M. Tech. (Computer Science) Dissertation Series

Feature Sensitive Level Set for Clustering

a dissertation submitted in partial fulfillment of the
requirements for the M. Tech. (Computer Science)
degree of the Indian Statistical Institute

by

Ashish Gupta

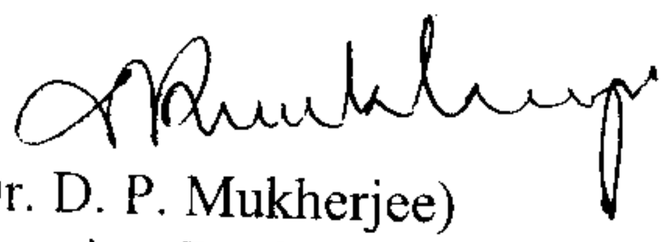
under the supervision of

Dr. D. P. Mukherjee
Associate Professor
Electronics and Communication Sciences Unit

INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road
Kolkata-700108.

Certificate of Approval

This is to certify that the thesis titled *Feature Sensitive Level Set for Clustering* submitted by Ashish Gupta, towards partial fulfillment of the requirements for the degree of M. Tech in Computer Science at the Indian Statistical Institute, Kolkata, embodies the work done under my supervision. His work is satisfactory.



(Dr. D. P. Mukherjee)
Associate Professor
Electronics and Communication Sciences Unit
Indian Statistical Institute, Kolkata

Acknowledgement

This is with affection and appreciation that I acknowledge my indebtedness to Dr. Dipti Prasad Mukherjee, Associate Professor, Indian Statistical Institute, Kolkata, for his advice, enthusiasm, and criticism throughout the course of this dissertation. It would have been very difficult to complete the dissertation without his able guidance. I am thankful to him.

Ashish Gupta 2/7/04

Ashish Gupta
M. Tech (Computer Science)
Indian Statistical Institute, Kolkata

Abstract

Level set analysis is an important tool for curve evolution. Assume a closed curve is embedded inside a 2D matrix. Level set is defined as a distance function of the closed curve where minimum distance of every matrix point from the curve is evaluated. These distances have positive values inside the curve and negative values outside the curve. Consequently, the embedded curve has zero distance (or level set function) value. The evolution of curve inside the matrix using level set analysis is iterative redefinition of the curve distance function based on certain PDE based energy minimization process. Therefore, the evolution of curve is equivalent to evolution of level set function.

In this thesis, we have extended level set analysis for clustering regions in feature space. The energy function for clustering is based on classical definition of clustering in terms of minimizing intra-cluster distances and maximizing inter-cluster distances. Multiple clusters in the feature space are detected using multiple level set functions. In each iteration, the curves are evolved using forces that engulf the nearby points. Again the curves are mutually repelled through maximization of distances between corresponding cluster centers. We have assumed that the number of clusters are known a priori. However, we have also proposed a heuristic that can introduce a new curve and corresponding level set function in case a set of points have a tendency to form a separate cluster. The efficacy of the clustering technique is demonstrated through its clustering performance on both synthetic and real images.

Contents

Chapter 1

Introduction

Image segmentation is one of the basic problems in image analysis and low-level vision. This problem has been studied extensively since early days of research in image processing and computer vision. Image segmentation remains a difficult task due to both the tremendous variability of object shapes and the variation in image quality. Images are often corrupted by noise, which can cause considerable difficulties when applying classical segmentation techniques such as edge detection and thresholding. As a result, these techniques either fail completely or require some kind of post processing step to remove invalid object boundaries in the segmentation results.

One of the main problem in edge detection is the object of interest is delineated with thin discontinuous edge which needs interpolation. To address these difficulties, active contours have been extensively studied. Active contours are curves or surfaces defined within an image domain that can move under the influence of *internal forces*, which are defined within the curve or surface itself, and *external forces*, which are computed from the image data. Active Contours are also referred as deformable models as a given curve is deformed under different forces. The internal forces are designed to keep the model smooth during deformation. The external forces are defined to move the model toward an object boundary or other desired features within an image.

The theory of deformable models is discussed in the paper "Snakes: Active Contours" originally proposed by Kass, Witkin and Terzopoulos [1] for image segmentation problem. Various names such as snakes, active contours or surfaces, balloons, and deformable contours or surfaces, have been used in literature to refer to deformable models.

There are basically two types of deformable models: *parametric deformable models* [1] and *geometric deformable models* [2,3]. Parametric deformable models represent curves and surfaces explicitly in their parametric forms during deformation. For parametric deformable models we have either energy minimization formulation or dynamic force formulation. In energy minimization model we associate energy with the deforming curve. The basic premise of the energy minimizing formulation of deformable contours is to find a parameterized curve that minimizes the energy associated with the curve. Total energy associated with the curve is weighted sum of internal energy and potential energy. The internal energy specifies the tension or the smoothness of the contour. The potential energy is defined over the image domain and takes small values at object boundaries as well as other features of interest. Minimizing the total energy yields internal forces and potential forces. Internal forces hold the curve together (elasticity forces) and keep it from bending too much (bending forces). External forces attract the curve toward the desired object boundaries. To find the object boundary, parametric curves are initialized within the image domain, and are forced to move toward the potential energy minima under the influence of both internal and external forces. It is sometimes more convenient to formulate the deformable model directly from a dynamic problem using a force formulation.

Such a formulation permits the use of more general types of external forces, i.e., forces that cannot be written as the negative gradient of potential energy functions.

Adaptation of the model topology, however, such as splitting and or merging parts during deformation, can be difficult using parametric models. Geometric deformable models provide an elegant solution to address the primary limitations of parametric deformable models. Geometric deformable models can handle topological changes naturally. These models are based on the theory of curve evolution [4] and the level set method [5,6]. In this model curves and surfaces are evolved using only geometric measures such as unit normal and curvature, resulting in an evolution that is independent of the parameterization. Since the evolution is independent of parameterization, the evolving curves and surfaces can be represented implicitly as a level set of higher dimensional function. As a result, topology changes can be handled automatically. Despite this fundamental difference, the underlying principles of both methods are very similar.

Now, we will discuss fundamental concepts in curve evolution theory and level set method.

1.1 Curve Evolution Theory

In curve evolution theory the deformation of curves are done using only geometric measures such as the unit normal and curvature as opposed to the quantities that depend on parameters such as the derivatives of an arbitrary parameterized curve.

Consider a moving curve $X(s,t)=[x(s,t),y(s,t)]$, where s is any parameterization and t is the time, and denote its inward unit normal as $\vec{N}$. Curve can deform in any arbitrary direction but that can always be decomposed into tangential and normal deformation. Tangential deformation affects only the curve's parameterization and not its shape and geometry so we can write the evolution of the curve along its normal direction that can be characterized by the following partial differential equation,

$$\frac{\partial X(s,t)}{\partial t} = V\vec{N}, \qquad (1.1)$$

where V is called speed function, since it determines the speed of the curve evolution. The most extensively used curve deformation in curve evolution theory is *curvature deformation* where speed function V is written in terms of curvature κ. Curvature deformation is given by the equation

$$\frac{\partial X(s,t)}{\partial t} = \alpha\kappa\vec{N}, \qquad (1.2)$$

where α is a positive constant. This equation smoothes the curve, eventually shrinking it to a circular point [15]. The use of curvature deformation has an effect similar to the use of the elastic internal force in parametric deformable models. We will now discuss level set method for implementing curve evolution.

1.2 Level Set Method

The level set method is proposed by Osher and Sethian [5]. The level set method is used to account for automatic topology adaptation, and it also provides the basis for a numerical scheme that is used by geometric deformable models.

