBP histogram by using uniform patterns [18]. An LBP pattern is a uniform pattern if it contains at most two bitwise transitions from 0 to 1 or 1 to 0 at its 8-bit binary representation. The binary string is considered circular (Figure 2.3).

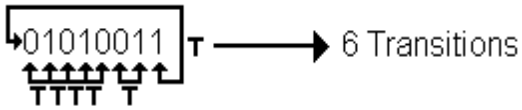


Figure 2.3: Circular binary string and 0 to 1 and 1 to 0 transition calculation. (T=transition)

For example, 00000000(with 0 transition), 11111111(with 0 transition), 11100000(with 2 transitions), 11100001(with 2 transitions) are uniform patterns. 01010011(with 6 transitions), 11110101(with 4 transitions) are non-uniform patterns. There are totally 58 different uniform patterns in 8-bit LBP representation and when we assign the remaining patterns in one non-uniform binary number we can represent our texture structure with a 59-bin histogram. According to Ojala et al.[18] the results of their experiments on different texture images shows that uniform patterns correspond to about 90% of all possible patterns in a texture when using 8-bit binary representation. There is also 8-bit binary representation with 2 pixel radius and 16-bit binary representation with 2 pixel radius of LBP (Figure 2.4).

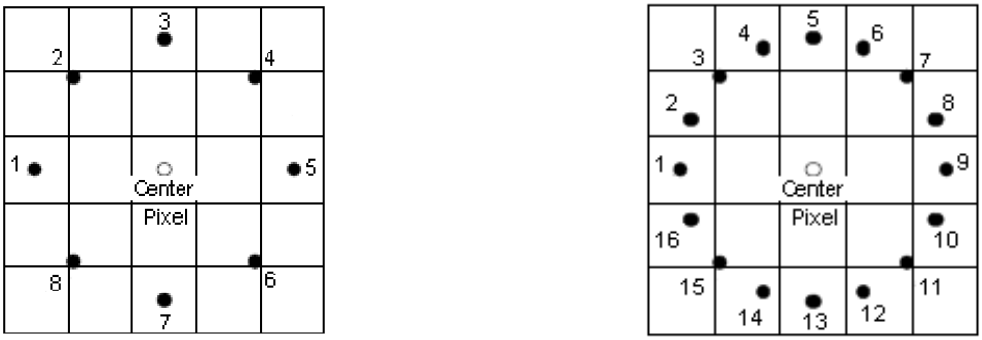


Figure 2.4: 8-bit binary representation with 2 pixel radius and 16-bit binary representation with 2 pixel radius of LBP

The gray values of neighbors which do not fall exactly in the center of pixels are estimated by interpolation. Inserting extra eight neighbors to the 3x3 neighborhood would not obtain too much new information. For this reason in 16-bit representation the size of the local neighborhood increased to 5 by 5 pixels and instead of 8-bit code 16-bit binary code; which enables the definition of spatial structure in a larger area; is used. The final texture feature employed in texture analysis is the histogram of the operator outputs, pattern labels, accumulated over a texture sample.

Two images or regions are compared by computing the L_1 distance between their histograms defined as:

$$L_1(H_1, H_2) = \sum_{i=1}^{n} |H_1[i] - H_2[i]| ,$$

where n is the bin numbers in LBP histograms H_1, H_2.

For a given textured image with two textured regions, texture segmentation procedure by using LBP is as follows: In training phase the images of the textures are converted to LBP domain and the LBP histograms are calculated. In testing phase testing image is also converted to LBP domain. Then test image is divided to m-by-m windows. For every window the LBP histograms are calculated for comparing their histograms with the training image histograms by using L_1 distance. Finally, according to the L_1 distance results, the class to which the region belongs is assigned. The region with minimum error is assigned to the region we want to segment.

The LBP is a powerful illumination invariant texture analysis operator. LBP method can be seen as a unifying approach to structural and statistical texture analysis since it is a histogram based method that the histogram represents the distribution of decimal representations of binary codes which define the structural information around every pixel in textured images. Besides, its computational complexity is quite low. Therefore, LBP has recently become one of the popular texture analysis methods.

2.4) Shape Analysis:

The shape of an object is a binary image representing the extent of the object which can be thought of as a silhouette of the object. The applications of shape analysis include object recognition, medical image analysis, and image segmentation. The aim of shape analysis is to provide a simplified representation of the original shape so that the important characteristics of the shape are preserved. The result of shape analysis can be either numeric (e.g. a vector) or non-numeric (e.g. a graph).

Kendall [10], Small [28] and Cootes et al. [5] have represented the shape of an object using landmarks. Cootes et al. [5] have proposed a technique called 'Active Shape Models'. In this technique Cootes has computed typical variability from a set of training shapes by using principal component analysis (PCA). For landmarks of good quality there should be one to one correspondence between landmarks in training shapes. However, automatically choosing good landmarks is not straightforward. For

this reason landmarks are often chosen manually. Since, the performance of landmark approaches depend on the quality of landmarks, this situation is the drawback of such approaches. Later, Cootes [7, 8], proposed a way to automate the choice of landmarks using the minimum description length (MDL) criterion, but the process is still computationally expensive.

In recent years, level set approaches became popular in shape analysis techniques. A shape is represented as a zero level set of a signed distance function [14,23,32] and shape priors are also built by signed distance functions of the training shapes. Tsai and Yezzi [32] used PCA of the signed distance functions of training data to capture the variability in shapes. Also this approach is useful in segmenting low SNR images or occluded images. However, the major problem about this approach is the space of signed distance functions is not closed under linear operations. For example, the direct average of distance functions (mean shape), is not a distance function.
For this reason, using linear analysis tools such as PCA causes inconsistency for shape modeling [32]. Paragios et al. [23] find a mean shape in the space of distance functions to get rid of this problem. They obtain the mean shape estimate by evolving a shape in the direction of reduction in distance from example shapes and reduction in distance from the space of signed distance functions. Cremers and Soatto [6] proposed several distance measures between two signed distance functions for shape-based image segmentation.

Soatto and Yezzi [29] proposed a method of extracting both the motion and the deformation of moving deformable objects. They proposed the notion of shape average and motions such that all the example shapes are obtained by rigid transformation of the shape average followed by diffeomorphism (deformation), where the shape average and motions are defined such that the total amount of deformation is minimized. In that work, the amount of such diffeomorphism is measured by a simple template metric.

In this thesis we have used a nonparametric shape prior model in the space of curves and the space of signed distance functions proposed by Kim et al. [11]. They estimate the underlying shape distribution of training shapes, by extending a Parzen density estimator to the space of shapes. Such density estimates are expressed in terms of distances between shapes. Suppose that we have an infinite dimensional space from example curves $\tilde{C}_1,...,\tilde{C}_n$, or signed distance functions $\tilde{\varphi}_1,..., \tilde{\varphi}_n$. Estimating the unknown shape probability density estimates $p_{\tilde{C}}(\tilde{C})$ or $p_{\tilde{\varphi}}(\tilde{\varphi})$, over infinite dimensional space is the challenging problem. To understand their approach easier let us consider their

approach for a finite dimensional random vector. Suppose that we have n samples x_1, x_2,..., $x_n \in R^m$ drawn from an m-dimensional density function p(x). The Parzen density estimate of p(x) is:

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} k(x - x_i, \Sigma) \tag{2.38}$$

In equation (2.38), k(.) is the m-dimensional Gaussian kernel $k(x, \Sigma) = N(x; 0, \Sigma^T \Sigma)$. If the kernel is spherical, i.e. $\Sigma = \sigma I$, equation (2.38) becomes

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} k(d(x, x_i), \sigma) \tag{2.39}$$

$d(x,x_i)$ is the Euclidean distance between x and x_i in R^m, and $k(x, \sigma)$ is the one dimensional Gaussian kernel $k(x, \sigma) = N(x; 0, \sigma^2)$.

They extend this Parzen density estimator with a spherical Gaussian kernel to the infinite dimensional space of curves C or space of signed distance functions D by using a distance measure d_C in C or d_D in D as follows:

$$\hat{p}_{\tilde{C}}(\tilde{C}) = \frac{1}{n}\sum_{i=1}^{n} k(d_C(\tilde{C}, \tilde{C}_i), \sigma) \tag{2.40}$$

$$\hat{p}_{\tilde{\varphi}}(\tilde{\varphi}) = \frac{1}{n}\sum_{i=1}^{n} k(d_D(\tilde{\varphi}, \tilde{\varphi}_i), \sigma) \tag{2.41}$$

$\tilde{C}$ and $\tilde{\varphi}$ represents the aligned curves in curve space and signed distance functions space respectively. In our algorithm we use the alignment method in [31]. For our shape space we align the shapes to eliminate the variations in shapes due to pose differences. In shape analysis methods distance metrics are very important to measure similarity and dissimilarity between shapes. Several metrics for the space of shapes have been proposed. Some of them are; Hausdorff Metric, Template Metric, Transport Metric, Optimal Diffeomorphism. For detailed explanations of these metrics we refer the reader to [11]. The nonparametric shape priors in (2.40) and (2.41) can be used with several kinds of metrics. In this thesis we have used the template metric and the *L2* distance between signed distance functions. Details of Parzen density estimation are provided at the following sections.

2.5) Nonparametric Density Estimation:

Most statistical approaches use probability density functions. For statistical analysis we need to know the underlying densities but in most cases it is not possible. For this reason we need to estimate the underlying densities. Parametric density estimation approaches assign a mathematical structure to the density and estimate the probability density function by estimating the parameters of the mathematical structure of the assigned density. Nonparametric density estimation approaches learn the density from the samples drawn from an unknown density, with some mild assumptions like smoothness of the density function. Compared to nonparametric approaches, parametric approaches have low computational cost but on the other hand nonparametric approaches have more modeling capability than the parametric approaches.

In this thesis we have used a Gaussian kernel based nonparametric density estimation by using Parzen density estimator to estimate the density for image intensity and shape densities. In this section we will provide some background about Parzen density estimation.

2.5.1) Parzen Density Estimator:

Emanuel Parzen proposed his nonparametric density estimation approach [24] in 1961 to estimate an unknown density p(x) from N independent identically distributed (i.i.d.) samples ($x_1 \ldots x_N$) drawn from p(x). Parzen density estimate which is a kernel based density estimator is as follows:

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sigma}k(\frac{x-x_i}{\sigma}) \tag{2.42}$$

In equation (2.42) $k(\cdot)$ is the kernel function which satisfies $\int k(x)dx = 1$ and $k(\cdot) \geq 0$, and σ is the kernel size. In this thesis we use a Gaussian kernel which is formulated as follows:

$$k(x,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-x^2/(2\sigma^2)} \tag{2.43}$$

For Gaussian kernels we use the following notation

$$k(x,\sigma) \text{ denote } \frac{1}{\sigma}k(\frac{x-x_i}{\sigma}) = N(x;0,\sigma^2)$$

Then we can write the Parzen density estimate as follows:

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} k(x - x_i, \sigma) \tag{2.44}$$

We can think of the Parzen density estimate as a smoother version of the histogram. We can create a histogram by using a uniform kernel with Parzen density estimate. For details see[11].

To understand the concept of Parzen density estimation the following example might be useful:

Suppose that we have the following density p(x) in figure (2.5) which is a mixture of Gaussians $p(x) = 0.7N(x; -5, 2^2) + 0.3N(x; 5, 2^2)$.

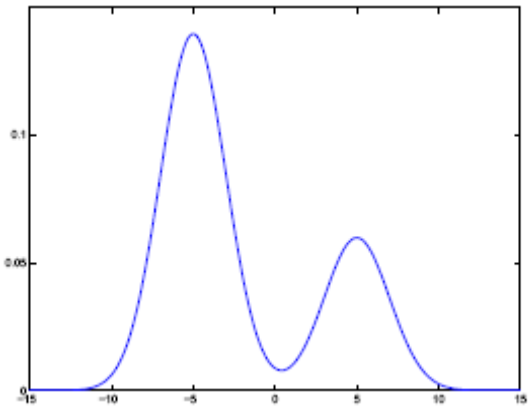


Figure (2.5): True density which is mixture of Gaussians $p(x) = 0.7N(x; -5, 2^2) + 0.3N(x; 5, 2^2)$

We draw 10 samples from the true density shown in figure (2.42) and estimate the density by using these samples and Parzen density estimator with a Gaussian kernel. The resulting density estimate is as follows in figure (2.6):

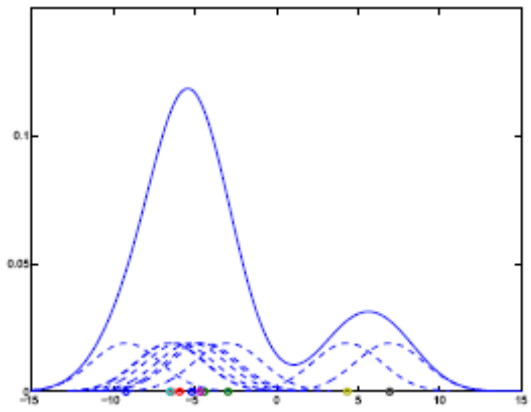


Figure (2.6): Parzen density estimate. Circles are the drawn samples, dashed lines are contribution of each kernel and the solid line represents the density estimate.

Kernel size has an important role for good density estimation. Changing the kernel size gives us different density estimates. The larger the kernel size, the spreader the density estimate, and the smaller the kernel size the peakier the density estimate. For accurate density estimation, proper choice of the kernel size is more important than the choice of the kernel shape [27]. For details of choosing a good kernel size we refer reader to [11,12,13].

2.5.2) Entropy Estimation:

In this subsection we provide some background about entropy estimation. In this thesis we have used Kim et al.'s information theoretic approach for image segmentation [11,12] which uses mutual information which is given in terms of entropies. In their approach they estimate the entropies from data by using a nonparametric density estimator with mean square consistency which is proposed by Ahmad and Lin [2]. To estimate the entropy $h(X)$; a functional of the underlying density; i.i.d. samples $X_1, \ldots, X_N$ are drawn from the unknown pdf $p(x)$. They avoid the integration in differential entropy

$$h(X) = -\int p(x) \log p(x) \tag{2.45}$$

$$= -E[\log p(X)] \tag{2.46}$$

and approximate the expectation by a sample mean as follows:

$$h_N = -\frac{1}{N}\sum_{i=1}^{N} \log \hat{p}(X_i) \tag{2.47}$$

Where h_N is the entropy estimate. If we substitute our Gaussian kernel based Parzen density estimate $\hat{p}(X_i)$ to equation (2.47) our entropy estimate with mean square consistency is as follows:

$$h_N = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{1}{N}\sum_{j=1}^{N} k(X_i - X_j, \sigma) \tag{2.48}$$

and mean square consistency is as follows:

$$\lim_{N\to\infty} = E(h_N - h(X))^2 = 0 \tag{2.49}$$

In this thesis we have used this entropy estimator to estimate the entropy of an intensity distribution over a certain region of an image. The problem with this approach is

computational complexity. For this reason Kim et al.,used the fast Gauss transform to speed up computing Parzen density estimates. For details see [11].

CHAPTER 3

Shape and Data Driven Texture Segmentation Using Local Binary Pattern (LBP)

There are several cases where intensity values, color, mean or variance of image intensity distributions and edge information cannot play a discriminative role in image segmentation. Depending on the problem, in such cases texture might be a good feature. A basic example for such cases can be an image which consists of two different textured regions with similar first order pdfs in the foreground and in the background. For such a problem the intensity distributions of these regions cannot play a discriminative role in the segmentation process. In this case, while pixel intensity-based features are not sufficient to discriminate regions, texture based features have a better chance to handle the segmentation problem. Texture analysis and texture segmentation are still challenging problems; there is no method which can clearly discriminate all kind of textures. Especially defining nonuniform structures of textures and discriminating one texture from another are still difficult problems in texture analysis. For this reason texture segmentation approaches may give unsatisfactory segmentation results with missing data or with corrupted boundaries of regions because of the wrong texture identification results of texture analysis methods. Using prior information about the shape of the object can aid segmentation which can also obtain a solution to occlusion problems.

In this chapter, we propose a shape and data driven texture segmentation method using local binary pattern. In particular, we train our LBP based texture filter with the texture which belongs to the region that we want to segment. Then we enter the textured image into our filter to produce a “filtered image” which means an image has been removed off its textural structure. In other words we aim to produce a filtered image which has independent pixel values as much as possible. For this purpose we are using a windowing process to compare the sample regions taken from the testing image with training texture in LBP domain. Windows are produced by considering every pixel in the image as center pixel of 17-by-17 windows. The window size depends on the

textural structure of test image. According to the texture structure if you choose a smaller window it is difficult to perceive the textural structure. In segmentation problems since we want to discriminate regions in an image from each other, using large window size would increase the inaccuracy. Also using larger window size will increase the computational load. The regions inside the windows are compared by calculating the L1 distance between their LBP histograms and the LBP histogram of training image. The L1 distance value of the compared window is assigned to the center pixel. The resulting image has low pixel intensity values where the regions that have similar texture characteristics with the training texture and has higher pixel intensity values where the regions that have different texture characteristics. With texture filtering process we reduce the texture segmentation problem to an intensity based image segmentation problem. Then by using nonparametric density estimation we estimate the unknown underlying probability density of the filtered image. We produce a data term for our energy equation which involves maximizing mutual information between the region labels and the image pixel intensities. Also we produce a nonparametric shape prior by estimating underlying shape distribution by using Parzen density estimator in space of shapes. Then we combine the data term and shape prior within a Bayesian framework to form the energy functional which we minimize using active contours for segmentation. The schematic representation of our approach is in Figure 3.1.