In the level set method, the curve is represented implicitly as a level set of a 2D scalar function – referred to as the level set function – which is usually defined on the same domain as the image. Let $\Phi(x,y,t)$ be the level set function. Here, x and y represents coordinates and t is the time. $\Phi(x,y,t)$ can be viewed as a 3D evolving surface. When we take the intersection-set of a plane (plane on which the curve resides) and 3D evolving surface we get our desired evolving curve. The evolving curve can be seen as the zero level set of the function. Zero level set is defined as the intersection-set of points (x,y) such that $\Phi(x,y,t)=0$. For example, a curve in $\Re^2$ can be represented as the zero-level set of a function $\Re^2 \to \Re$ (Figure 1.1(a)(b)). The function is signed distance to the curve, positive inside and negative outside the curve. Signed distance of any point is defined as the minimum distance of the point from the zero level set. Instead of tracking through time, the level set method evolves the curve by updating the level set function at fixed coordinates through time. Figure 1.1(b) shows the height map of the distance function.

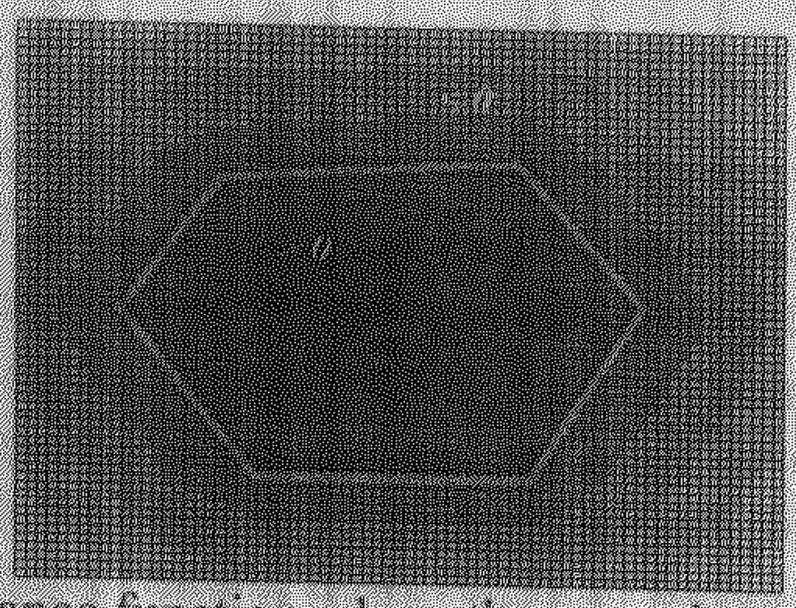


Figure 1.1(a): The distance function where the curve is embedded as the zero level set (in red).

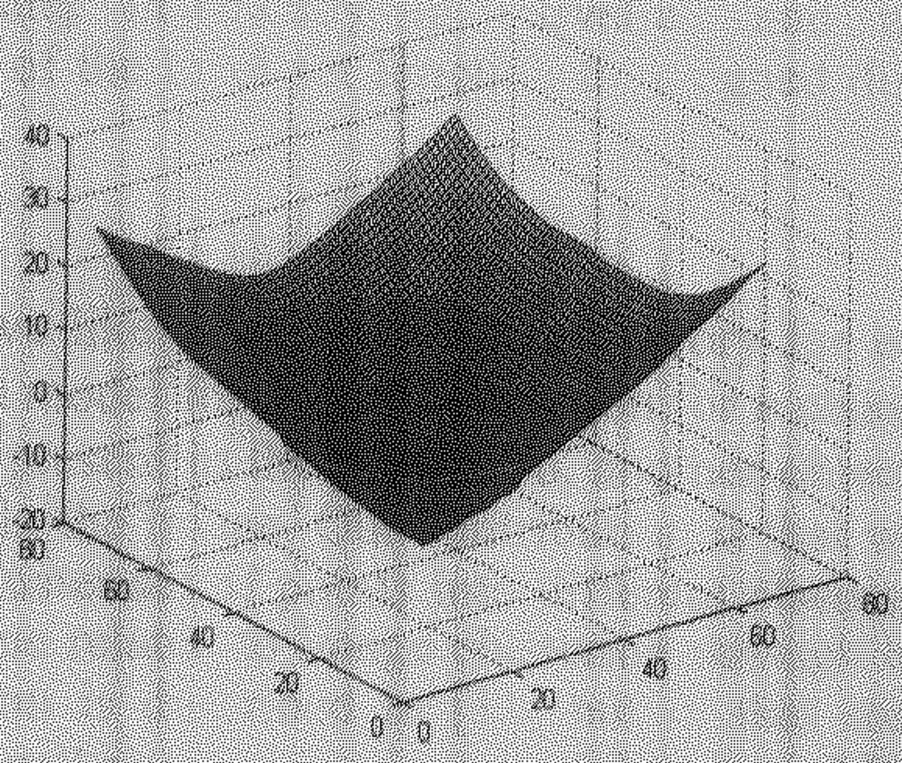


Figure 1.1(b): The height map of the distance function with its zero level set depicted in red.

We now derive the level set embedding of the curve evolution equation. Given a level set function $\Phi(x,y,t)$ with the curve $X(s,t)$ as its zero level set, we have

$$\Phi[X(s,t),t]=0. \qquad (1.3)$$

Differentiating the above equation with respect to t and using the chain rule, we obtain

$$\frac{\partial\Phi}{\partial t}+\nabla\Phi\frac{\partial X}{\partial t}=0, \qquad (1.4)$$

where $\nabla\Phi$ denotes the gradient of Φ.

We assume that Φ is negative inside the zero level set and positive outside. Accordingly, the inward unit normal $\vec{N}$ to the level set curve is given by

$$\vec{N}=-\frac{\nabla\Phi}{|\nabla\Phi|}. \qquad (1.5)$$

Using equation (1.5) and equation (1.1), we can rewrite equation (1.4) as

$$\frac{\partial\Phi}{\partial t}=V|\nabla\Phi|. \qquad (1.6)$$

The relationship between equation (1.1) and equation (1.6) provides the basis for performing curve evolution using the level set method.

There are certain issues, which need to be considered in order to implement geometric deformable contours:

1. An initial function $\Phi(x,y,t=0)$ must be considered such that its zero level set corresponds to the position of the initial curve. A common choice is to set $\Phi(x,y,0)=D(x,y)$, where $D(x,y)$ is the signed distance from each grid point to the zero level set.
2. Since the evolution equation (1.6) is derived from the zero level set only, the speed function V, in general, is not defined on other level sets. Hence, we need a method to extend the speed function V to all of the level sets [6]. We note that the expressions for the unit normal and the curvature, however, hold for all level sets. The level set function that evolves using extended speed functions can lose the property of being a signed distance function, causing inaccuracy in curvature and normal calculations. As a result, reinitialization of the level set function to a signed distance function is often required for these schemes.

In this report we like to extend level set theory for image clustering. Our objective is to develop a generalized model for n curve evolution for n classes in the image. These n classes can have $m \geq n$ segments present in the image. So, another objective is to tackle different clusters of same classes present in the image. Our contribution is that we design a model based on level set method, which can cluster set of points into different classes. The number of classes is known a priori. No work is done related to image clustering through level set method. Some related work on image classification is described in [11] but in that case there is complete knowledge of number of classes as well as mean, variances of features of different classes a priori.

Figure 1.2 shows three different objects including background. There are two segments of class 1 (red), and one each of class 2 (blue) and class 3 (green) respectively. We want to evolve three curves to track these three classes.