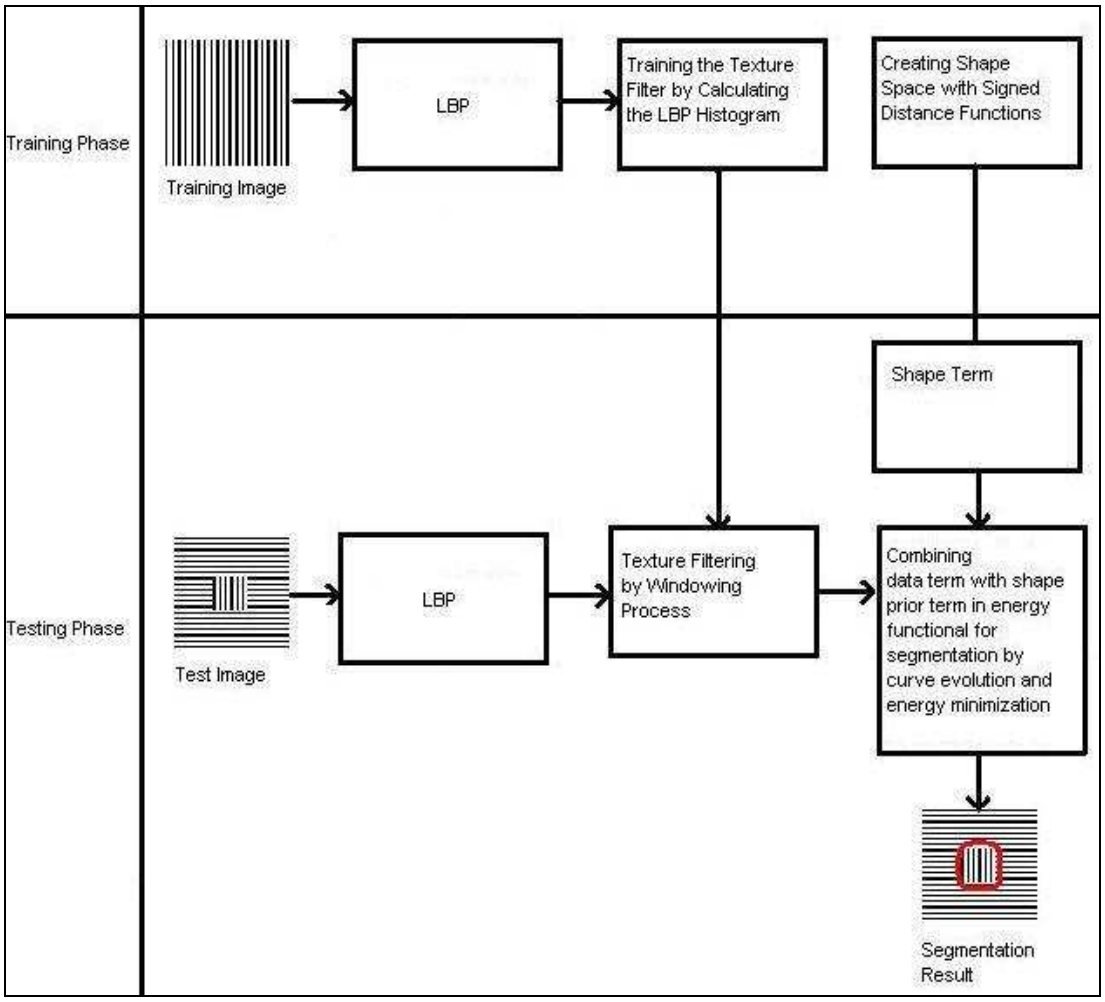


Figure 3.1: The schematic representation of our approach

Organization of this chapter is as follows. In Section 3.1 we present the LBP based texture filtering step in detail. Section 3.2 presents nonparametric density estimation in filtered image and the information theoretic data term. Section 3.3 presents nonparametric shape priors and shape term. Section 3.4 presents gradient flows for data and shape term for segmentation. Section 3.5 presents the experimental results.

3.1) Local Binary Partition Based Texture Filtering:

As we have mentioned in section 2.2 in textured images pixel intensities are structurally dependent on each other. For this reason treating image intensity distribution in a textured image as identical independently distributed (i.i.d.) is not a proper approach. For instance consider the following synthetic textured image in figure 3.2

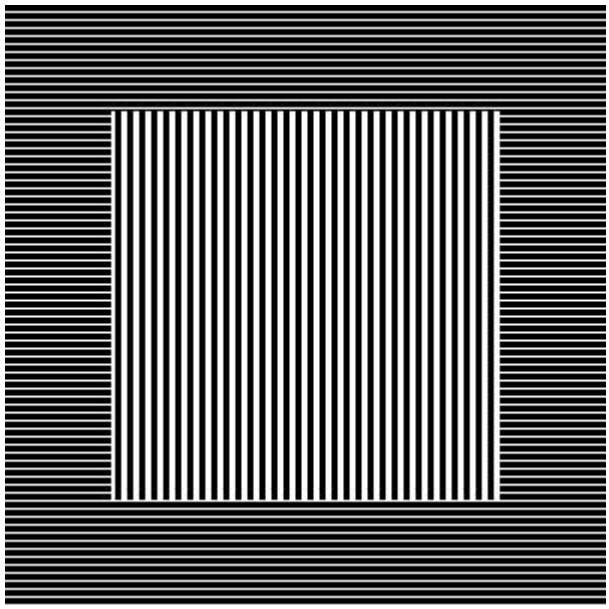


Figure 3.2 Synthetic textured image with same first order pixel intensity pdfs in two regions

In figure 3.2 there are two textured regions. The textures are different, but when we look at the histograms of the two textured regions individually, both of them have two peaks located at bins 0 and 255. Their intensity distributions are same but they are different regions. This situation leads us to use a texture analysis tool to produce a texture filter that obtains the opportunity to represent the textured regions separately and as much identical independently distributed (i.i.d.) as possible. In ideal cases it is expected texture filters to produce filtering results with i.i.d. pixel intensities, but depending on texture filter quality it is a difficult process to completely elude textured images from textural structures. In our approach we aim to reduce the texture segmentation problem to a pixel intensity value based image segmentation problem by using a texture filter. For this purpose we use a Local Binary Partition (LBP) based texture filter. We discussed LBP as a texture segmentation method in section 2.3.1. For this reason, in this section we discuss how we use LBP in our approach, rather than the

details of LBP.

Suppose that we have the image in figure 3.3.

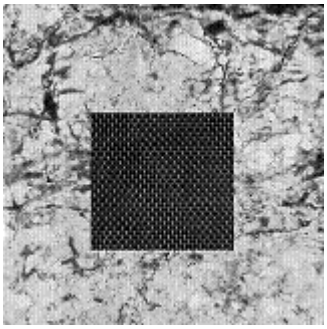


Figure 3.3: Textured image

We want to segment the textured square from the image. In this thesis we have used 8 bit representation of LBP with 2 pixel radius (Figure 3.4.).

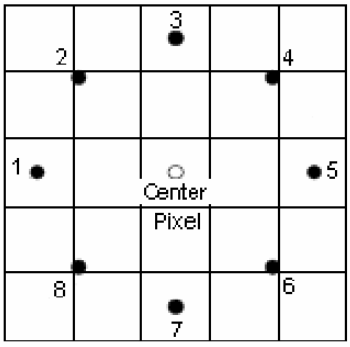


Figure 3.4: 8-bit binary representation with 2 pixel radius

The formulation of calculating the LBP of a pixel is as follows:

$$LBP_{8\_2}(x) = \sum_{i=1}^{8} s(G(x_i) - G(x))2^{i-1}, \tag{3.1}$$

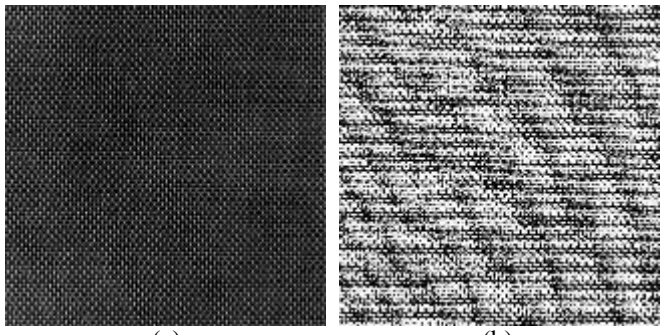
$$s(t) = \begin{cases} 1, t \geq 0 \\ 0, t < 0 \end{cases}, \tag{3.2}$$

where x is the location of the center pixel. x_i is the location of the ith neighboring pixel to center pixel.

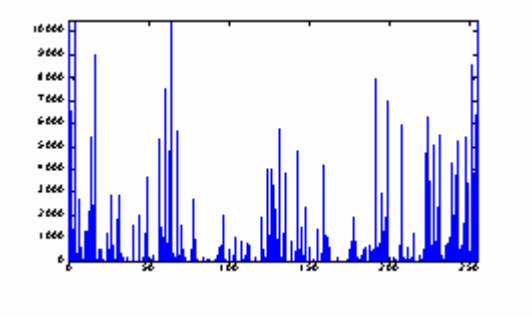
As we have mentioned briefly in the introduction of this chapter, our texture filtering approach works with histogram comparison principle which makes our texture segmentation method a supervised approach. For this reason we train our texture filter with the texture in Figure 3.5(a) which belongs to the region that we want to segment. Our training phase is actually to find out the structural information of the texture we want to segment by using a texture template.

In training image for every pixel at location x and intensity value G(x) we generate LBP value by equations (3.1) and (3.2) and produce a new image in LBP domain which is in Figure (3.5)(b). Then we calculate the histogram of the image which is in LBP domain (Figure 3.5(c)).

Note that since LBP values, which are discrete random variables, are produced from decimal values of 8 bit binary LBP code, it would be a wrong approach to treat an LBP histogram as a continuous histogram. LBP histograms represent the characteristic structure of textures by their bin numbers. For instance, in image domain when we produce the histogram of an image the closest values to bin number 116 are 115 and 117. On the other hand in the LBP domain histogram the bin number 116 is the decimal value of $(01110100)_2$. So, the decimal values of binary numbers with 1 bit difference from $(01110100)_2$ are the closest values for 116 in structural manner. (e.g.: 117, 118, 124, 244).

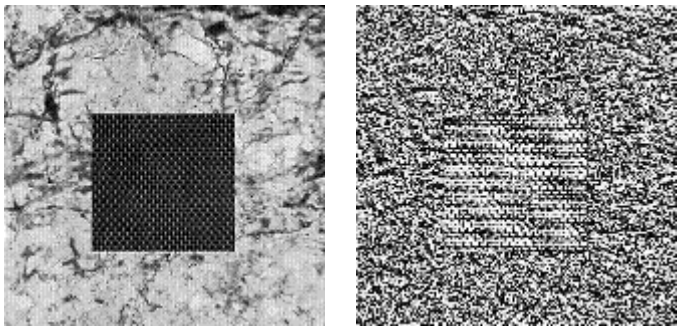


(a) (b)



(c)

Figure 3.5: **(a)** Training image **(b)** Training image in the LBP domain **(c)** LBP histogram of training image in LBP domain.



(a) (b)

Figure 3.6: **(a)** Test image **(b)** Test image in the LBP domain

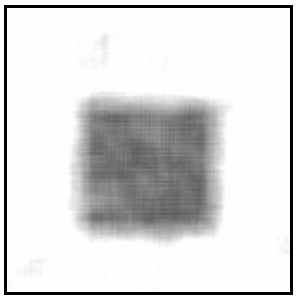


Figure 3.7: Filtered image

Then we input our test image Figure 3.6(a) to our filter and produce its representation in LBP domain (Figure 3.6(b)) by the same way that we use to produce for the training image.

In the LBP domain for every pixel in the test image we take a 17 by 17 window where the pixel is located at the center of this window. Then we compare this window's histogram with the normalized training image LBP histogram by using the L_1 distance metric. The formulation of the L1 distance between two histograms is as follows:

$$L_1(H_{Train}, H_{test}) = \sum_{i=1}^{n} |H_{Train}[i] - H_{test}[i]| \quad (3.3)$$

where H_{Train} and H_{test} are respectively the histograms of training and test images. *n* is the number of histogram bins. Then we assign $L_1(H_{Train}, H_{test})$ distance value as pixel intensity value of our image in image domain.

$$G(x) = L_1(H_{Train}, H_{test}) \quad (3.4)$$

We apply this procedure to every pixel in the test image in the LBP domain. If the texture in the window is similar to the texture in training image, then this pixel will have a low value. If the texture in the window is not similar to the texture in training image, then this pixel will have a high value. For the textured image in Figure 3.3 the resulting image with L_1 distance intensity values is in Figure 3.7.

By using window based comparison in LBP domain we take into account the structural information about all the pixels in window. For this reason the L_1 distance value which we assign as a pixel value is the result of strong information about the structure of the texture. This is the property that makes our filter robust. However, while the windowing process obtains robustness to our approach; it also results in smoothing in forming the filtered images. Smoothness is a good effect for eluding the image from the textural structure but on the other hand it brings about boundary effect in our approach. The reason of boundary effect is as follows: during the windowing process the homogeneities in the windows of the pixels which are closer than 8 pixels to the boundaries are low. Because the window of such pixels are shared by two different textured regions (Figure 3.8). For this reason when we compare this window's LBP histogram with the LBP histogram of the training image by the L1 distance, the result will be higher compared to a pixel's window which is inside the textured region that we want to segment.

Yet, contribution of the windowing process to robustness is more important compared to the boundary effect. Also, the boundary effect can be minimized by several approaches. For instance, at the boundaries using a smaller window size would be a good approach for boundary effect. The size of the windows can be decided according to the homogeneities inside the windows at the boundaries. The homogeneity criterion can be formed by calculating the entropies inside the windows.

Using likelihood values directly (without using L_1 distance comparison) from the LBP histograms also could be another filtering approach. In particular, for a given pixel intensity value in LBP domain we calculate its likelihood from the training image's LBP histogram. High likelihood valued pixels are more likely to the training texture. Since we are not using a windowing procedure we won't have any boundary effect. On the other hand since we are only using one pixel's structural information this approach is not as robust as our approach. In addition, this approach works well for uniform textures but when we use nonuniform textures the results are not satisfactory. Suppose that we have the test image in Figure 3.9 which consists of two uniform textured regions in Figure 3.10(a) and 3.10(b).

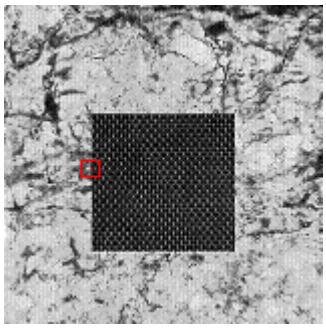
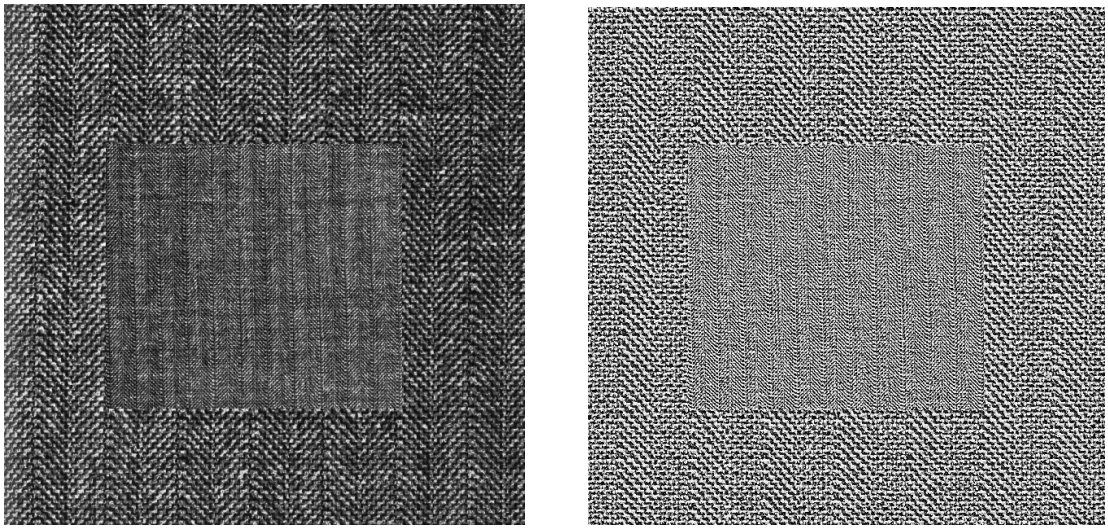


Figure 3.8: Window of a pixel at the boundary.