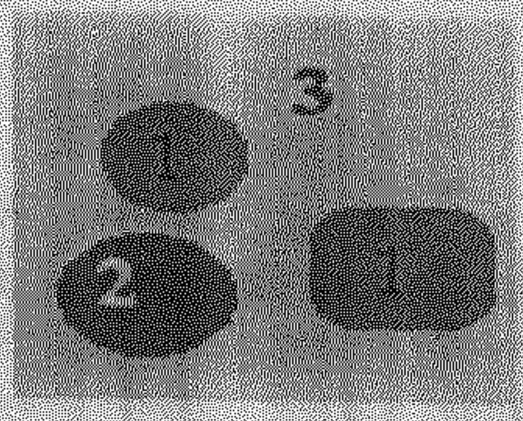


Figure 1.2: Image having three classes.

In Chapter 2, we will describe the literature survey. In Chapter 3, we will discuss our proposed model for curve evolution and in chapter 4 we show the results of our model on real and synthetic images.

Chapter 2

Related Work

The basic idea in active contour models or snakes is to evolve a curve, subject to constraints from a given image, in order to detect objects in that image. For example, starting a curve around the object to be detected, the curve moves in its normal direction and has to stop on the boundary of the object. Hence forward, the terms active contours and zero level set curve means the same thing.

Chan and Vese [7] have implemented active contours without edges to capture image edges through minimizing an energy functional F given by

$$F(c_1, c_2, C) = \mu \text{Length}(C) + \nu \text{Area}(\text{inside}(C)) + \lambda_1 \iint_{\text{inside}(C)} |u_0(x,y) - c_1|^2 \, dxdy + \lambda_2 \iint_{\text{outside}(C)} |u_0(x,y) - c_2|^2 \, dxdy \quad (2.1)$$

The initial curve is C having mean intensities inside and outside the curve as c_1 and c_2 respectively. The input image subjected to image segmentation is specified by $u_0(x,y)$. The weights defining relative importance of each of the energy terms are given by $\mu \geq 0$, $\nu \geq 0$, λ_1 and $\lambda_2 > 0$. The advantage of this approach is that for segmentation no edge information is required and therefore, the zero level set is more likely to be robust with respect to image noise.

In the level set formulation, curve C in Ω is represented by the zero level set of a real valued function $\Phi : \Omega \to \Re$. Ω represents the domain in which the curve is evolved e.g. image domain. The real valued function Φ is defined as the level set function.

Terms in the energy F is expressed in the following way:

$$\begin{aligned} \text{Length}\{\Phi = 0\} &= \iint_{\Omega} |\nabla H(\Phi(x,y))| \, dxdy \\ &= \int_{\Omega} \delta_0(\Phi(x,y)) |\nabla \Phi(x,y)| \, dxdy, \quad (2.2) \\ \text{Area}\{\Phi \geq 0\} &= \int_{\Omega} H(\Phi(x,y)) dxdy, \quad (2.3) \end{aligned}$$

where Heavyside function H, and the one-dimensional Dirac measure δ_0, are defined, respectively by,

$$H(z) = \begin{cases} 1, \text{ if } z \geq 0 \\ 0, \text{ if } z < 0, \end{cases}$$

$$\delta_0(z) = \frac{d}{dz} H(z).$$

Other terms can be written as

$$\int_{\Phi>0} |u_0(x,y) - c_1|^2 \, dxdy = \iint_{\Omega} |u_0(x,y) - c_1|^2 \, H(\Phi(x,y)) dxdy, \quad (2.4)$$

$$\int_{\Phi<0} |u_0(x,y) - c_2|^2 \, dxdy = \iint_{\Omega} |u_0(x,y) - c_2|^2 \, (1 - H(\Phi(x,y))) dxdy. \quad (2.5)$$

Now, for narrow band implementation, regularized versions of H and δ_0, denoted by H_ε and δ_ε as $\varepsilon \to 0$, are considered. Using equations (2.2), (2.3), (2.4) and (2.5) the associated regularized functional is given by [7]

$$F_\varepsilon(c_1,c_2,\Phi) = \mu \int_\Omega \delta_\varepsilon(\Phi(x,y))|\nabla\Phi(x,y)|\,dxdy + \nu \int_\Omega H_\varepsilon(\Phi(x,y))dxdy$$
$$+\lambda_1 \int_\Omega |u_0(x,y)-c_1|^2 H_\varepsilon(\Phi(x,y))dxdy + \lambda_2 \int_\Omega |u_0(x,y)-c_2|^2 (1-H_\varepsilon(\Phi(x,y)))dxdy. \quad (2.6)$$

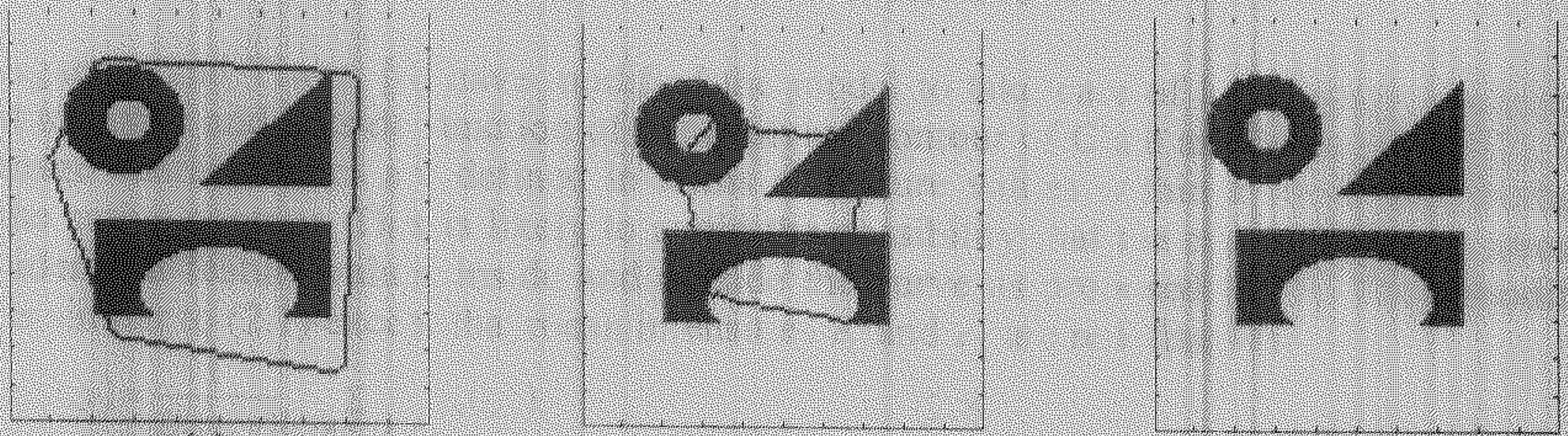
The associated Euler-Lagrange equation [7] is obtained by minimizing the functional F_ε with respect to Φ. Parameterizing the descent direction by an artificial time $t \geq 0$, the equation in $\Phi(x,y,t)$ is

$$\frac{\partial\Phi}{\partial t} = \delta_\varepsilon(\Phi)\left[\mu \operatorname{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right) - \nu - \lambda_1(u_0-c_1)^2 + \lambda_2(u_0-c_2)^2\right] = 0, \quad (2.7)$$

where $\Phi(x,y,0) = \Phi_0(x,y)$ is the initial contour.

The evolution of the active contour in Chan-Vese model [7] is effectively guided by the difference of mean intensities of inside and outside of the active contour. In case this difference is marginal, the speed of evolution of active contour decreases. As they have taken $\lambda_1 = \lambda_2$ the last two terms cancel out, thereby decreasing the evolution speed.

Figure 2.1(a) shows the image having three objects and initial contour. Figure 2.1(b) and 2.1(c) shows how the active contour captures the desired objects.



(a) (b) (c)

Figure 2.1: (a) Image of size 100 x 100 having three objects and zero level set curve at iteration t = 0 (b) Zero level set curve at iteration t = 10. (c) Zero level set curve at iteration t = 30.