(a) (b)

Figure 3.9: (a) Test image consists of two uniform textured regions Figure 3.10(a)(b) **(b)**LBP representation of (a)

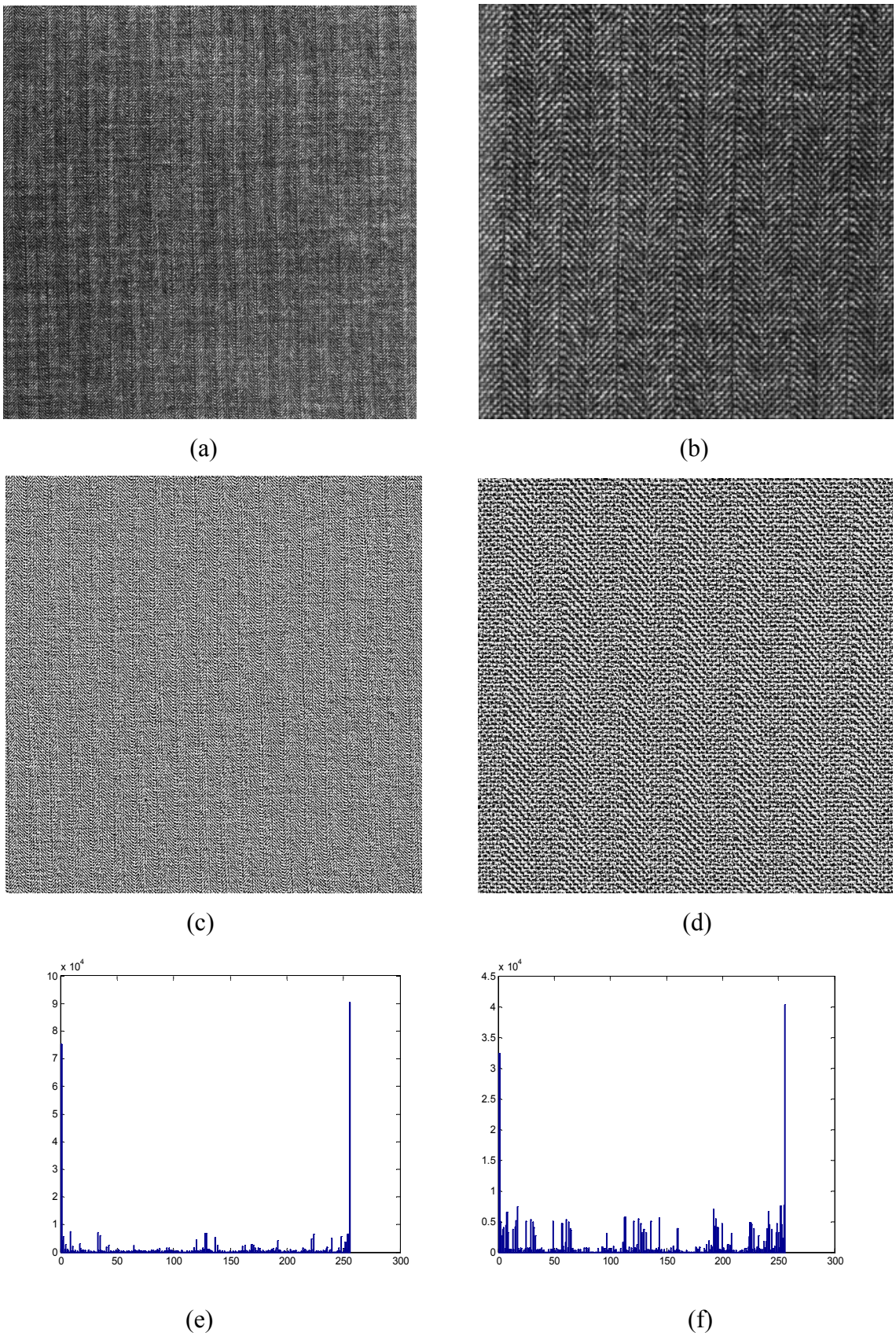


Figure 3.10: **(a)** Uniform texture of the square in the test image. **(b)** Uniform texture of the background in the test image. **(c)** LBP representation of texture (a). **(d)** LBP representation of texture (b) **(e)** LBP histogram of (a) **(f)** LBP histogram of (b)

To filter our test image in Figure 3.9(a) according to likelihood values, we calculate the likelihood values by using the LBP histograms in Figure 3.10(e) and (f) for every pixel in LBP representation of our test image in Figure 3.9(b). We assign the likelihood values to the pixel values to produce our filtered image. The likelihood based filtering results with respect to texture in Figure 3.10(a) and 3.10(b) are respectively in Figure 3.11(a) and Figure 3.11(b). When we look at the filtering results we can perceive two regions, but they are not robustly distinguished. Because the structural information is only depends on the each pixel for this reason the results are noisy. When we look at the filtering results of our approach in figure 3.12 we can clearly see the robustness of our approach compared to the likelihood approach. Also we can realize the robustness of our system when we look at the histograms of filtering results. When we use nonuniform textures the problem will become harder. Suppose that we have the test image with the textures in Figure 3.13. The texture at the background is a nonuniform and the texture inside the square is a uniform texture.

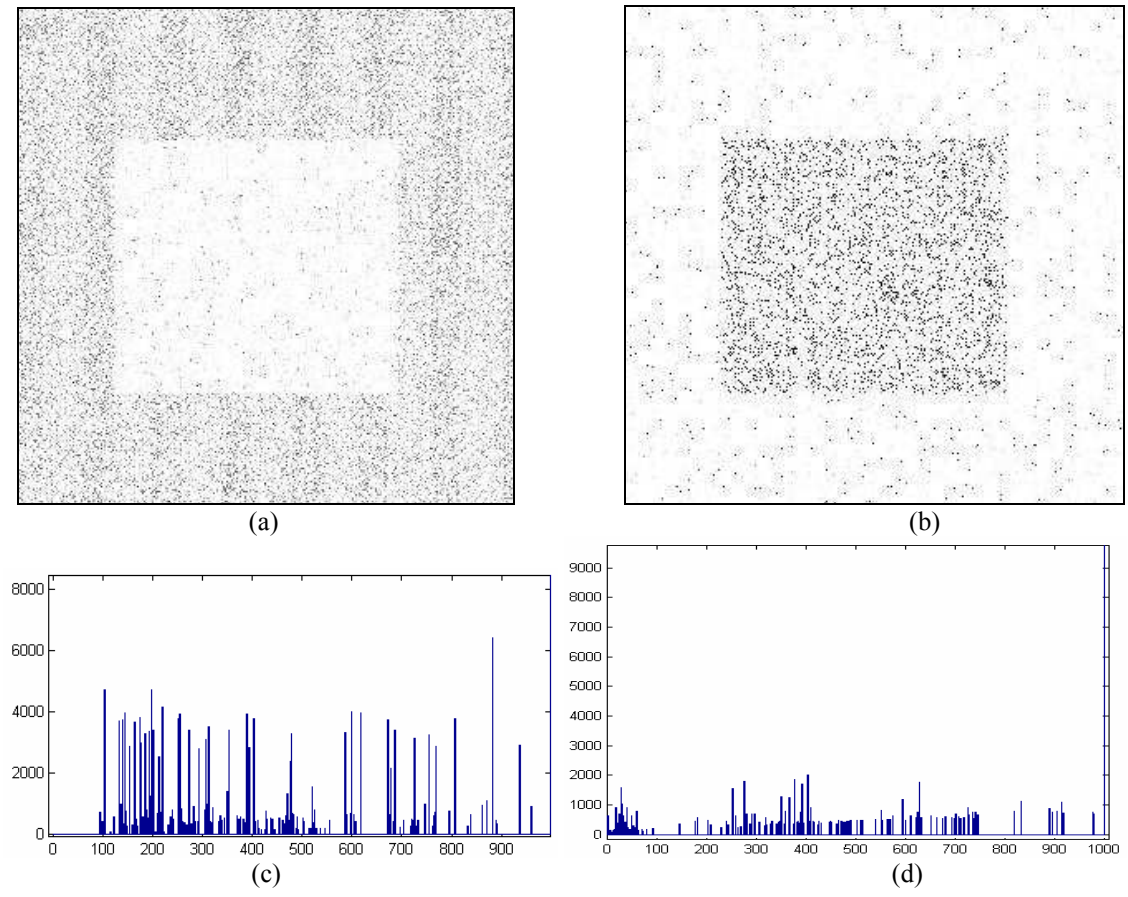


Figure 3.11: **(a)** The likelihood based filtering result with respect to texture in Figure 3.10(a) **(b)** The likelihood based filtering result with respect to texture in Figure 3.10(b). **(c)** Histogram of (a) **(d)** Histogram of (b)

The problem with the nonuniform textures is they include different textural structures. For this reason it is probable to observe pixels with similar textural structure with the uniform textures. As a result of this it is probable to observe noisy filtering results. When we look at the filtering results in Figure 3.14 we see that when we filter the image according to the nonuniform texture in Figure 3.13(b) both approaches produce bad results. Likelihood based filtering approach doesn't produce any region in the foreground (Figure 3.14(b)). Our approach produce an indistinct region in the foreground but it is not strong enough to reach a satisfactory segmentation result (Figure 3.14(d)). When we filter the image according to the uniform pattern in Figure 3.13(c) our approach produces more robust filtering results compared to likelihood based approach as shown in Figure 3.14(a),(c). In following sections we will compare the segmentation results of the likelihood based approach with segmentation results of our approach.

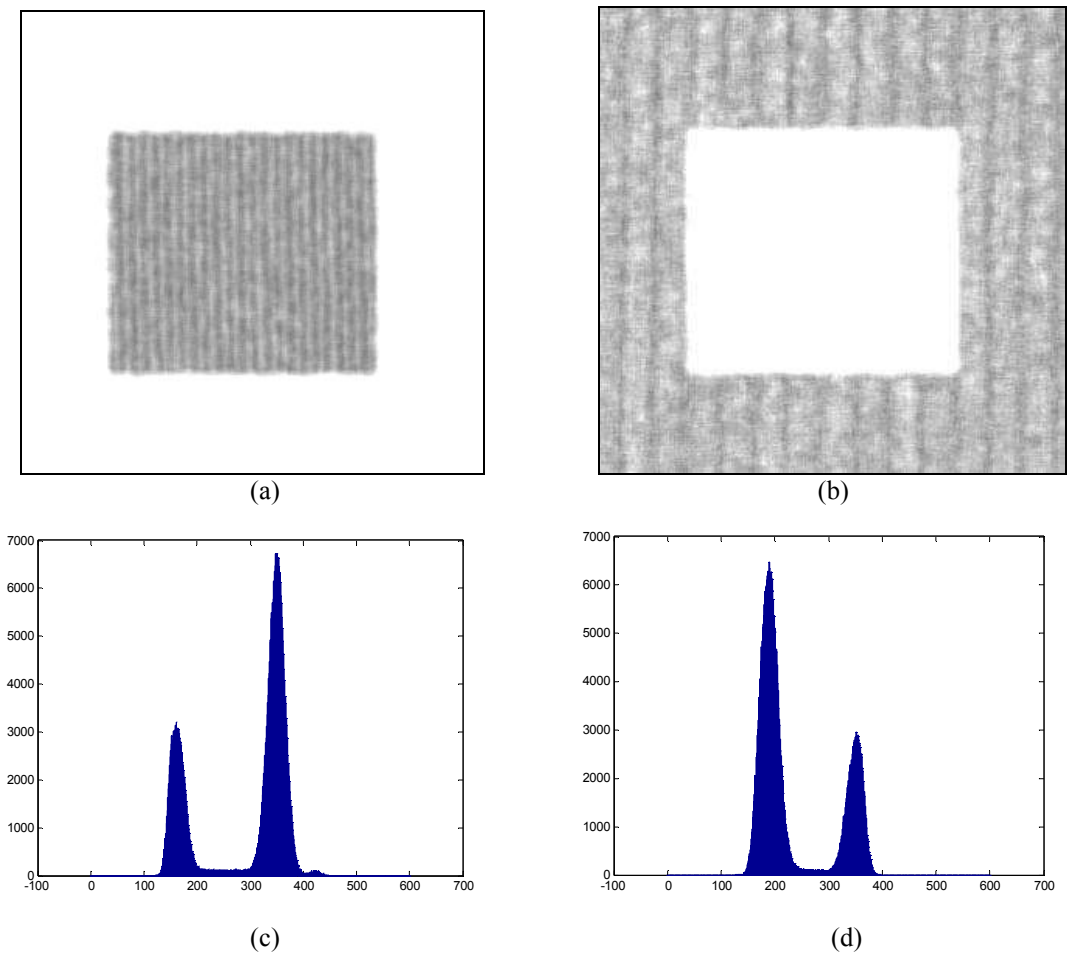
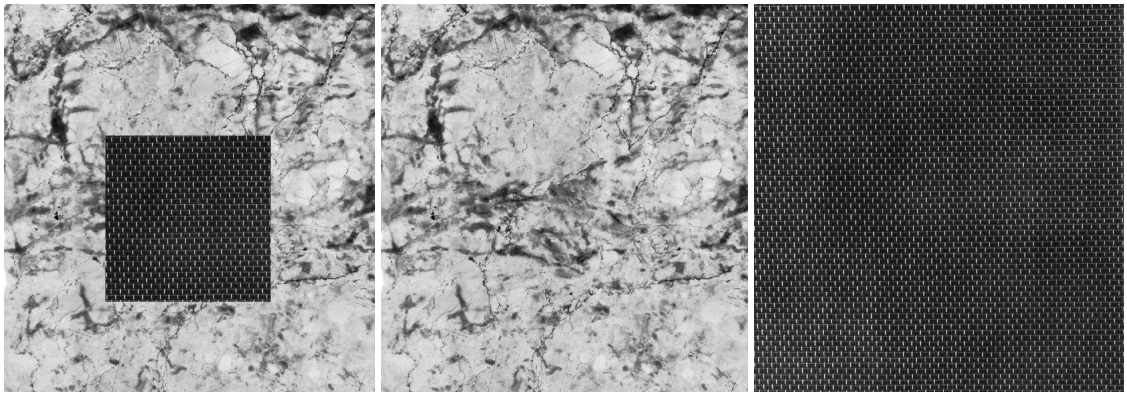


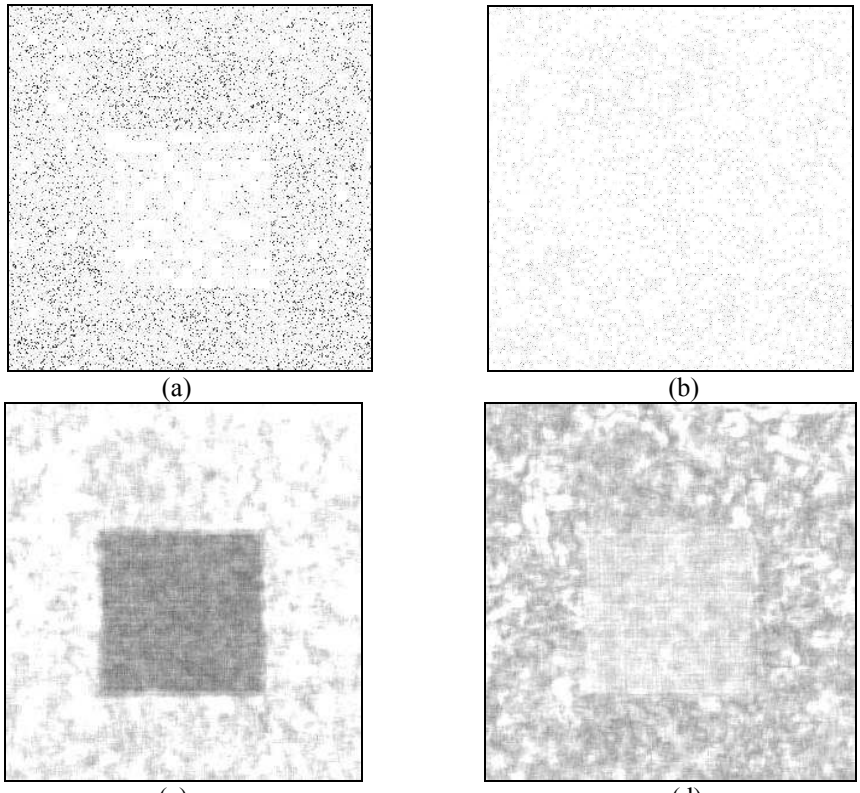
Figure 3.12: **(a)** Our filtering result with respect to texture in Figure 3.10(a) **(b)** Our filtering result with respect to texture in Figure 3.10(b). **(c)** Histogram of (a) **(d)** Histogram of (b)

Another approach can be using a vector valued texture analysis method like Gabor filter. Gabor filters are very powerful texture analysis tools. Compared to LBP, Gabor filters have computational complexity. Also when we applied Gabor filter to the image in figure 3.15(a) we observed that our filter produced more satisfactory results (figure 3.15(b), (c)).