The basic model is targeted for binary images. This model fails for images having multiple contrasting objects. Figure 2.2 shows an example where Chan and Vese [7] model fails. This happens because the image contains two objects of contrasting intensity values and this model looks for approximation of the image as a function of only two values representative of image statistics (mean values) inside and outside the evolving curve. The object having lighter contrast is not captured because the intensity of that object is closer to the background intensity as compared to the intensity of the object having darker contrast. Hence, this model could not capture both the objects present in the image.

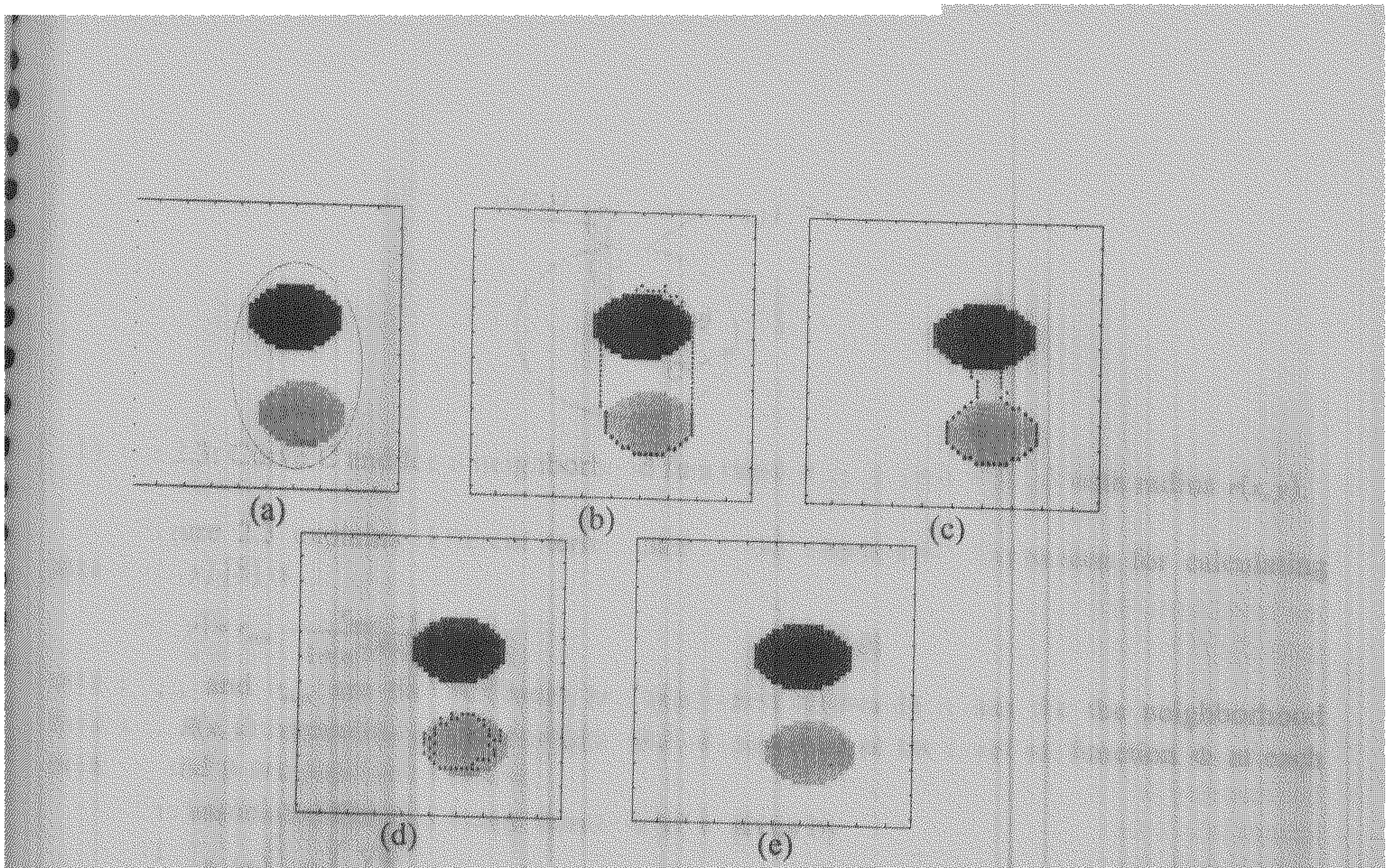


Figure 2.2: (a) Image of size 100 x 100 having two contrasting objects and initial zero level set curve (b) zero level set curve at iteration $t = 3$ (c) zero level set curve at iteration $t = 6$ (d) zero level set curve at iteration $t = 9$ (e) zero level set curve at iteration $t = 15$.

Agarwal and Tiwari [8] have modified the model to take care of local image statistics in contrast to inside and outside function of the evolving contour of Chan and Vese model [7]. The local image characteristics are defined in terms of feature values within a small neighborhood along the active contour. This is likely to avoid the possibility that the curve motion is arrested due to identical feature values inside and outside the curve as proposed in [7]. Their modified model is similar to (2.1) except that mean intensities c_1 and c_2 are redefined as

$$c_1(x,y) = \frac{1}{L_-(x,y)} \sum_{[i,j] \in N_-(x,y)} u_0(i,j) \tag{2.8}$$

and

$$c_2(x,y) = \frac{1}{L_+(x,y)} \sum_{[i,j] \in N_+(x,y)} u_0(i,j), \tag{2.9}$$

where $N_+(x,y)$ and $N_-(x,y)$ for any position (x,y) represents the local neighborhood at location (x,y) of radius $r(x,y)$ at a distance $r(x,y)$ in the normal direction to the contour, outside and inside of curve C, respectively. The radius $r(x,y)$ at any point depends on curvature of the curve at that point. If curvature is high then radius is small and vice versa. $L_+(x,y)$ and $L_-(x,y)$ are the cardinality of the sets $N_+(x,y)$ and $N_-(x,y)$ respectively.

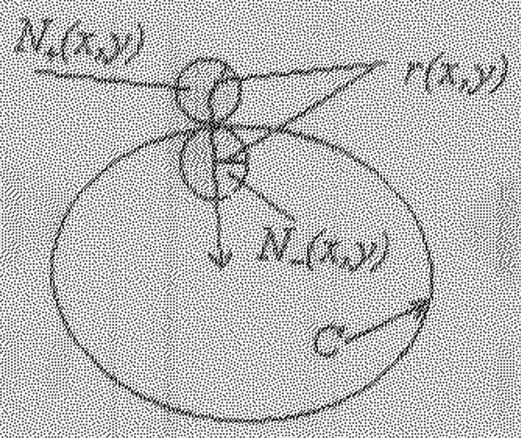


Figure 2.3: Curve C and local neighborhood functions $N_+(x,y)$ and $N_-(x,y)$ with radius $r(x,y)$.

Figure 2.3 explains different terms used in the expression. Expression for calculating radius $r(x,y)$ [8] is

$$r(x,y) = r_{min} + \frac{r_{max} - r_{min}}{1 + e^{-(R(x,y) - r_0)/\beta}}, \tag{2.10}$$

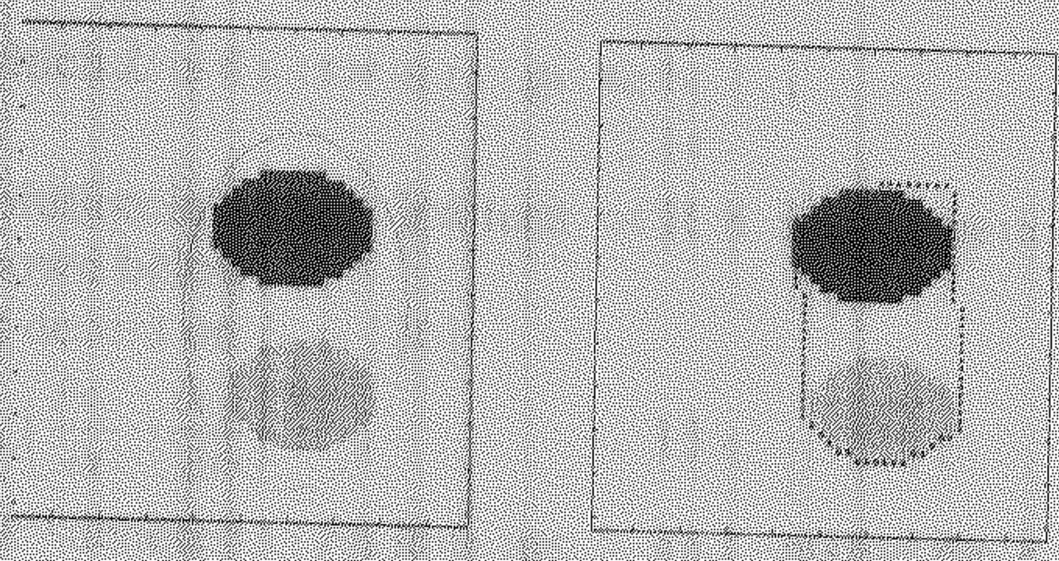
where r_{min} and r_{max} are the minimum and maximum allowable radius for the neighborhood function. $R(x,y)$ represents the estimated radius of curvature of the level set function Φ at each pixel. r_0 and β are positive constants.

The associated Euler-lagrange equation [8] is given by

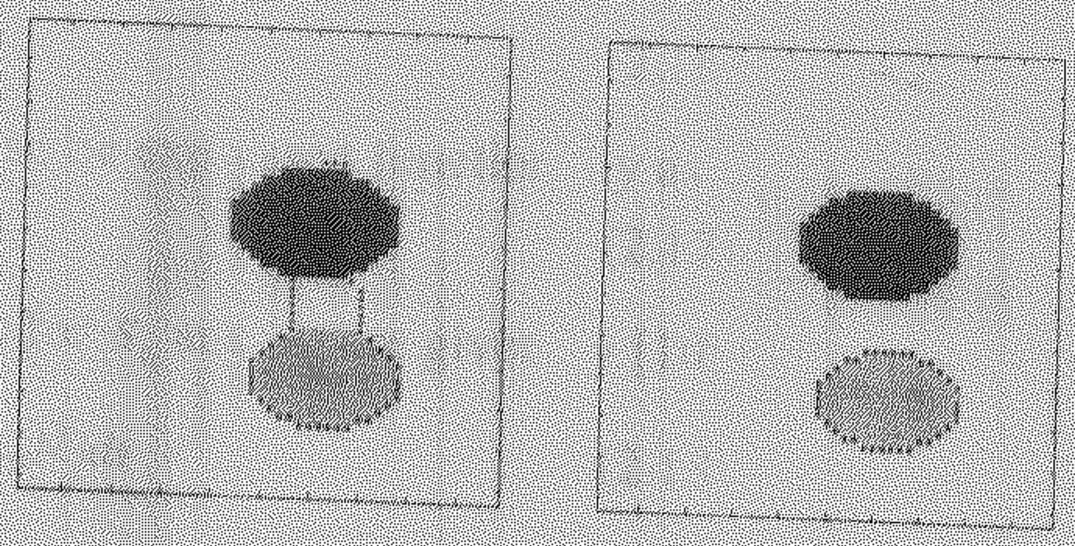
$$\frac{\partial \Phi}{\partial t} = \delta_\varepsilon(\Phi)\left[\mu \operatorname{div}\left(\frac{\nabla \Phi}{|\nabla \Phi|}\right) - \nu - \lambda_1(u_0 - c_1(x,y))^2 + \lambda_2(u_0 - c_2(x,y))^2\right] = 0, \tag{2.11}$$

where $\Phi(x,y,0) = \Phi_0(x,y)$ is the initial contour.

Figure 2.4 shows the same example where Chan and Vese model [7] fails but Tiwari and Agrawal [8] model works. Model proposed by Tiwari and Agrawal [8] correctly captures both the objects in the image but this model is computationally expensive, as model needs local calculation for every point on the curve.



(a) (b)



(c) (d)

Figure 2.4: (a) Image of size 100 x 100 having two contrasting objects and initial zero level set curve (b) zero level set curve at iteration $t = 3$ (c) zero level set curve at iteration $t = 6$ (d) zero level set curve at iteration $t = 9$.

Models proposed in [7] and [8] cannot be generalized for segmenting more than three regions. Tsai and Yezzi [9] proposed coupled curve evolution model, which can be generalized for multiple regions derived from a single level set. For binary images, the Tsai and Yezzi model [9] is almost identical to Chan and Vese model [8]. The energy functional for binary flow (since the curve is separating object from background) is given by,

$$E = -\frac{1}{2}(c_1 - c_2)^2 + \alpha L, \tag{2.12}$$

where c_1 and c_2 are mean feature values for inside and outside of the curve as defined in (2.1). The evolution of curve is such that the feature c_1 and c_2 are maximally separated. The negative sign in the energy functional is introduced because energy E is to be minimized. L is the length of the curve C. $\alpha \geq 0$ is a parameter that adjusts the relative strength of the curve length penalty term.

L can be written as

$$L = \oint_C ds, \tag{2.13}$$

The evolution in which L is decreasing is given by [9]

$$C_t = -\kappa \vec{N}, \tag{2.14}$$

For finding the evolution in which first term of R.H.S. of equation (2.12) decreases most rapidly, the derivation is shown below.

The gradient decent on E is [10]

$$\frac{\partial E}{\partial t} = -(c_1 - c_2)\left(\frac{\partial c_1}{\partial t} - \frac{\partial c_2}{\partial t}\right). \tag{2.15}$$

Let R^{c_1} and R^{c_2} denote the regions inside and outside the curve C, respectively. Since c_1 is the average intensity of the region inside the curve C, it can be written as $\frac{S_{c_1}}{A_{c_1}}$ where S_{c_1} is the sum of intensity in R^{c_1}. So,

$$S_{c_1} = \iint_{R^{c_1}} u_0 dA \,. \tag{2.16}$$

The area A_{c_1} can also be expressed functionally as

$$A_{c_1} = \iint_{R^{c_1}} dA \,. \tag{2.17}$$

The gradient flows for S_{c_1} and A_{c_1} are given by [10]

$$\frac{\partial S_{c_1}}{\partial t} = \oint_C \langle C_t, u_0 \vec{N} \rangle ds \tag{2.18}$$

and

$$\frac{\partial A_{c_1}}{\partial t} = \oint_C \langle C_t, \vec{N} \rangle ds \,. \tag{2.19}$$

where $\vec{N}$ is the outward normal with respect to R^{c_1}. $\langle\ \rangle$ denotes the dot product. Using quotient rule we get

$$\frac{\partial c_1}{\partial t} = \frac{A_{c_1} \dfrac{\partial S_{c_1}}{\partial t} - S_{c_1} \dfrac{\partial A_{c_1}}{\partial t}}{A_{c_1}^2}$$

$$= \frac{1}{A_{c_1}} \oint_C \langle C_t, (u_0 - c_1)\vec{N} \rangle ds \,. \tag{2.20}$$

A similar result holds for gradient flow for c_2 as

$$\frac{\partial c_2}{\partial t} = -\frac{1}{A_{c_2}} \oint_C \langle C_t, (u_0 - c_2)\vec{N} \rangle ds \tag{2.21}$$