(a) (b) (c)

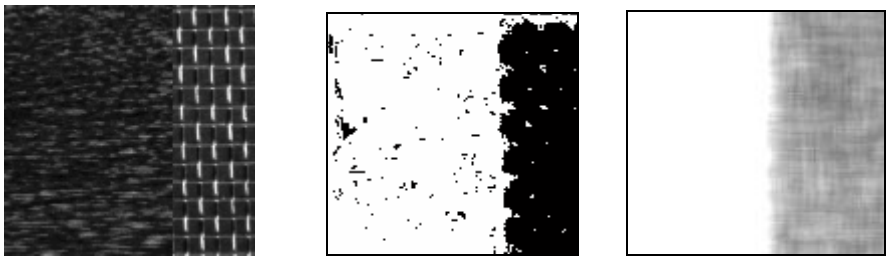
Figure 3.13: **(a)** Test image **(b)** Nonuniform texture. **(c)** Uniform texture



(a) (b)

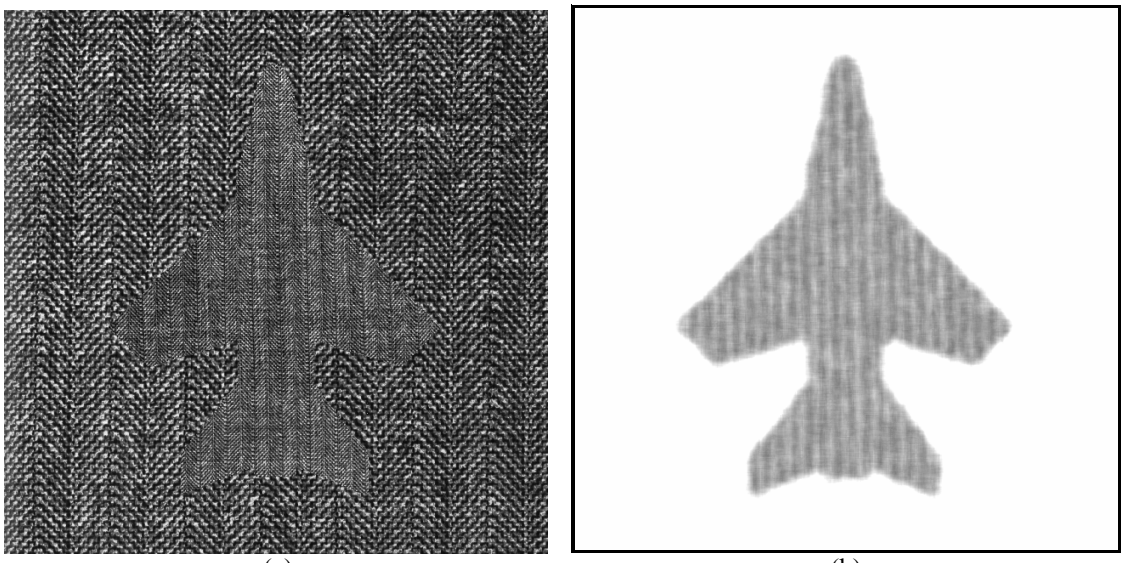
(c) (d)

Figure 3.14: **(a)** Likelihood based filtering result according to texture in Figure 3.13(c) **(b)** Likelihood based filtering result according to texture in Figure 3.13(b) **(c)** Our filtering result according to texture in Figure 3.13(c) **(d)** Our filtering result according to texture in Figure 3.13(b)



(a) (b) (c)

Figure 3.15: (a) Test image (b) Gabor filtering result (c) Our filtering result



(a) (b)

Figure 3.16: a) Textured Fighter Image b) Filtered Fighter Image

3.2) Data Term for Energy Functional:

After the filtering process we produce our data term in the filtered image. Let us continue with a more complicated shape. Suppose that we have the following textured jet fighter image in figure 3.16. After filtering process we reduced the texture segmentation problem to pixel intensity based image segmentation problem, because we have eluded the textured image from the structural properties of the textures. For image segmentation in our filtered image we use Kim, et al.[11] energy functional where they have combined mutual information based data term with shape prior term under a Bayesian framework. For data term they developed a nonparametric information-theoretic method which can deal with variety of intensity distributions [12]. They cast

the segmentation problem as maximization of mutual information between region labels and image pixel intensities [12]. Their energy functional is as follows:

$$E(\underset{\sim}{C}) = -|\Omega|\hat{I}(G(X); L_{\underset{\sim}{C}}(X)) + \alpha \oint_{\underset{\sim}{C}} ds \qquad (3.5)$$

Where $\hat{I}(.)$ is the estimated mutual information between intensity values ($G(X)$) and binary label ($L_{\underset{\sim}{C}}(X)$). $|\Omega|$ is the area of the image domain and $\oint_{\underset{\sim}{C}} ds$ is the length of the curve $\underset{\sim}{C}$ which is a regularization term used as curve length penalty. α is a scalar constant and X is the pixel index. We have discussed the derivation of mutual information based data term in Chapter 2 for details we refer reader to Chapter 2.1 and [11,12]. Using level set based active contours by using energy equation in equation 3.5 the image segmentation results are very successful. Even without our texture filtering process in some textured images, which include different regions with different first order pdfs, Kim, et al.'s method gives satisfactory segmentation results on the boundaries of shapes as shown in Figure 3.17. On the other hand when we apply this method without our texture filtering process to Figure 3.16 a), which has textured regions with similar first order pdfs the segmentation result is in Figure 3.18.

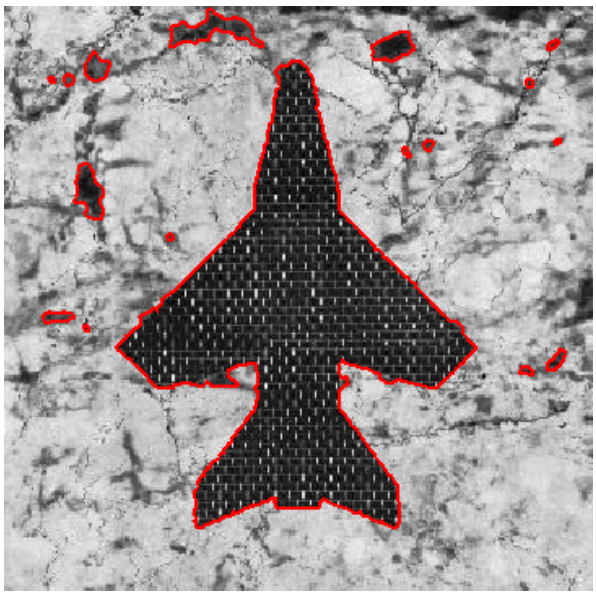
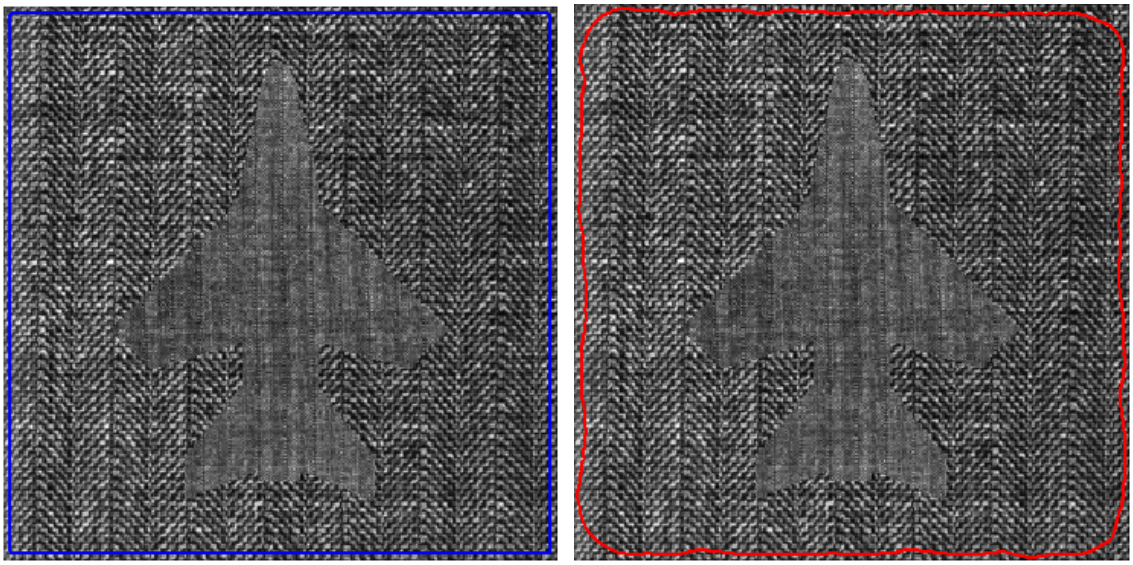


Figure 3.17: Segmentation result of Kim, et al.'s method at textures with different first order pdf's



(a) (b)

Figure 3.18: a)Initial position of the curve b) Segmentation result of Kim et al.'s method at textures with close first order pdf's without our texture filtering process

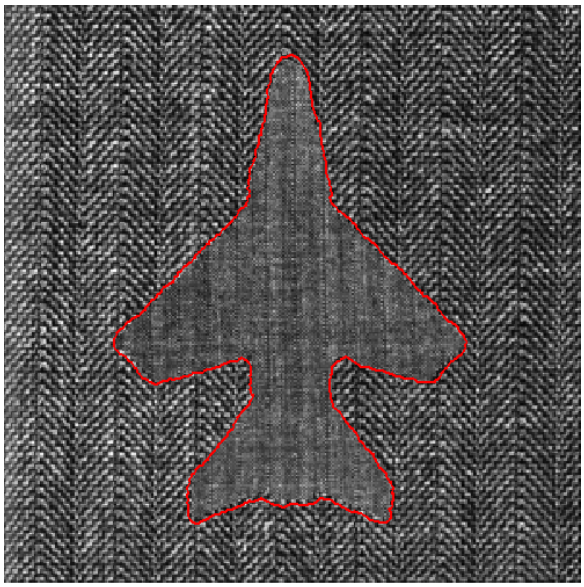


Figure 3.19: Our segmentation result by using our LBP based texture filter

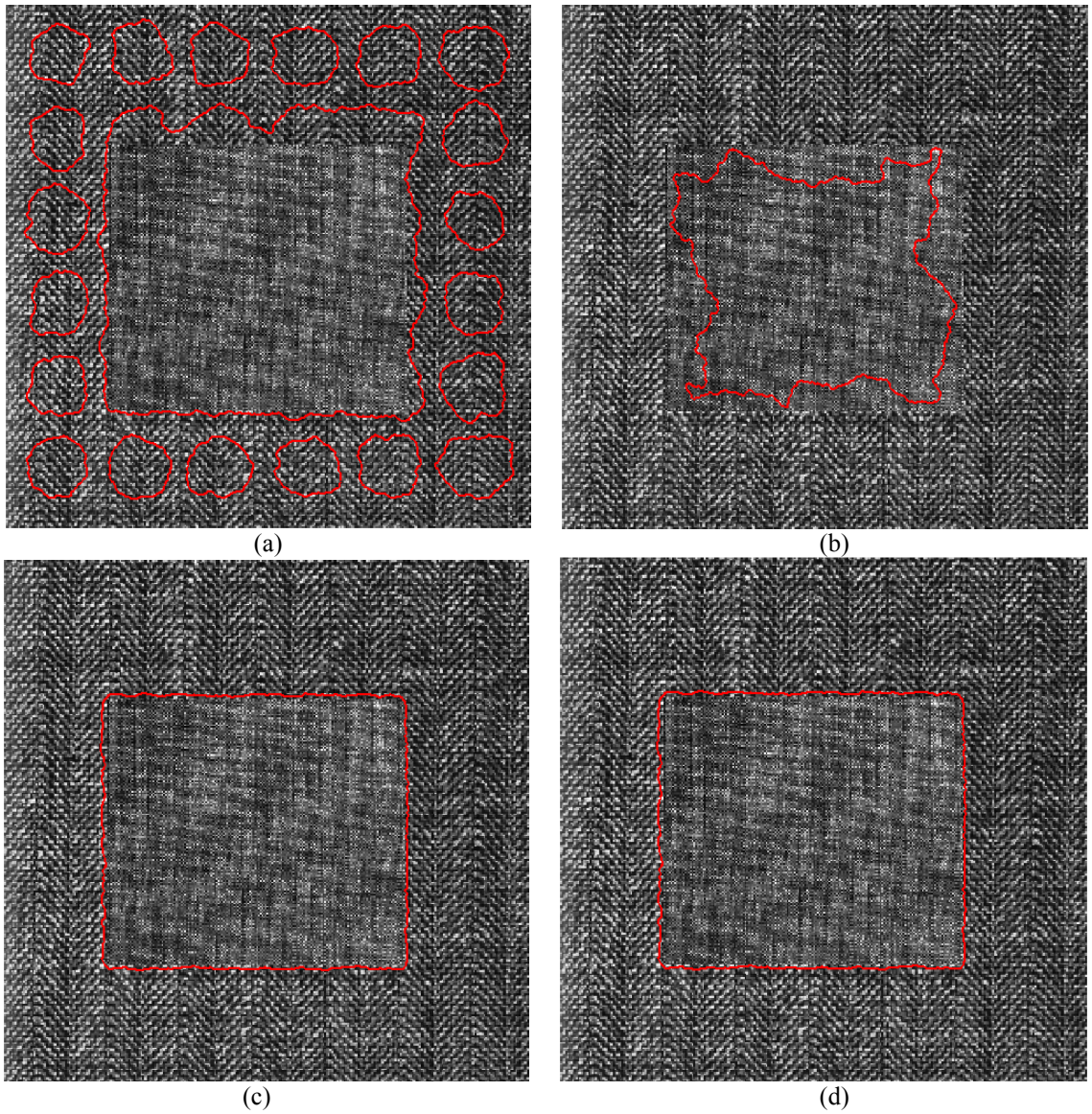
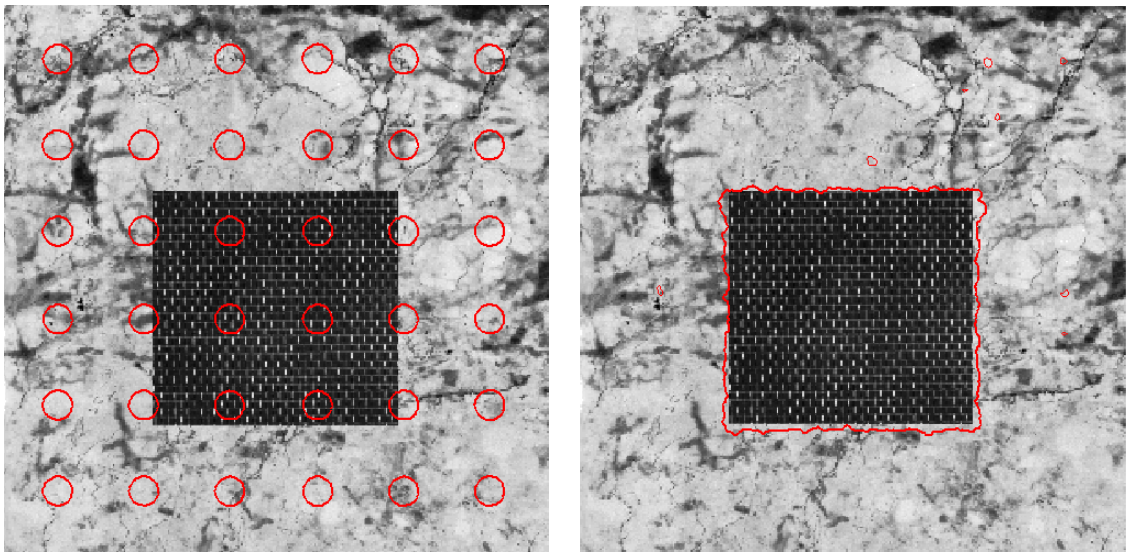


Figure 3.20: **(a)** Curve evolution based segmentation result with respect to figure 3.10(a) **(b)**Curve evolution based segmentation result with respect to 3.10(b) **(c)** Curve evolution based segmentation result with respect to figure 3.10(a) by our approach **(d)**Curve evolution based segmentation result with respect to figure 3.10 (b) by our approach



(a) (b)

Figure 3.21: (a) Segmentation result by using likelihood based filtering according to texture in figure 3.13(c) **(b)** Our segmentation result according to texture in Figure 3.13(c)

When we apply our filtering process before using Kim, et al.'s energy equation for the image in Figure 3.16 our segmentation result is as shown in figure 3.19. As we have mentioned in section 3.1 we present the segmentation results using likelihood based filtering approach in Figure 3.20(a)(b) and our segmentation results with the same image in figure 3.20(c)(d). When we look at the segmentation results we can see the robustness of our approach when we filter the test image with respect to both foreground and background textures. When we use the likelihood based filtering with respect to texture in figure 3.10(a) we cannot reach to satisfactory segmentation results in figure 3.20(b), also when we filter the image with respect to texture in figure 3.10(b) the segmentation result in figure 3.20(a) is not satisfactory either. When we look at the segmentation results in figure 3.21 for the test image in figure 3.13 (a) with nonuniform texture in the background the segmentation results with likelihood based approach in figure 3.21(a) is not satisfactory either. Our segmentation result in figure 3.21(b) for test image in figure 3.13(a) is satisfactory because of the robust filtering result.

After our filtering process using Kim et al.'s mutual information based data term is not the only choice we can use. Because, the output images of our filter mostly consist of two easily separable regions by intensity distributions. For this reason using other data terms can be suitable for our approach. For instance separation of means or Chen and Vese's [3] approach can be suitable for our approach. When our texture filter produces noisy results (Figure 3.14(c)) at difficult textures using separation of means or other approaches does not give satisfactory results. For this reason we choose mutual

information based data term. In experimental results section we discuss the cases when separation of means based data term can be suitable.

3.3) Shape Priors:

Shape prior information can aid the segmentation problems with missing data (e.g.: Occlusion problems) or low quality images. Also in texture segmentation when the texture analysis method is not sufficient to represent the textured regions clearly separated or produce too noisy results, prior shape information about regions to be segmented would be very useful additional feature to reach satisfactory segmentation results. Kim, et al. used an energy functional where their nonparametric shape prior and their data term combined under a Bayesian framework. They have considered the regularization term in equation (3.5) as a prior term which is a negative logarithm of a prior density. The idea behind this is shorter curves are more likely than longer ones.

$$E(C) = -\log p(data \mid C) - \log p_C(C) \tag{3.10}$$

As we have mentioned before we use their shape and data driven image segmentation approach [11,13]. Data term in equation (3.10) is the first term in equation (3.5) which is based on mutual information. For the shape term they propose a nonparametric shape prior approach. They produce a shape prior distribution such that given an arbitrary shape; likelihood of this shape can be evaluated. They estimate the underlying shape distribution by using Parzen density estimator in space of shapes. Density estimates are expressed in term of distances between shapes. In section 2.5.1 we mentioned the details of Parzen density estimation. So, by using our background about Parzen density estimation we can present Kim, et al.'s Parzen density estimate in shape space as follows:

$$\hat{p}_{\tilde{C}}(\tilde{C}) = \frac{1}{n}\sum_{i=1}^{n} k(d_C(\tilde{C}, \tilde{C}_i), \sigma) \tag{3.11}$$

$$\hat{p}_{\tilde{\phi}}(\tilde{\phi}) = \frac{1}{n}\sum_{i=1}^{n} k(d_D(\tilde{\phi}, \tilde{\phi}_i), \sigma) \tag{3.12}$$

Equation (3.11) is the Parzen density estimate in space of curves and equation (3.12) is the Parzen density estimate in space of signed distance functions. n is the number of training shapes. $\widetilde{C}$ is the aligned candidate curve and $\widetilde{C}_i$ are the aligned training curves in the curve space, $\widetilde{\phi}$ is the aligned candidate curve and $\widetilde{\phi}_i$ are the aligned training curves in signed distance functions space. σ is the Gaussian kernel size. d_C and d_D are respectively distance metrics in space of curves and space of signed distance functions. They have used two distance metrics which are template metric and L_2 distance between signed distance functions. Template metric can be expressed as L_1 distance between two binary maps I_1 and I_2 whose values are 1 inside the shape and 0 outside the shape, namely:

$$d_T(\widetilde{C},\widetilde{C}_i) = \int_\Omega \left\| I_1(x) - I_2(x) \right\| dx \tag{3.13}$$

and L_2 distance is a norm of difference between two signed distance functions which is as follows:

$$d_{L_2}(\widetilde{\phi},\widetilde{\phi}_i) = \sqrt{\left\langle \widetilde{\phi} - \widetilde{\phi}_i, \widetilde{\phi} - \widetilde{\phi}_i \right\rangle} \tag{3.14}$$

where inner product is defined as:

$$\left\langle \widetilde{\phi},\widetilde{\phi}_i \right\rangle = \frac{1}{|\Omega|}\int_\Omega \widetilde{\phi}(x)\widetilde{\phi}_i(x)dx$$

By using these nonparametric shape prior terms (Equations (3.11) and (3.12)) we combine the mutual information based data term (Equation (3.5)) which is produced from the output of our texture filter under Bayesian framework in Equation (3.10) for shape based texture segmentation.

3.4) Gradient Flows for Data and Shape Term for Segmentation:

For segmentation we have to derive a curve evolution formula to evolve our active contour to fit to the boundary of the shape by minimizing our energy functional. For this process we have to minimize our shape and data terms in our energy functional. We implement the curve evolution for the gradient flow using level set method with narrow band approach. For details of narrow band approach see [11].

The gradient flow equation for data term proposed by Kim, et al. [11,12] is as follows:

$$\frac{\partial \vec{C}}{\partial t} = \left[\log \frac{\hat{p}_+(G(\vec{C}))}{\hat{p}_-(G(\vec{C}))} + \frac{1}{|R_+|} \int_{R_+} \frac{K(G(x) - G(\vec{C}))}{\hat{p}_+(G(x))} dx - \frac{1}{|R_-|} \int_{R_-} \frac{K(G(x) - G(\vec{C}))}{\hat{p}_-(G(x))} dx \right] \vec{N} \quad (3.15)$$

And the gradient flow equation for the shape term [11,13] is as follows:

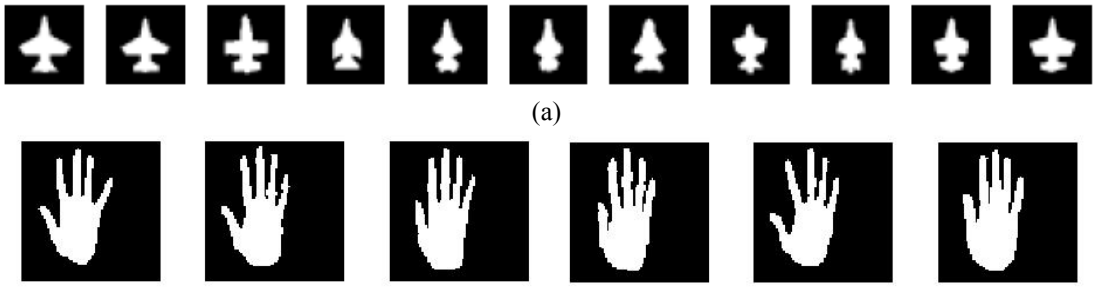
$$\frac{\partial \widetilde{C}}{\partial t} = \frac{1}{p_{\widetilde{C}}(\widetilde{C})} \frac{1}{\sigma^2} \frac{1}{n} \sum_i k(d_T(\widetilde{C}, \widetilde{C}_i), \sigma) d_T(\widetilde{C}, \widetilde{C}_i)(1 - 2H(\phi_i)) \vec{N} \quad (3.16)$$

For the details of derivations of equations (3.15) and (3.16) see [11,12,13]. Note that equation 3.16 is in terms of the aligned curve. In the actual curve evolution process, each iteration involves the flow in equation 3.16 as well as the alignment of the evolving curve to the training shapes using the alignment algorithm of [31]. For minimizing our energy functional we are using gradient descent algorithm. Note that although we would ideally want to find the global minimum of the cost function in equation 3.10, due to the difficulty of the optimization task, we use a local minimization scheme, namely gradient descent. Hence the best we can expect to reach is a local minimum of the cost function. However in many practical cases we find such local minima to be good solutions.

Since we perform local optimization, our algorithm can have some sensitivity to the initial position of the evolving curve. However we have not observed convergence to drastically different solutions when we start from a different initial curve. Of course initializing with a curve that is in some sense close to the actual boundaries would speed up the convergence and provide robustness in challenging cases.

3.5) Experimental Results:

In this section we present our experimental results. In our experiments we have used natural and synthetic textured images for different shapes. We have chosen the synthetic textured images from Brodatz database and we have made our experiments with hand and fighter shapes. We have tried to choose the test images to point the cases when other features are not able to discriminate regions. We have used eleven jet fighter and six hand shapes for training in our experiments. Our aligned training shapes for jet fighter and hand are shown in Figure 3.22. We align our shapes by using the technique in [31]. In subsection 3.5.1. we show our experimental results with synthetic images. In subsection 3.5.2. we present our results at occlusion problems. In subsection 3.5.3. we present our results at natural textured images.



(a)

(b)

Figure 3.22: Aligned shapes for our nonparametric shape prior space

3.5.1.) Experimental Results with Synthetic Images:

-Segmentation Results for Uniform Textures:

In this subsection we present our segmentation results at uniform textured images. In some of the results we have chosen textures with close first order pdfs to point the problems which can not be solved by using other features.

In figure 3.23 we have two textured regions which are taken from a jumper and from a doormat. These textures have low contrast between each other. Their first order pdfs are close. So, it is difficult to distinguish these regions by using other features than texture. By using texture as a feature with shape prior information about the hand shaped region we have reached to satisfactory segmentation result at Figure 3.23(d).

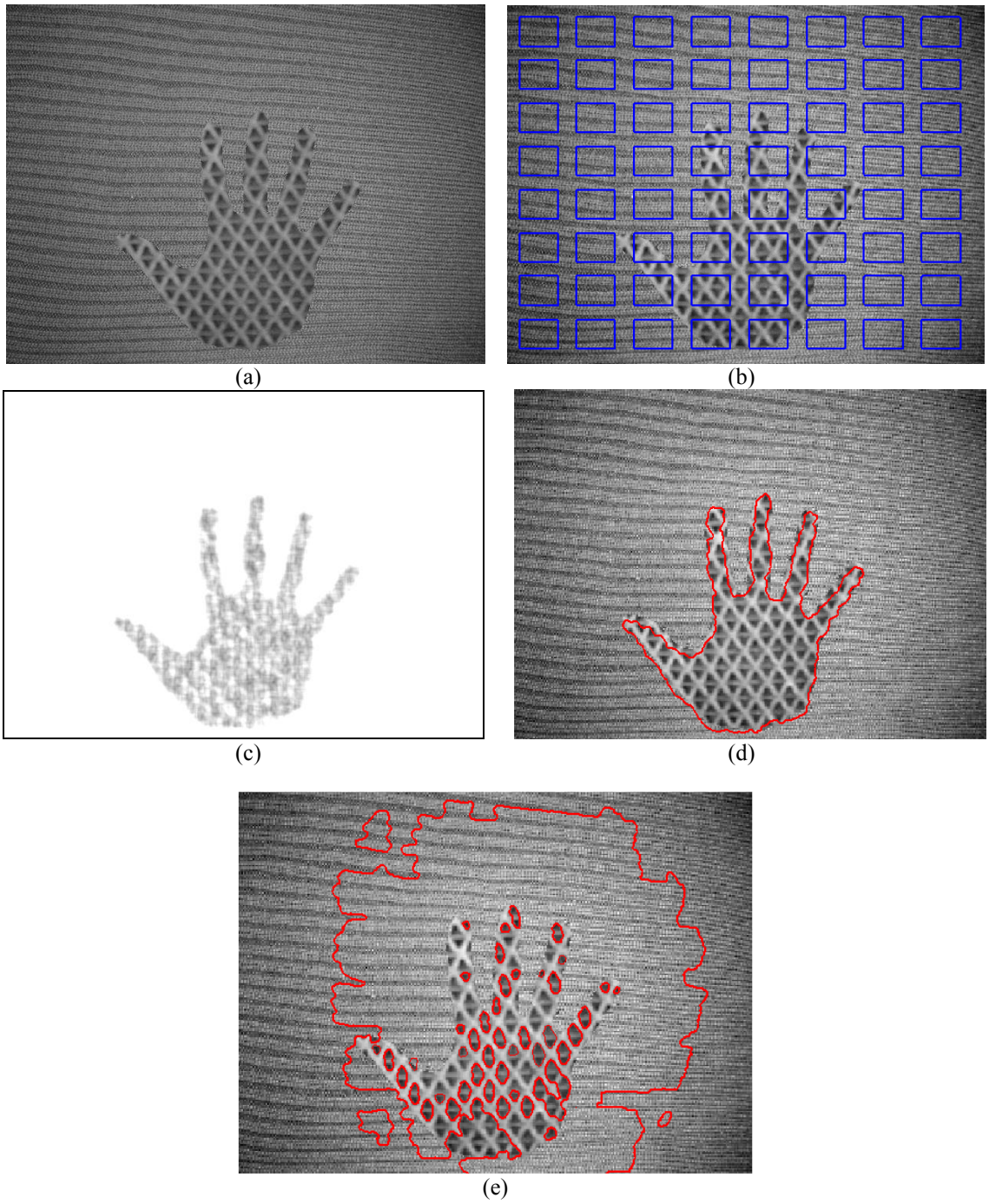
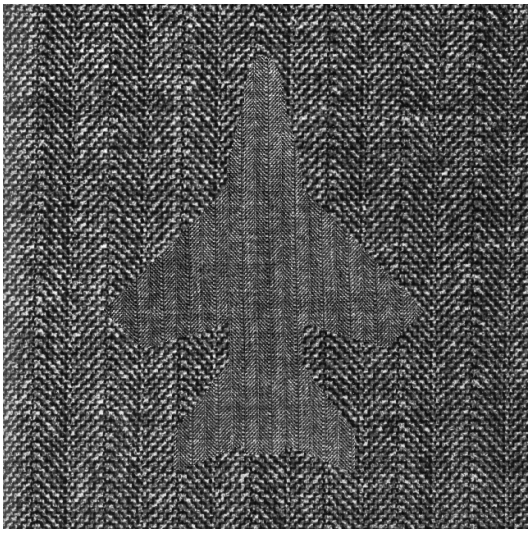
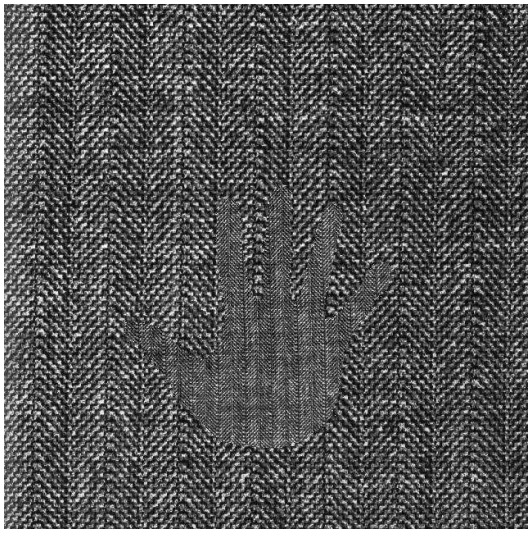


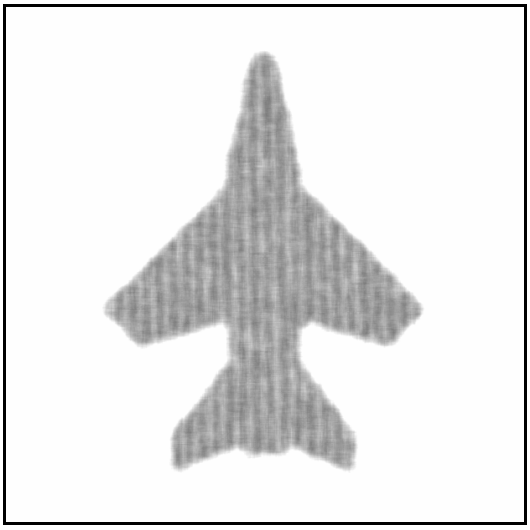
Figure 3.23 : (a) Test image **(b)**Initial curve **(c)** Filtered image **(d)** Segmentation result **(e)** Kim's segmentation result without our filtering process



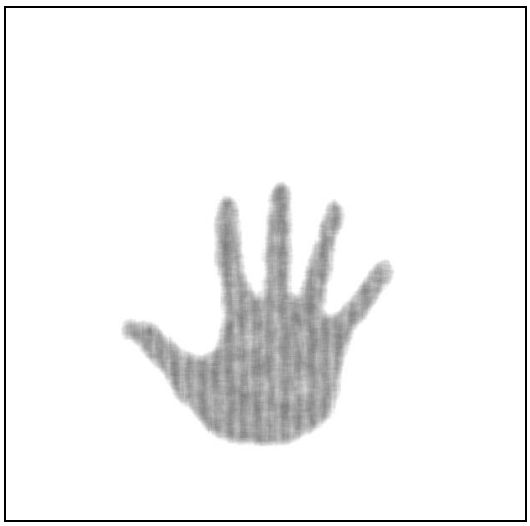
(a)



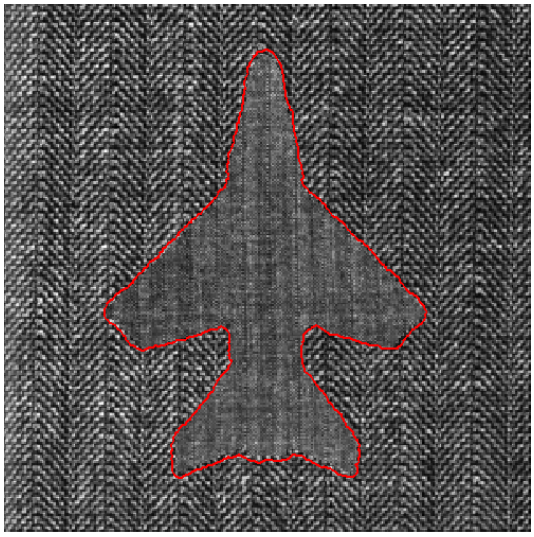
(b)



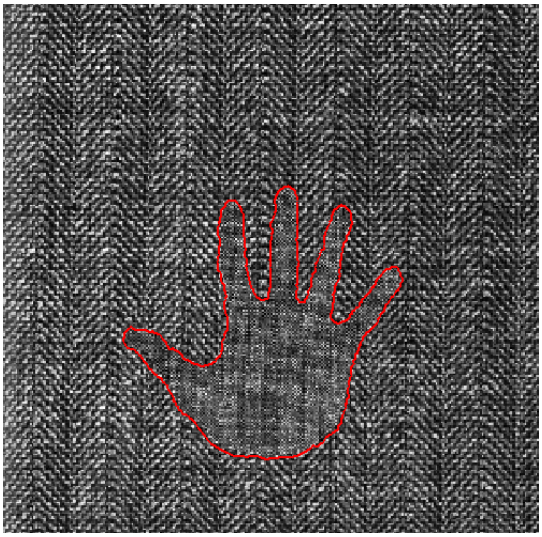
(c)



(d)



(e)



(f)

Figure 3.24 **(a)** Textured image with fighter shaped region **(b)** Textured image with hand shaped region **(c)** Filtering result of (a) **(d)** Filtering result of (b) **(e)** Segmentation result of (a) **(f)** Filtering result of (b)

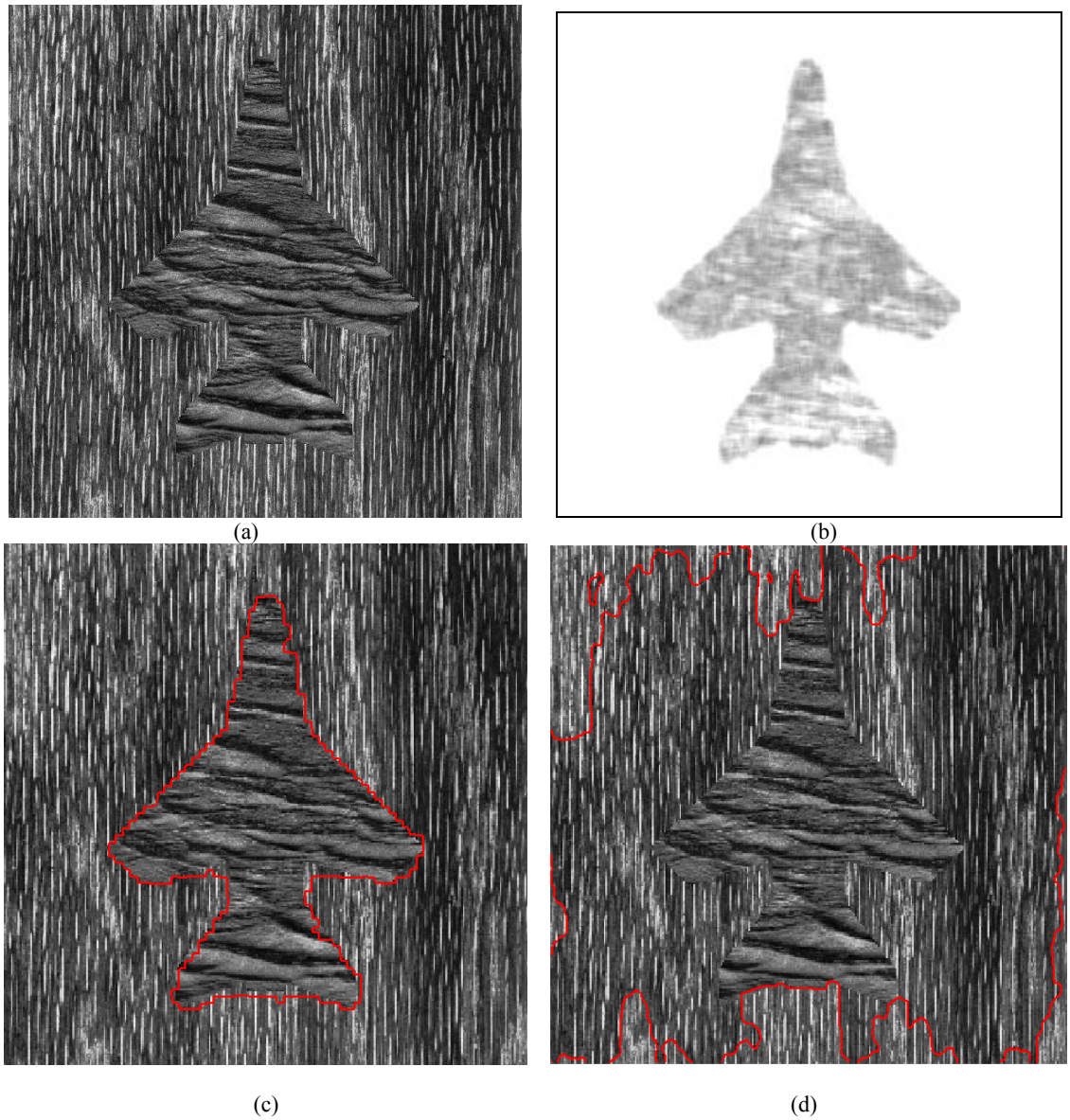


Figure 3.25: **(a)** Textured image with fighter shaped region **(b)** Filtering result of (a) **(c)** Segmentation result of (a) **(d)** Kim's segmentation result without our filtering process

Textures in Figures 3.24 and Figure 3.25 are also difficult cases for intensity based segmentation methods. The regions are not separable by using intensity distributions. Our texture filter provides output images with high contrast between regions which can be distinguished clearly. Then by the information provided from our mutual information based data term and shape prior term we can reach good segmentation results as shown in figures 3.25(c), 3.24(e), and 3.24(f). Since the filtering outputs of these images consist of two clearly separable regions we have reached very close results by using separation of means based data term. When the outputs of the filter become noisy mutual information based data term provide more successful results. We will discuss such examples in the following examples. Figure 3.25(d) represents the segmentation

result using Kim's energy functional (equation 3.5) without our texture filtering process. Since the pixel intensity distributions and first order pdfs of regions are similar, without using our texture filter Kim's approach produces unsatisfactory results.