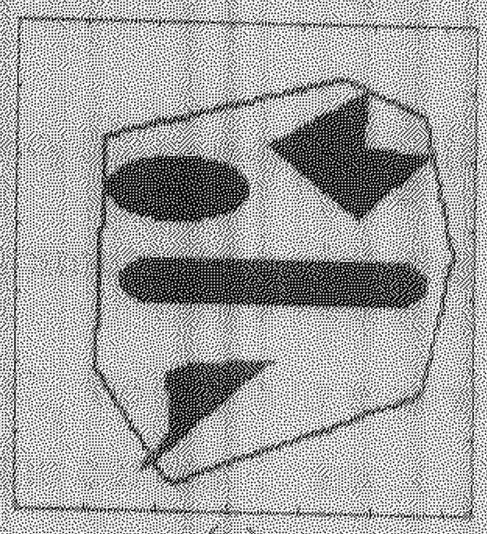
Substituting the expressions of (2.20) and (2.21) into (2.15) the curve evolution that decreases E most rapidly [10] is given by

$$C_t = -\frac{\partial E}{\partial t} = (c_1 - c_2)\left(\frac{u_0 - c_1}{A_{c_1}} + \frac{u_0 - c_2}{A_{c_2}} \right) \vec{N} \,. \tag{2.22}$$

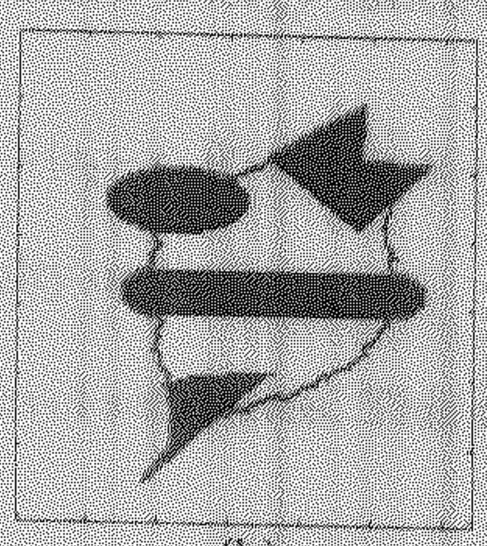
Combining (2.14) and (2.22) we get the corresponding curve evolution equation [9] as

$$\Phi_t = (c_2 - c_1)\left(\frac{u_0 - c_1}{A_{c_1}} + \frac{u_0 - c_2}{A_{c_2}} \right) + \alpha \nabla \cdot \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) |\nabla \Phi| \tag{2.23}$$

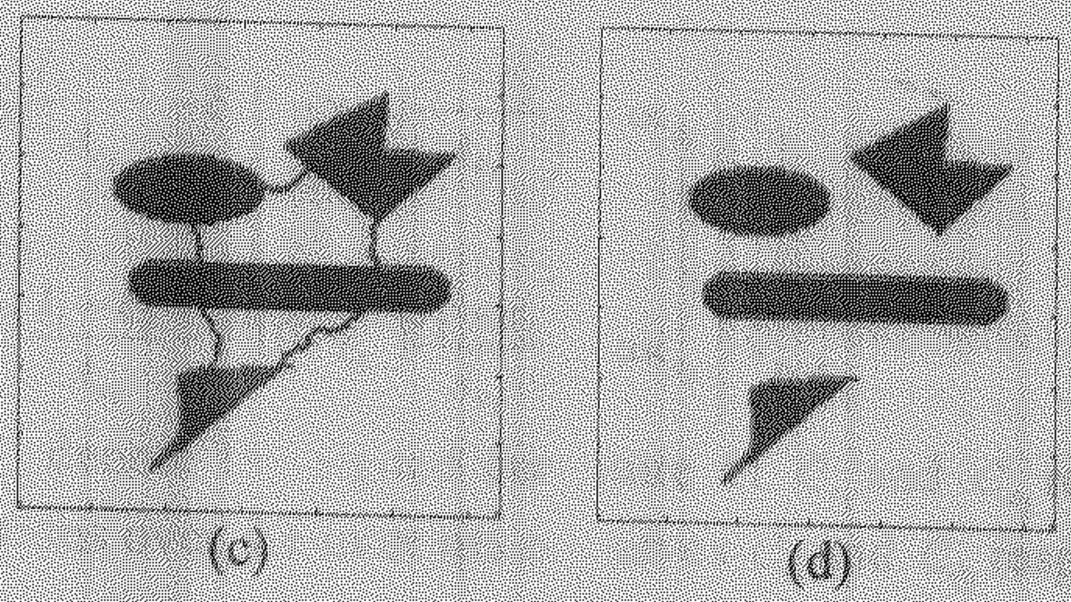
Figure 2.5 shows multiple objects are captured by a single contour using [9].



(a)



(b)



(c) (d)

Figure 2.5: (a) Image of size 125 x 140 having two regions (black and white) and initial zero level set curve (b) zero level set curve at iteration $t = 50$ (c) zero level set curve at iteration $t = 100$ (d) zero level set curve at iteration $t = 170$.

This model [9] can be extended for ternary flows where two level set curves are being evolved to capture two different objects in the image from the image background. Let the image consists of two regions R^a and R^b and a background region R^c with distinct intensities I^a, I^b and I^c. A closed curve in image domain generally encloses some portion of each region. Thus, average intensity inside the curve can be written as convex combination of I^a, I^b and I^c. If image takes its values in $\Re$, there is no unique convex combination since any three points in $\Re$ are collinear. But this model relies on geometrically independent statistics to distinguish different regions.

To get geometrically independent statistics, image u_0 is considered as a vector-valued image with vectors in $\Re^2$. All the intensity information can be expressed in two components of vector-valued image like red channel and green channel information etc. Now, the average intensities like $c_1 = (c_1^1, c_1^2)$ can be uniquely expressed as a convex combination of $I^a = (I_1^a, I_2^a)$, $I^b = (I_1^b, I_2^b)$ and $I^c = (I_1^c, I_2^c)$. Because of geometric independence I^a, I^b and I^c forms the vertices of triangle T_{abc}. Also, the mean intensities c_1, c_2 and c_3 form the vertices of triangle $T_{c_1c_2c_3}$ which is completely contained in T_{abc}. Here, c_3 is the mean intensity of the region mutually outside to both the evolving curves. Energy functional is such that it maximizes the area of triangle $T_{c_1c_2c_3}$. The vector-valued image can take values in $\Re^n$ where $n \geq 2$. Also, in the energy functional geometric penalties to curve lengths are added. Energy functional E for trimodal imagery [9] becomes

$$E = -\frac{1}{2}\| c_1 - c_3 \|^2 \| c_2 - c_3 \|^2 + \frac{1}{2}((c_1 - c_3)\cdot(c_2 - c_3))^2 + \alpha\left(\oint_{C_1} ds + \oint_{C_2} ds\right), \qquad (2.24)$$

where $\alpha \geq 0$ is a parameter to weigh the penalty on both curves.

Let C_{c_1} and C_{c_2} be the two evolving level set curves. This model [9] generates the level set equation for both the curves C_{c_1} and C_{c_2} as

$$\frac{\partial C_{c_1}}{\partial t} = \left\{ \sum_{i=1}^{n} \left(\bar{c}_3^i \frac{I_i - c_1^i}{A_{c_1}} - \bar{c}_2^i (1 - \chi_{c_2}) \frac{I_i - c_3^i}{A_{c_2}} \right) - \alpha \kappa_{c_1} \right\} \vec{N}_{c_1} \tag{2.25}$$

$$\frac{\partial C_{c_2}}{\partial t} = \left\{ \sum_{i=1}^{n} \left(\bar{c}_1^i \frac{I_i - c_2^i}{A_{c_2}} - \bar{c}_2^i (1 - \chi_{c_1}) \frac{I_i - c_3^i}{A_{c_2}} \right) - \alpha \kappa_{c_2} \right\} \vec{N}_{c_2}, \tag{2.26}$$

where κ_{c_1} and κ_{c_2} denote the signed curvature of C_{c_1} and C_{c_2} respectively, χ_{c_1} and χ_{c_2} are the characteristic functions over the interiors of curves C_{c_1} and C_{c_2} respectively.