In Figure 3.26(a) we provide another textured region. The difference with this example is, segmentation problem in this image can be handled by intensity distribution based approaches. There is high contrast between two regions. The reason we have used this image is the texture structures of the regions are similar. For this reason these kinds of textures cause our filter to produce noisy results as in Figure 3.26(b). Since our method is not totally working depending on the filtering process we can reach good segmentation results as shown in Figure 3.26(c).

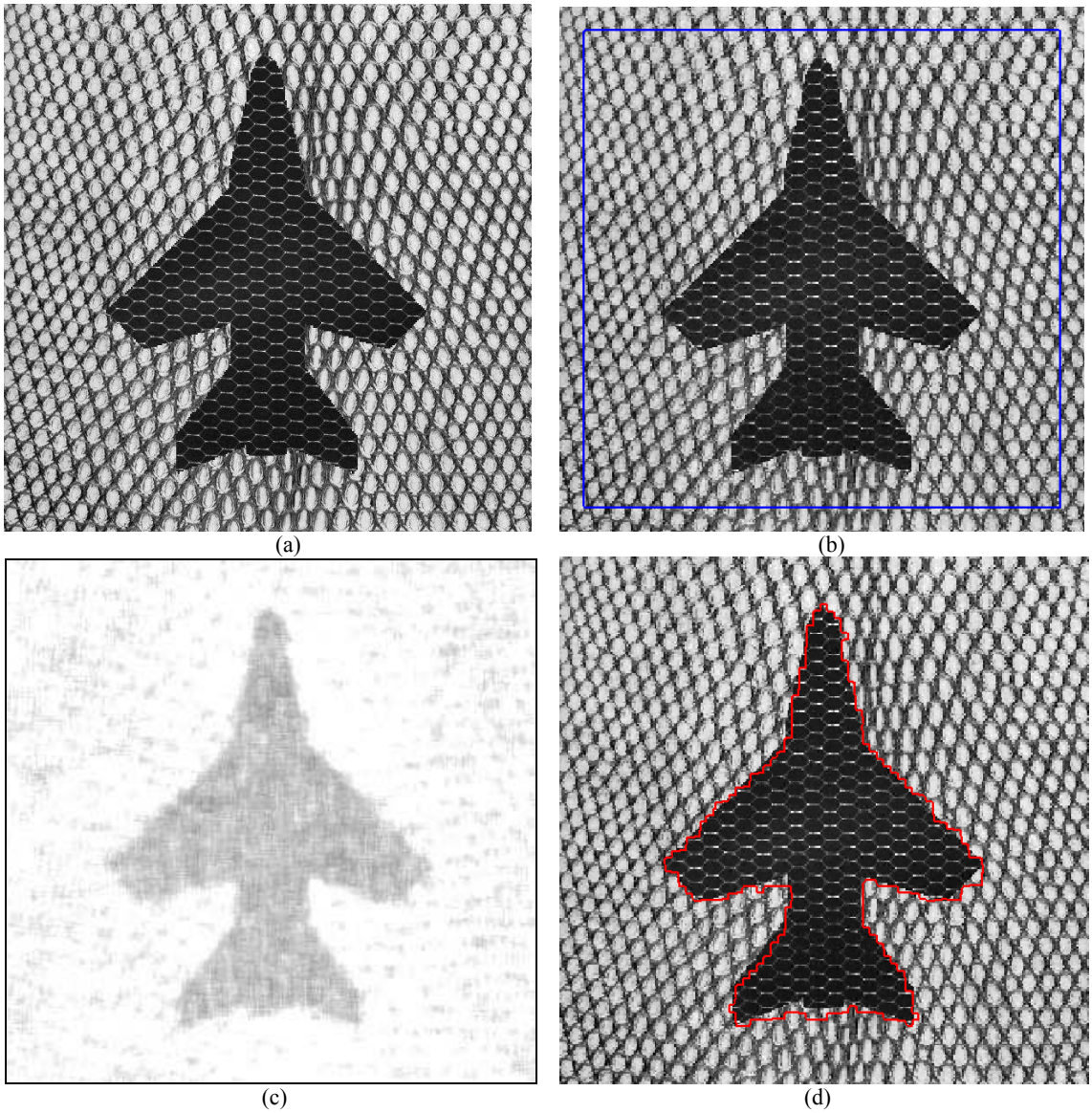
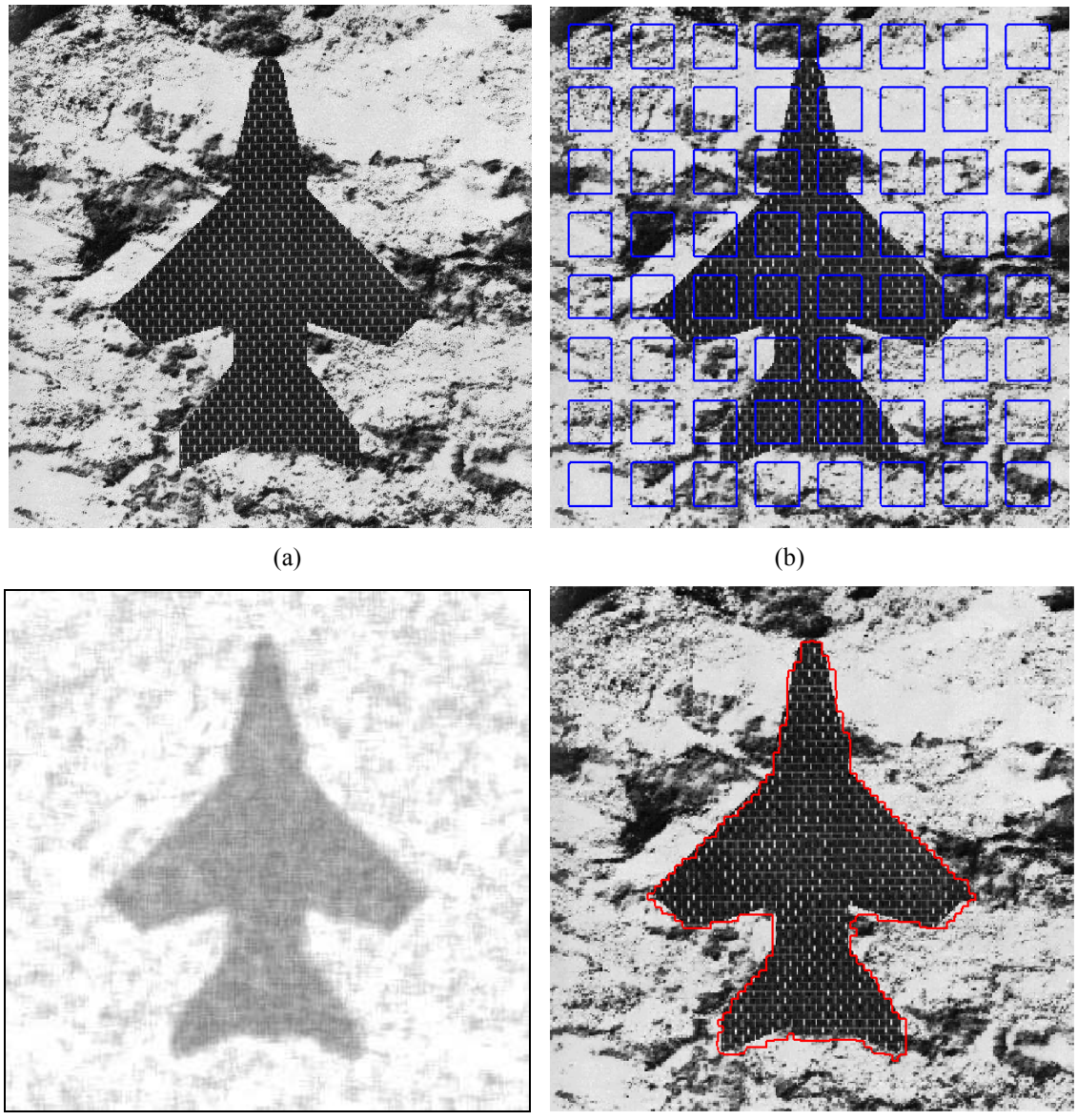


Figure 3.26: **(a)** Textured image with fighter shaped region **(b)**Initial curve **(c)** Filtering result of (a) **(d)** Segmentation result of (a)

-Segmentation Results for Nonuniform Textures:

In this subsection we present our segmentation results on nonuniform textured images. For nonuniform textures we have tried to choose test images which are structurally difficult to analyze. The difficulty with nonuniform textures is, it is probable that nonuniform textures include similar structures with other textures. For this reason texture analysis tools have some problems to clearly identify and discriminate nonuniform textures from other textures. Our texture filter produces some noisy results with nonuniform textures but we compensate this fact by our mutual information based data term and shape prior information in our energy equation.

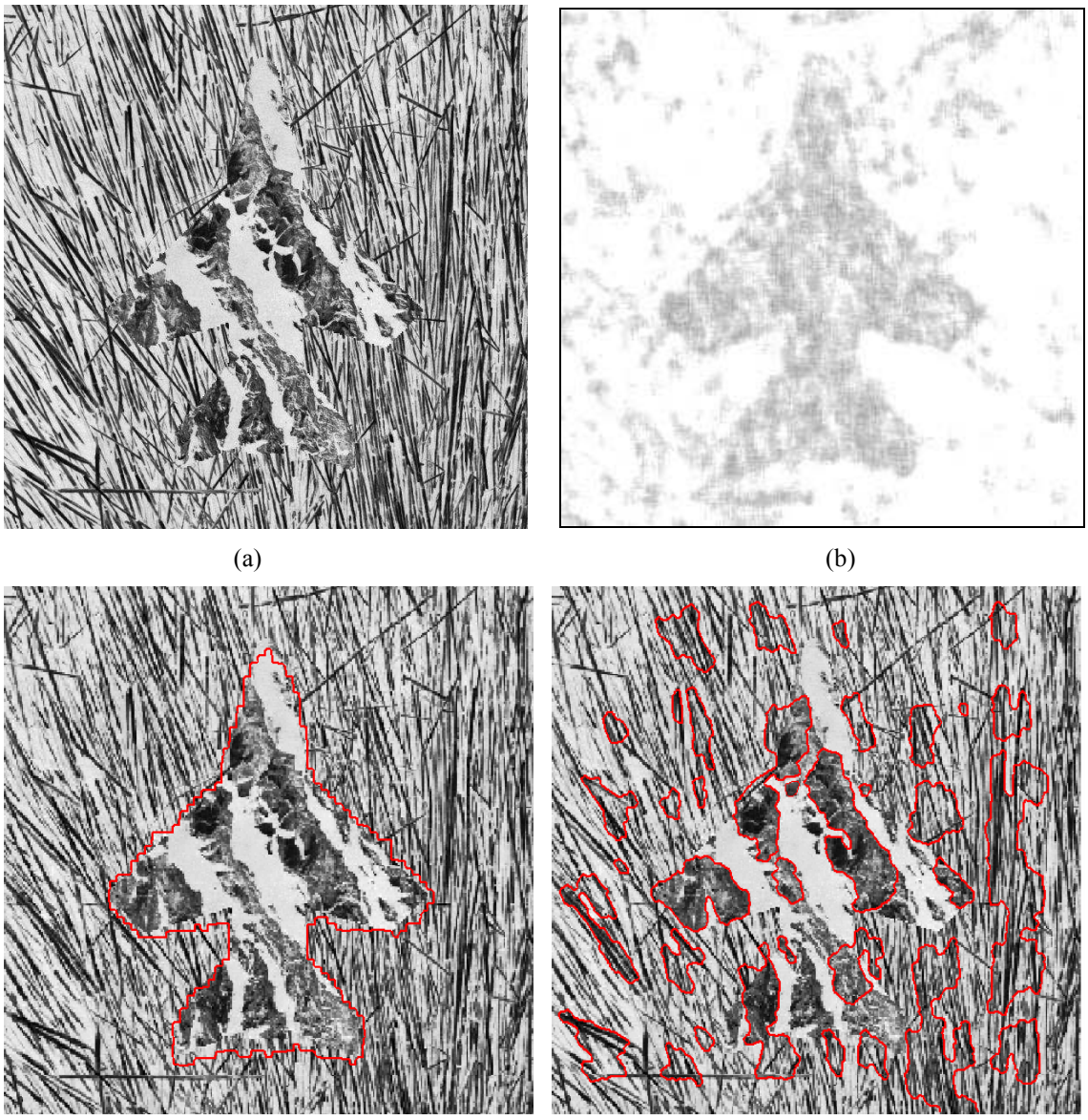


(a) (b)

(c) (d)

Figure 3.27: (a) Test image with nonuniform textured background **(b)**Initial curve **(c)** Filtering result **(d)** Segmentation result

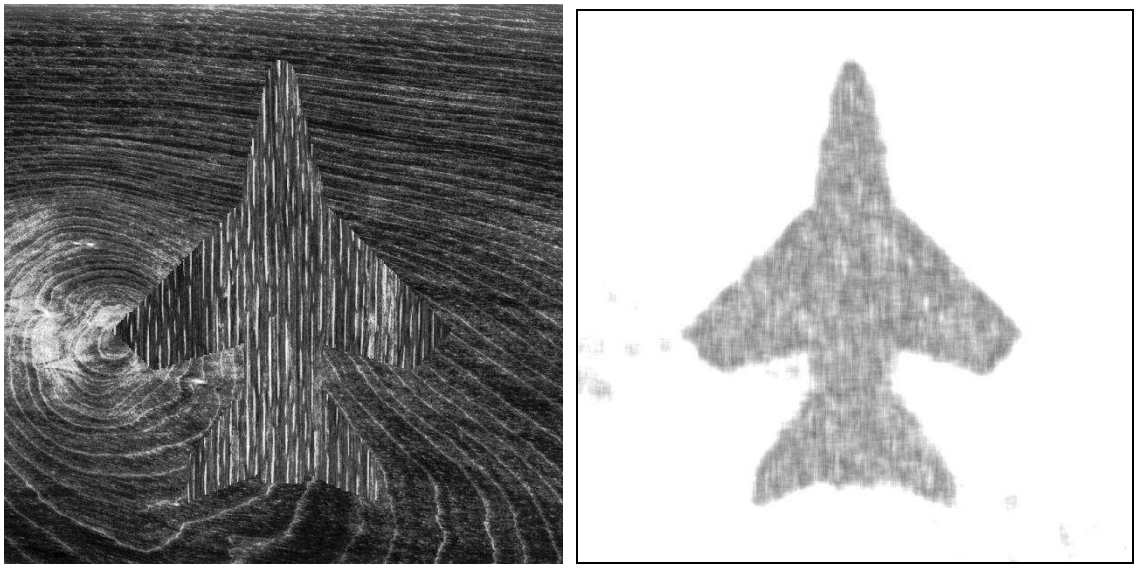
By our approach we have reached to good segmentation results with the nonuniform textures in Figures 3.27, 3.28, 3.29,3.30. In Figures 3.28 and 3.29 we have used nonuniform textures with close first order pdfs. in both foreground and background which makes the problem harder for both pixel intensity distribution based approaches and our approach.