Figure 2.6 shows three objects including background which are captured by coupled equation (2.25) and (2.26).

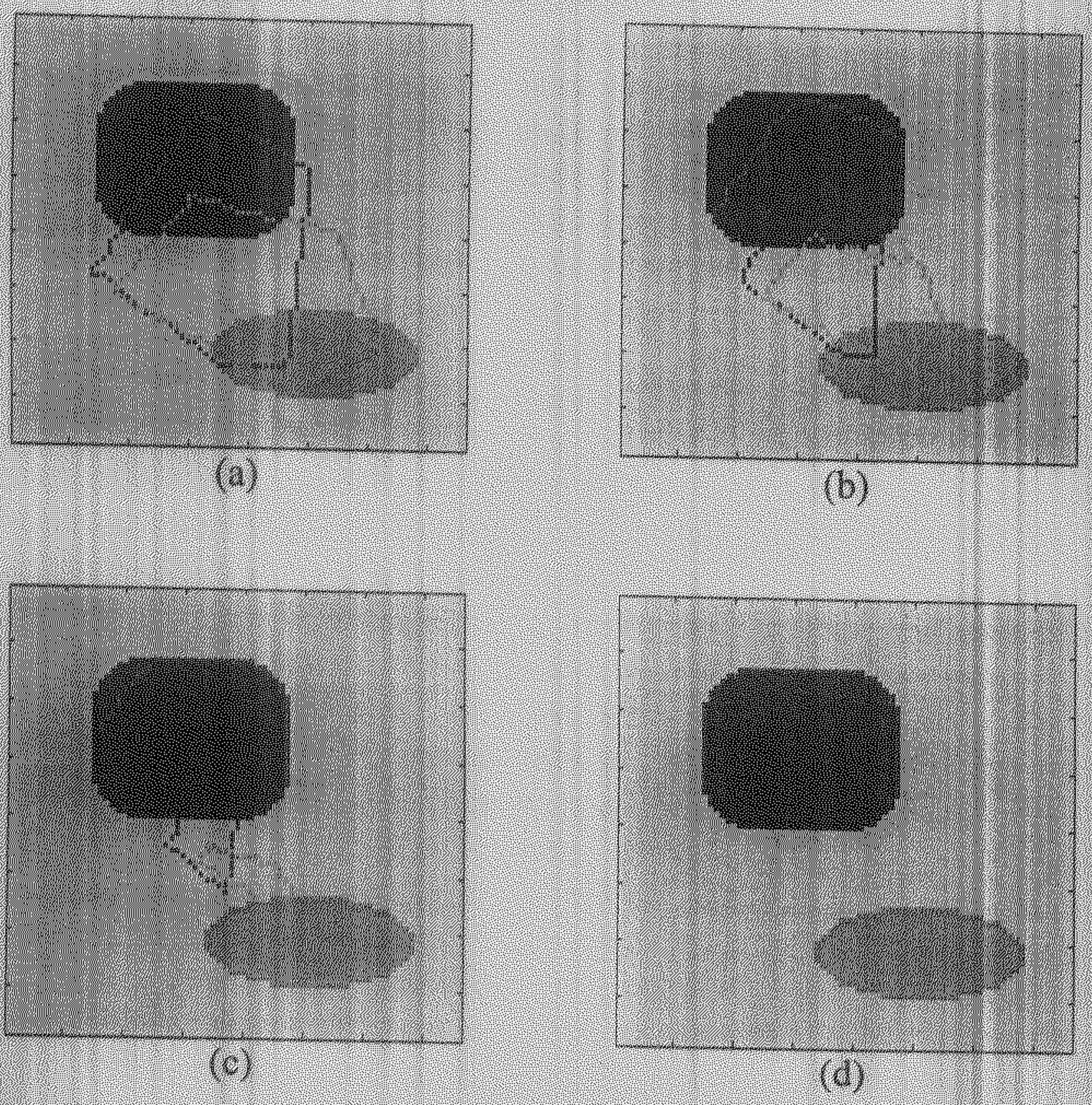


Figure 2.6: (a) Image of size 80 x 80 having three regions and two initial zero level set curves (b) zero level set curves at iteration $t = 8$ (c) zero level set curves at iteration $t = 16$ (d) zero level set curves at iteration $t = 25$.

The drawback of these models [7,8,9] is that they cannot be easily generalized for segmenting multiple regions present in the image. For model [9] in order to segment $n \geq 3$ regions we require a generalized equation for volume of simplex, which is a difficult problem. So, we need a generalized model, which can be easily extended for segmenting multiple regions. In the next chapter we are going to discuss our proposed generalized model for segmenting n regions. In our model we are evolving the curves in feature space instead of image space. So, problem of segmenting different regions is converted into clustering problem in feature space.

Chapter 3

Development of Generalized Model

Our aim is to develop a generalized model for n curve evolution for n clusters. The need for this is evident, as all the models described above [7,8,9] cannot be easily extended to n curve evolution. Tsai-Yezzi model [9] can be generalized to separate n regions. For that we require $(n-1)$ geometrically independent statistic, which forms a simplex in $\Re^{n-1}$. Then maximization of volume of simplex in $\Re^{n-1}$ results in $(n-1)$ coupled evolution equations. The problem with this approach [9] is the need of generalized equation for volume of simplex, which gives the desired evolution equation. But this is quite a difficult problem.

One of the major differences in our proposed model and models described earlier is that we are evolving curves in feature space instead of image space. The idea is to calculate two features on each pixel based on local information of the image. The features can be anything like mean, standard deviation, red channel information of image etc. Ideally, features should be such that they separate the classes well apart in feature space.

So, first step is to transform the points in image space into feature space. Now, our problem becomes clustering of points in feature space. We have extended level set analysis for clustering of points.

Feature space is treated as a binary image having points (which are class pixels (say 1)) and background pixels (say 0). The advantage of doing classification in feature space is that different clusters of same class in image space can be easily tackled, as all the points of same class are grouped together in a single region in feature space. We have used 2D feature space but it may be extended to 3D feature space for 3D curve evolution.

In clustering we want to maximize the inter-class distance and minimize the intra-class distance. We will evolve n level sets for n clusters such that each level set will encompass all the points belonging to one class. Hence, curve evolution will be based on some distance measure (discussed later) as opposed to other models where curve evolution depends on maximization of difference of inside and outside region of curve statistics.

The main idea is each level set is deformed in such a way that it encompasses the near by points and try to push itself away from the points which belongs to other clusters. These conditions ensure the maximization of inter-class distance and minimization of intra-class distance. Also, some geometrical constraints are imposed like length shortening on evolving level sets. Length shortening constraint is added to keep the level set tightly packed with the captured points. Another constraint in clustering is if one point belongs to one cluster then it cannot belong to any other cluster. It means level sets should be non-overlapping after convergence. All these issues are taken care while designing the curve evolution equations.