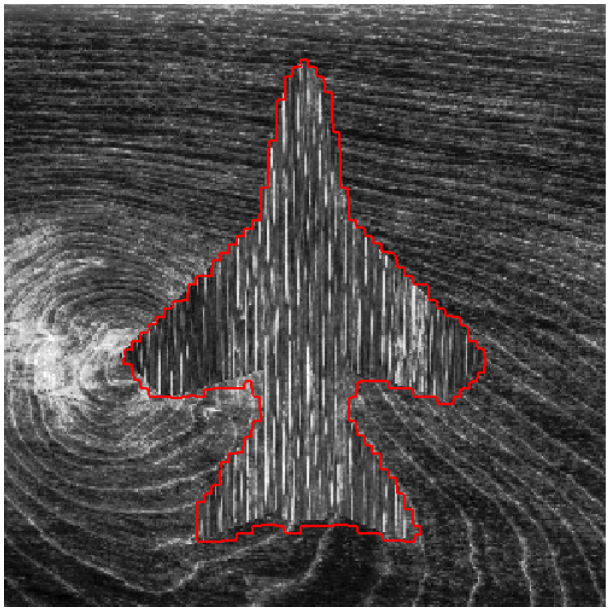
(a) (b)

(c) (d)

Figure 3.28: (a) Test image with nonuniform textured background and foreground **(b)** Filtering result **(c)** Segmentation result **(d)** Kim's segmentation result without our texture filtering process



(a) (b)



(c)

Figure 3.29: (a) Test image with nonuniform textured background and foreground **(b)** Filtering result **(c)** Segmentation result

In Figure 3.30 we have used two nonuniform textures. The textures are structurally very similar. When we look at the test image it is difficult to perceive and distinguish textured regions. Since the textures are nonuniform and similar we have noisy filtering results, but our algorithm is still be able to estimate the region of the shape according to the data and the shape prior information. Even we have too noisy filtering result our algorithm produces satisfactory segmentation result at this difficult problem.

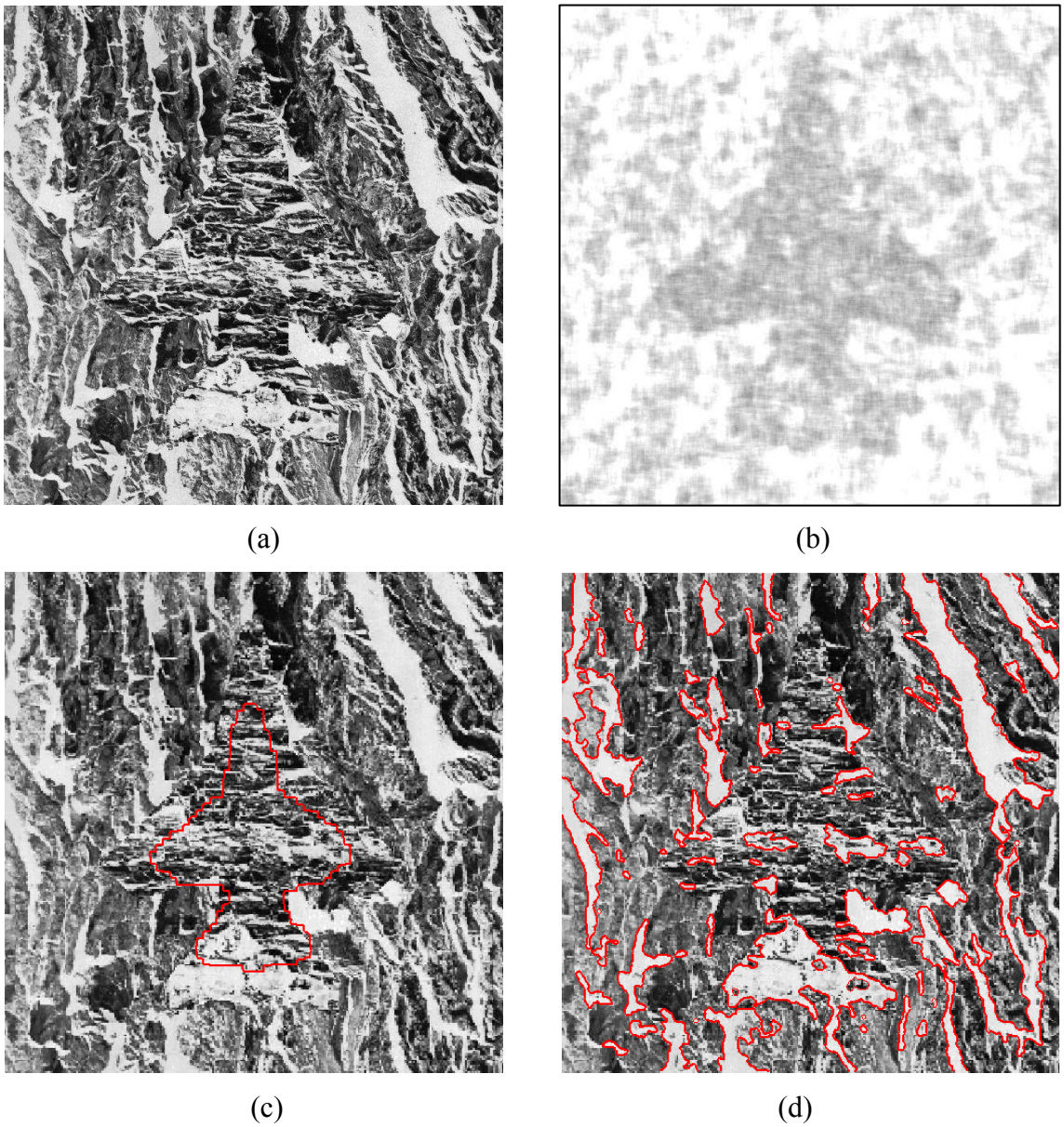
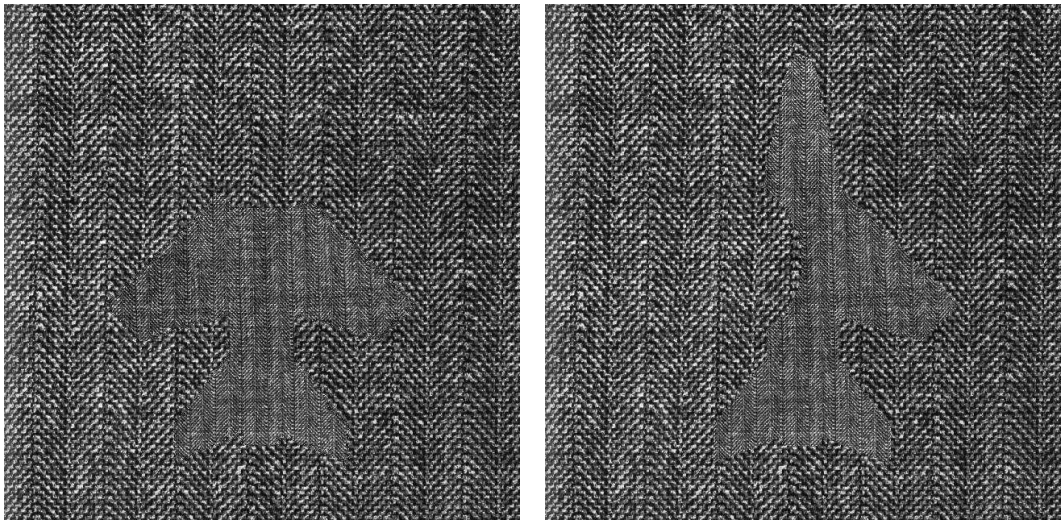


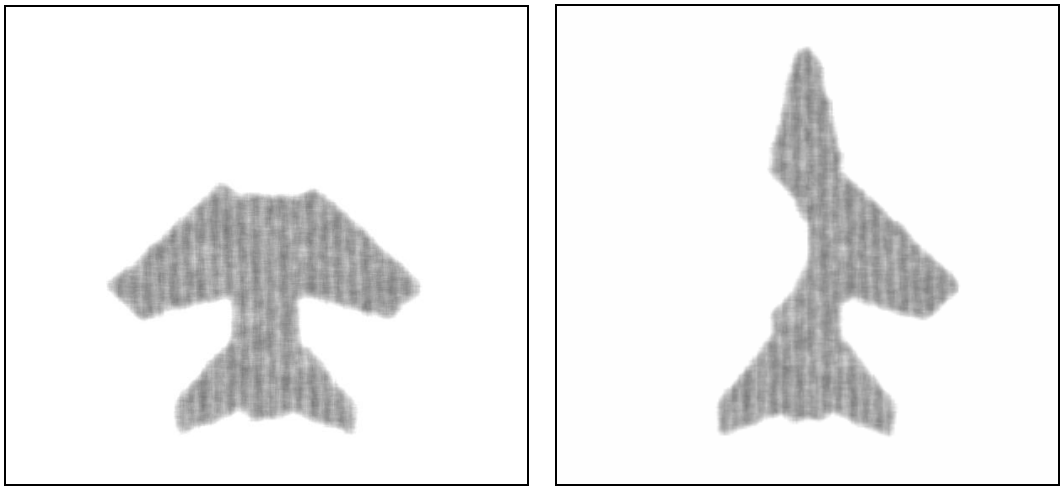
Figure 3.30: (a) Test image with nonuniform textured background and foreground **(b)** Filtering result **(c)** Segmentation result **(d)** Kim's segmentation result without our texture filtering process

3.5.2.) Experimental Results with Occluded Textured Shapes:

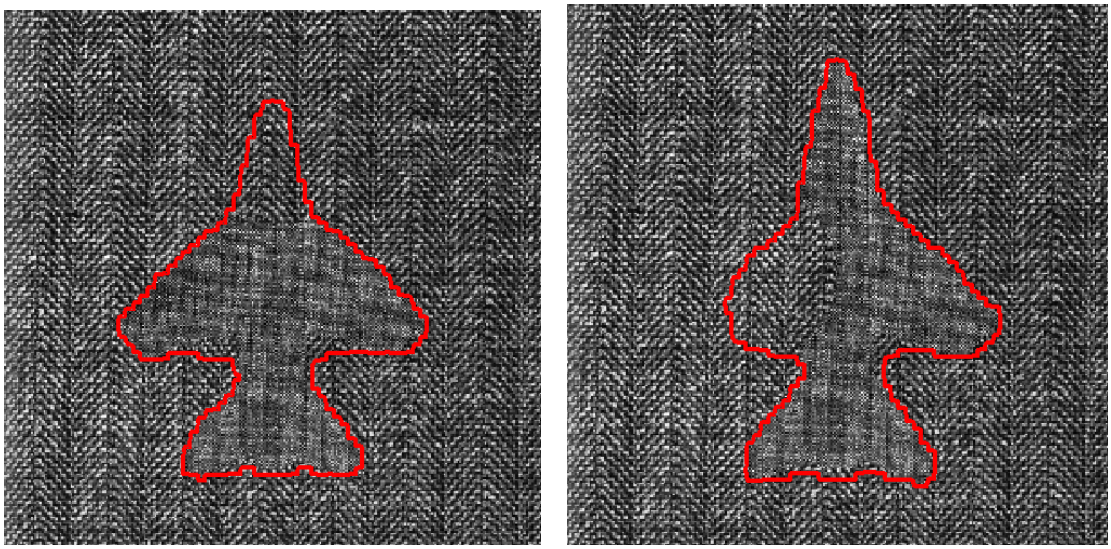
In this section we present our segmentation results at textured images with occlusion problems. We have used the fighter shape for our experiments. We have occluded parts of the fighter and expect our approach to fill the occluded parts and segment the fighter from the textured image. In figure 3.31 we observe that our curve estimates occuluded parts of the fighters accorgind to the information from data and the training shapes. Our algorithm does not have any prior information on the missing data of the shape in the image.



(a) (b)



(c) (d)



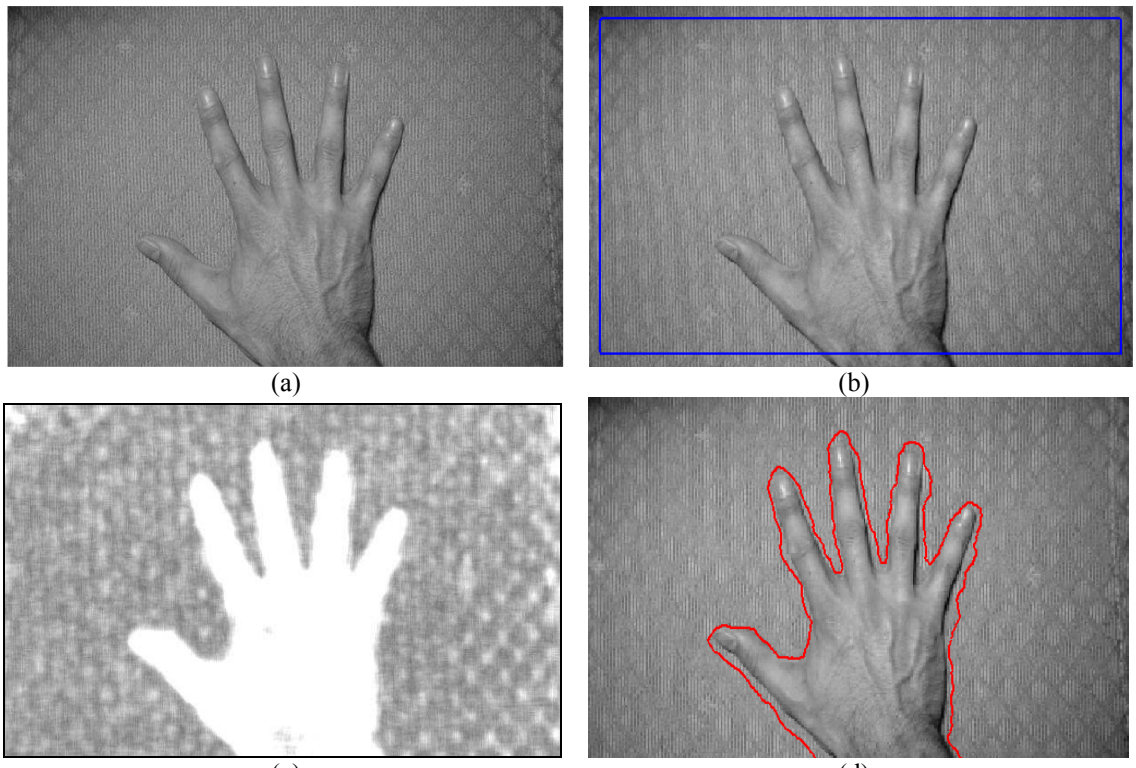
(e) (f)

Figure 3.31: **(a)** Textured image with occluded fighter shaped region **(b)** Textured image with occluded fighter wing shaped region **(c)** Filtering result of (a) **(d)** Filtering result of (b) **(e)** Segmentation result of (a) **(f)** Filtering result of (b)

3.5.3.) Experimental Results with Natural Textured Images:

In this subsection we present our segmentation results on natural images. Our aim is hand segmentation from natural textured images. Shape of the hand is a difficult example for segmentation. Texture structure of hand is also as difficult as its shape. Because texture of hand is a nonuniform texture which includes many different textural structures. Also in some parts of the hand it is difficult to catch the texture structure. For this reason hand segmentation is a challenging problem in texture segmentation. In our experiments we have applied our texture filtering process according to the background of the hands in images. We tried to choose our test images from surfaces with misleading texture structures. Our results are in Figures 3.32, 3.33, 3.34, 3.35, 3.36.

In figure 3.32 we have used a fabric as a background in our test image. The image has low contrast between hand and the fabric and the textural structure of the fabric is smooth at some parts which are probable to observe similar regions in texture of hand. Our segmentation result is satisfactory. There are some over flows and shifts at boundaries which is the result of boundary effect we have mentioned in previous sections. We observe the same problem at other segmentation results too.



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

Figure 3.32) (a) Testimage with hand on a fabric **(b)** Initial curve **(c)** Texture filtering result **(d)** Segmentation result

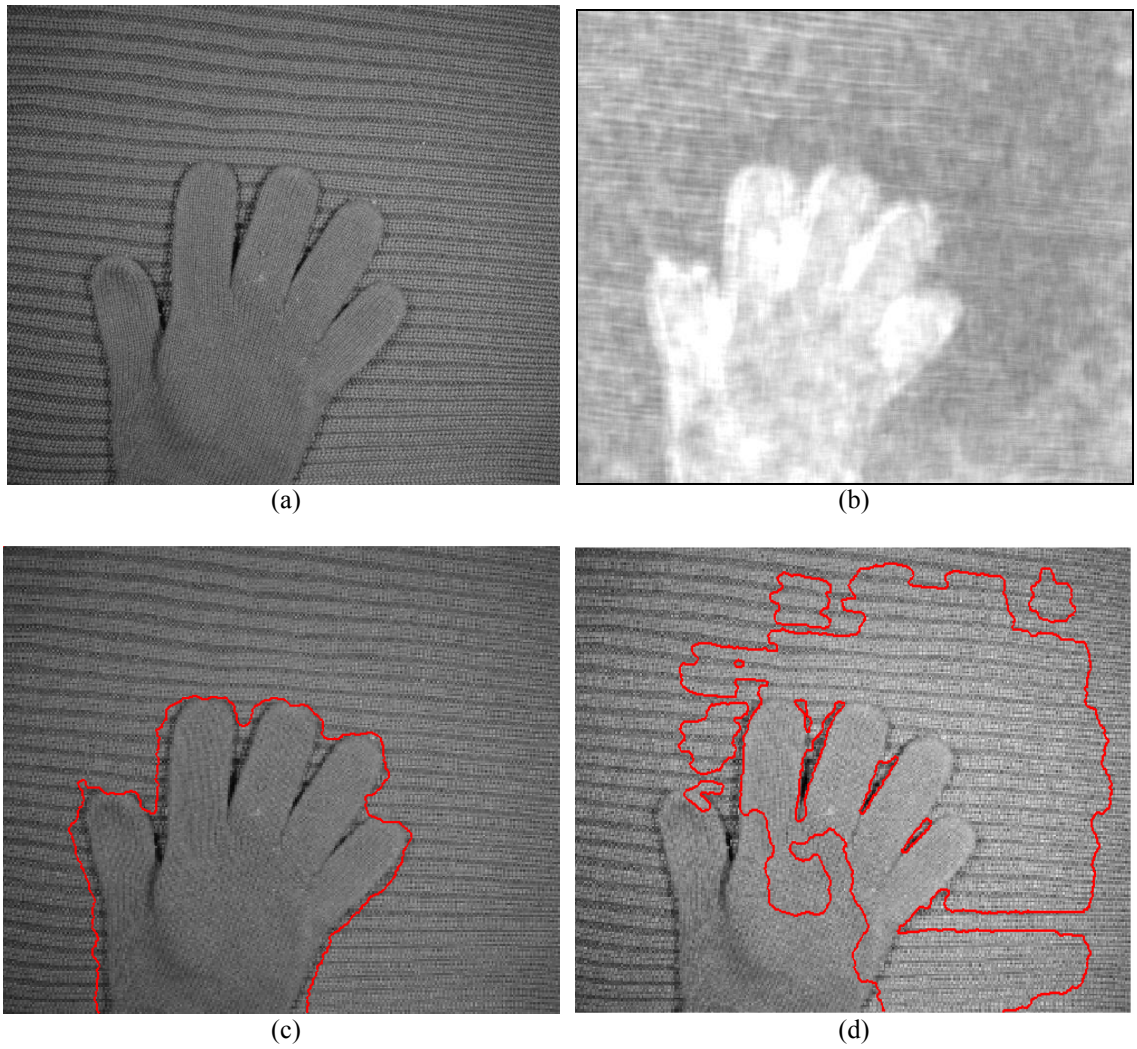
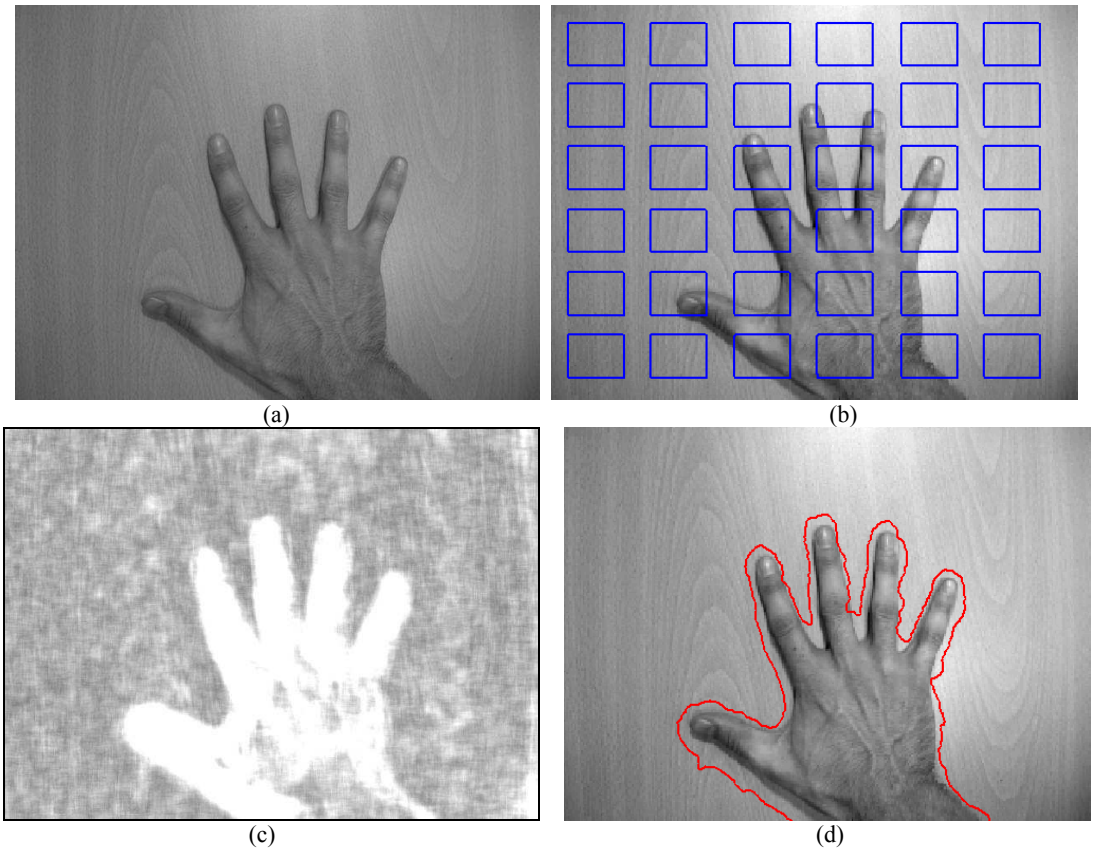


Figure 3.33 : **(a)** Textured image with glove shaped region on a texture of a jumper **(b)** Filtering result of (a) **(c)** Segmentation result of (a) **(d)** Kim's segmentation result without our texture filtering process

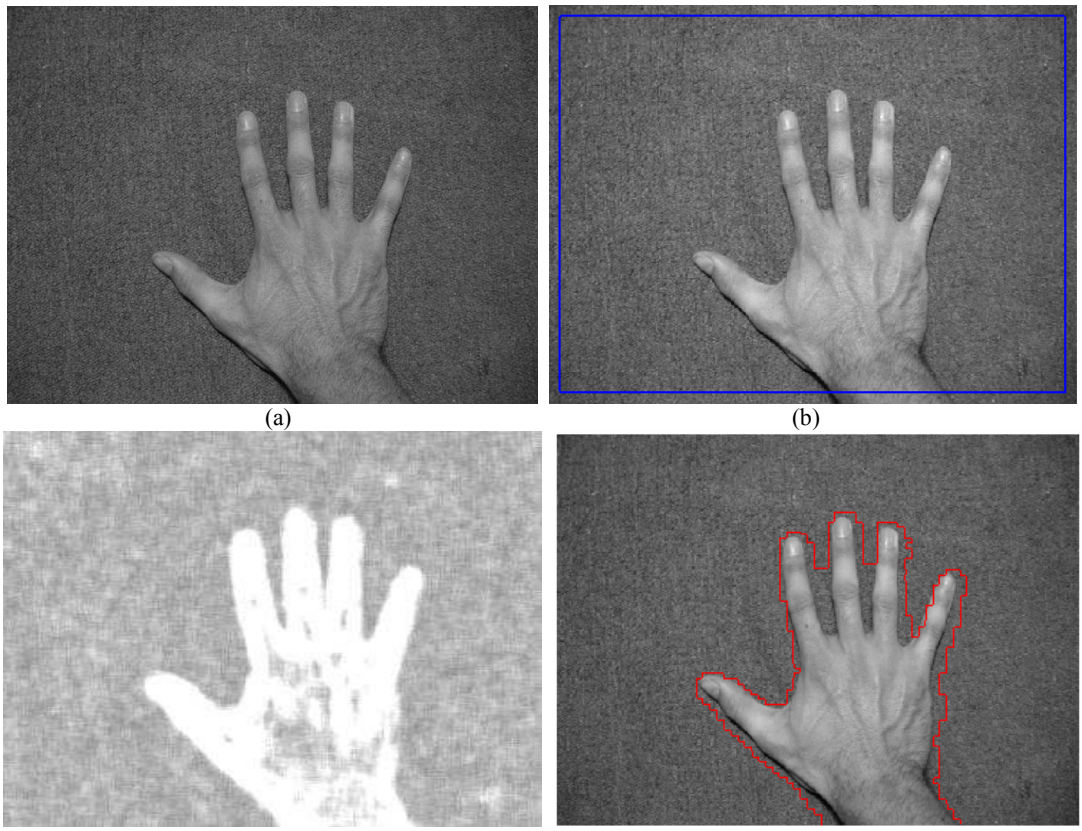
In figure 3.33 (a) we present a real life example. In the image we provide two naturally textured regions with texture of the glove and the texture of the jumper. As we can see from the image there is low contrast between two regions. Also the texture structures of the two regions are misleading since both regions are textile products. As a result of this although our filter produce noisy result, it produces a result with sufficient contrast between regions for satisfactory segmentation.



(a) (b)

(c) (d)

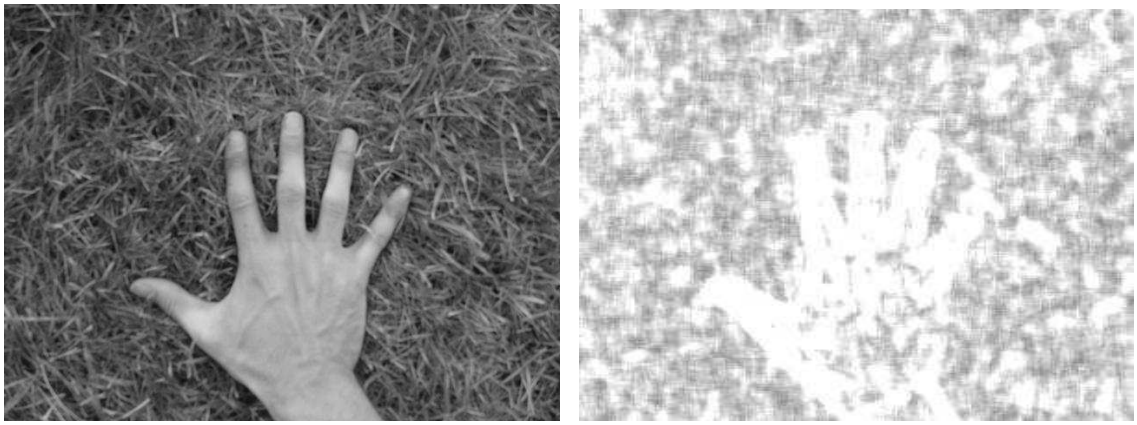
Figure 3.34: (a) Test image with hand on a wooden surface **(b)** The initial Curve **(c)** Texture Filtering Result **(d)** Segmentation Result



(a) (b)

(c) (d)

Figure 3.35: (a) Test image with hand on a carpet **(b)** The initial Curve **(c)** Texture Filtering Result **(d)** Segmentation Result



(a) (b)

(c)

Figure 3.36: Hand segmentation results on grass

When we look our segmentation result we see that our approach gives satisfactory segmentation result. As a result of boundary effect of windowing process, we have some over flows at the boundaries. However, this problem can be minimized by using several approaches which we discuss at our conclusion section.

CHAPTER 4

CONCLUSION

4.1) Summary:

In this thesis we have proposed a shape and data driven texture segmentation approach using an LBP based texture filter and active contours. We have used the texture analysis tool LBP to create our texture filter for eluding the textured images from structural correlation between pixels. In other words we have used the filtering process to produce regions which can be distinguished by pixel intensity values. Then we have produced a data term for our energy functional from filtered image. For this process we have used Kim et al.'s mutual information based data term. For shape term we have used Kim et al.'s framework which estimates shape priors from training shapes in a nonparametric way. Then we have combined nonparametric shape prior and our data term by using Kim et al.'s energy equation. By curve evolution equations for the shape and data term we have minimized the energy functional for segmentation. Our experimental results have shown that we have reached satisfactory segmentation results with uniform and nonuniformed synthetic and real textures. Also in occlusion problems our approach produced satisfactory results.

4.2) Future Work:

- Boundary Effect:

For future research directions, solving the boundary effect problem, which is the result of the windowing process, might be a good starting point. This problem can be alleviated by changing the window size according to an entropy based homogeneity measure in the windows which are close to the boundaries. Another approach can be producing a different texture filter which works without windowing process.

- Vector Valued Version:

Also our approach can be extended for vector valued approaches like Gabor filters by using a suitable distance metric in multi dimensional space.

- Multidimensional Texture Space

Texture structures can be defined on a manifold in a multidimensional space. By using a diffeomorphic metric space the similarity between different textures can be defined.

- Unsupervised Version:

Producing an unsupervised version of our approach would be an important expansion to our work.

- Different Distance Metric:

Defining a different distance metric in shape space to define similarity measure between shapes would be a good contribution to shape analysis part. Distance metric can be defined as a diffeomorphic deformation metric in shape space which measures the amount of deformations required to approximate the target shape.

REFERENCES

[1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces.
J. Comput. Phys., 118:269–277, 1995.

[2] Ibrahim A. Ahmad and Pi-Erh Lin. A nonparametric estimation of the entropy for absolutely continuous distributions.
IEEE Transactions on Information Theory, pages 372–375, May 1976.

[3] T. Chan and L. Vese. Active contours without edges.
IEEE Trans. on ImageProcessing, 10(2):266–277, February 2001.

[4] D. L. Chopp. Computing minimal surfaces via level set curvature flow.
J. Comput. Phys., 106:77–91, 1993.

[5] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application.
Computer Vision and Image Understanding, 61(1):38–59, 1995.

[6] Daniel Cremers and Stefano Soatto. A pseudo-distance for shape priors in level set segmentation.
In *IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, 2003.

[7] Rhodri H. Davies, Tim F. Cootes, and Chris. J. Taylor. A minimum description length approach to statistical shape modeling.
In *Proc. Information Processing in Medical Imaging*, pages 50–63, 2001.

[8] Rhodri H. Davies, Carole J. Twining, Tim F. Cootes, John C. Waterton, and Chris. J. Taylor. A minimum description length approach to statistical shape modeling.
IEEE Trans. on Medical Imaging, 21(5):525–537, 2002.

[9] R.M. Haralick, Statistical and Structural Approaches to Texture,
PIEEE(67), No. 5, May 1979, pp. 786-804.

[10] D. G. Kendall. Shape manifolds, procrustean metrics and complex projective spaces.
Bull. of London Math. Soc., 16:81–121, 1984.

[11] Junmo Kim, PhD. Thesis, Nonparametric Statistical Methods for Image segmentation and Shape Analysis.
Massachusetts Institute of Technology (2000)

[12] Junmo Kim, John W. Fisher, Anthony Yezzi, Jr., Müjdat Çetin, and Alan S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution.
IEEE Trans. Image Processing, vol. 14, no. 10, pp. 1486-1502, October 2005.

[13] Junmo Kim, John, Müjdat Çetin, and Alan S. Willsky. Nonparametric shape priors for active contour based image segmentation.
EURASIP European Signal Processing Conference (EUSIPCO),
September 2005

[14] Michael E. Leventon, W. Eric L. Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours.
In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 316–323. IEEE, 2000.

[15] David Mumford. The problem of robust shape descriptors.
In *Proceedings of the IEEE First International Conference on Computer Vision*, pages 602–606, 1987.

[16] David Mumford and Jayant Shah. Boundary detection by minimizing functionals.
In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 22–26, 1985.

[17] T. Ojala, M. Pietikäinen & D. Harwood (1996) A comparative study of texture measures with classification based on featured distribution**.**
Pattern Recognition, 29(1):51-59.

[18] T. Ojala, M. Pietikäinen & T. Mäenpää (2000).Gray scale and rotation invariant texture classification with local binary patterns.
In: Computer Vision, ECCV 2000 Proceedings, Lecture Notes in Computer Science 1842, Springer, 404-420.

[19] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on the Hamilton-Jacobi formulation.
Journal of Computational Physics, 79:12–49, 1988.

[20] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*.
Springer, 2003.

[21] Maria Petrou, Pedro Garcia Sevilla. Dealing with Texture.
Wiley, 2006

[22] Nikos Paragios and Richid Deriche. Geodesic active regions and level set methods for supervised texture segmentation.
Int. J. Computer Vision, 2002.

[23] Nikos Paragios and Mikael Rousson. Shape priors for level set representations. In *Proc. European Conference in Computer Vision*, 2002.

[24] Emanuel Parzen. On estimation of a probability density function and mode.
Annals of Mathematical Statistics, 33(3):1065–1076, 1962.

[25] C. Sagiv, N. A. Sochen, and Y. Y. Zeevi. "Texture segmentation via a diffusion segmentation scheme in the gabor feature space",
In Proc. Texture 2002, 2nd International Workshop on Texture analysis and Synthesis, Copenhagen, June 2002.

[26] J.A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science*.
Cambridge University Press, 1996.

[27] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*.
Chapman and Hall, 1986.

[28] C. G. Small. *The Statistical Theory of Shapes*.
Springer-Verlag, 1996.

[29] S. Soatto and Anthony Yezzi, Jr. Deformotion, deforming motion, shape average and the joint registration ad segmentation of images.
In *Proceedings of the 7th European Conference on Computer Vision*, pages 32–47, May 2002.

[30] Walter A. Strauss. *Partial Differential Equations: An Introduction*.
Wiley, 1992.

[31] Andy Tsai, Anthony Yezzi, Jr., William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W. Eric Grimson, and Alan Willsky. Model-based curve evolution technique for image segmentation.
In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 463–468, 2001.

[32] Andy Tsai, Anthony Yezzi, Jr., William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W. Eric Grimson, and Alan Willsky. A shape-based approach to the segmentation of medical imagery using level sets.
IEEE Trans. on Medical Imaging, 22(2):137–154, February 2003.

[33] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories.
IEEE Trans. On Automatic Control, 40:1528–1538, 1995.

[34] G.B. Unal, H. Krim, A. Yezzi, Information-Theoretic Active Polygons for Unsupervised Texture Segmentation
International Journal of Computer Vision Volume 62 , Issue 3 (May 2005)
Pages: 199 - 220, 2005 ISSN:0920-5691

[35] Luminita A. Vese and Tony F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model.
International Journal of Computer Vision, 50(3):271–293, 2002.

[36] Zhizhou Wang and Baba C. Vemuri, Tensor Field Segmentation Using Region Based Active Contour Model. T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3024, pp. 304–315, 2004. Springer-Verlag Berlin Heidelberg 2004

[37] Jun Xie Yifeng Jiang Hung-tat Tsui. Segmentation of kidney from ultrasound images based on texture and shape priors.
IEEE Transactions on Medical Imaging, pages: 45- 57 ISSN: 0278-0062, 2005

[38] Anthony Yezzi, Jr., Andy Tsai, and Alan Willsky. A statistical approach to snakes for bimodal and trimodal imagery.
In *Int. Conf. on Computer Vision*, pages 898–903, 1999.

[39] Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion.
Journal of Computational Physics, 127:179–195, 1996.