3.1 Proposed Approach

Let us define the evolving curve C in Ω, as the boundary of an open subset ω of Ω (i.e. $\omega \subset \Omega$ and $C = \partial\omega$). Inside(C) denotes the region ω and outside(C) denotes the region $\Omega \setminus \omega$. In the level set method, $C \subset \Omega$ is represented by the zero level set of a function $\Phi : \Omega \to R$ such that

$$C = \partial\omega = \{(x, y) \in \Omega : \Phi(x, y) = 0\}$$
$$\text{inside}(C) = \omega = \{(x, y) \in \Omega : \Phi(x, y) > 0\}$$
$$\text{outside}(C) = \Omega \setminus \omega = \{(x, y) \in \Omega : \Phi(x, y) < 0\}$$

For the level set formulation, our geometric deformable contour formulation takes the following form

$$\frac{\partial \Phi}{\partial t} = \alpha\kappa |\nabla\Phi| + \beta V |\nabla\Phi| \qquad (3.1)$$

where κ is the curvature given by $\nabla \cdot \frac{\nabla\Phi}{|\nabla\Phi|}$. First term of R.H.S of (3.1) accounts for the length shortening of the curve and second term takes care of maximization of inter-class distance and minimization of intra-class distance. V is the speed term, which is designed in such a way that clustering constraints are satisfied. Clustering constraints are maximization of inter-class distance and minimization of intra-class distance. V is dynamic in nature meaning thereby, it changes with time. $\alpha > 0$ is a parameter to weigh the penalty of length of the curve and $\beta > 0$ is the weight for second term.

The main challenge is to design the dynamic speed term V. Let $u_0(x, y)$ be the binary image representing feature space. To find n clusters we are having n curves and, therefore, n level sets Φ_i i=1 to n. Curve evolution equations looks like

$$\frac{\partial \Phi_i}{\partial t} = \alpha\kappa_i |\nabla\Phi_i| + \beta V_i |\nabla\Phi_i|, \quad \forall i \qquad (3.2)$$

Calculation of speed term V_i

Let $(\bar{x}_i, \bar{y}_i)$ be the coordinate of center of gravity of inside of curve C_i. In order to find the coordinate of center of gravity we need to consider points lying inside the level set curve C_i only. For finding the points lying inside curve C_i we are using Heavyside function defined as

$$H(z) = \begin{cases} 1, \text{ if } z \geq 0 \\ \\ 0, \text{ if } z < 0. \end{cases}$$

Now, the center of gravity of curve C_i is given by

$$\bar{x}_i = \frac{\int_\Omega xH(\Phi_i(x, y))dxdy}{\text{Area}(\text{inside}(C_i))} \qquad (3.3)$$

$$\bar{y}_i = \frac{\int_\Omega yH(\Phi_i(x, y))dxdy}{\text{Area}(\text{inside}(C_i))} \qquad (3.4)$$

and

$$\text{Area}(\text{inside}(C_i)) = \int_{\Omega} H(\Phi_i(x,y))dxdy . \tag{3.5}$$

$u_0(x,y)$ represents the binary image of feature space, which takes the value either 1 (for class pixels) or 0. Now, speed function $V_i(x,y)$ is defined as follows:

$$V_i(x,y) = \begin{cases} u_0(x,y)/(1+r_i(x,y)) & \text{if } \Phi_i(x,y) \geq 0 \\ -u_0(x,y) * r_i(x,y) & \text{if } \Phi_j(x,y) \geq 0 \text{ for any } j \neq i, \\ u_0(x,y)/(1+r_i(x,y)) & \text{if } \Phi_k(x,y) < 0 \text{ for all } k = 1,2,\ldots,n \end{cases} \tag{3.6}$$

where (x,y) is the coordinate of binary image representing feature space. $r_i(x,y)$ is the distance between the points (x,y) and $(\bar{x}_i, \bar{y}_i)$. Distance $r_i(x,y)$ can be taken as

$$r_i(x,y) = |x - \bar{x}_i| + |y - \bar{y}_i| . \tag{3.7}$$

The speed term is designed in such a way that speed decreases as the distance between the center of gravity of the level set and the point (x,y) is increased. For points having less distance with center of gravity of level set, speed term is high and the level set capture the points. On the other hand if points belongs to other cluster then speed term is made negative so that level set pushes itself away from the points. For the points, which are yet to be clustered, competition goes on between level sets based on their distance from the center of gravity. The speed term is dynamic with respect to time as center of gravity for the level set changes with time.

In most level set implementation, narrow band curve evolution is incorporated to gain computational efficiency [6]. In narrow band implementation, we need to calculate values only in the small neighborhood of zero level set curve and not on the entire space. We are taking small neighborhood into account through small quantity ε. To incorporate narrow band level set curve evolution we can modify our evolution equation as

$$\frac{\partial \Phi_i}{\partial t} = \delta_\varepsilon(\Phi_i)[\alpha\kappa_i |\nabla\Phi_i| + \beta V_i |\nabla\Phi_i|], \quad \forall i \tag{3.8}$$

where $\delta_\varepsilon(\Phi_i) = \frac{d}{d\Phi_i} H_\varepsilon(\Phi_i)$ and $\varepsilon \to 0$.

Multiplication with the term $\delta_\varepsilon(\Phi_i)$ ensures the updation of values of function Φ_i only in the small neighborhood of zero level set curve.

3.2 Initialization of the level set

We do seed initialization of the level set. Initialization should be proper so that different level sets can track different clusters in the feature space. In case of 2 clusters one way is to randomly pick some point and chose second point farthest from this point. We initialize 2 level sets, which encompass chosen points. Gradually, level sets grow and encompass other points belonging to same class.

3.3 Stopping criteria

The curve evolution stops automatically when all the points in feature space are classified. This is because when every point belongs to one of the level set then the points exert negative forces on other level sets so that other level sets never try to acquire points belonging to other clusters and stops evolving.

3.4 Analysis of Model

When points are separated well and forms very distinct clusters, our proposed method captures the clusters nicely. In some situations, points of different classes are not well clustered. In other words, the points of different classes are overlapping and we want minimum number of misclassification. In our proposed method the level set whose center of gravity is having minimum distance with the points capture the points. There is always a competition between different level sets to capture the points but once a level set acquires a point every other level set tries to push itself away from the point.

In our proposed method we assume that number of classes is known a priori. Now, let us investigate the problem where number of classes is not known a priori. When some set of points are not very near to any of the evolving level sets then a decision has to be made whether the set of points should go to nearest cluster or a new cluster has to be formed.

We are proposing a heuristic to decide whether a point should belong to existing cluster or a new cluster. We are defining the index for clustering tendency as follows.

Let P be the nearest outside point to the evolving level set curve C, the distance of point P with the evolving level set curve is d. We take a point Q inside the level set whose distance from the curve is minimum and let the distance be s. In Figure 3.1, points P and Q are shown. Point P is yet to be clustered.

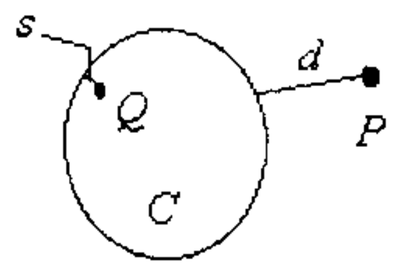


Figure 3.1: C is the evolving curve. P and Q are two points inside and outside the curve C.

Then index for clustering tendency for the evolving level set is defined as

$$p = \frac{d}{d+s} \tag{3.9}$$

If the index p for a level set curve is greater than threshold (say 0.9) then we stop the evolution of the curve otherwise we evolve the level set curve to capture point P. In other words, when the distance of nearest point outside the evolving level set is 9 times greater than the distance of nearest point inside the level set, we stop evolving the curve. We say the clustering tendency test is true when p is greater than threshold (say 0.9) otherwise false. The clustering

tendency test is performed in every iteration for every level set. If number of level set initialized is less than number of classes, then after some iteration clustering tendency test becomes true for all the level sets and level sets stops evolving. But some points still remain to be clustered. At this point we need to initialize new level set to capture the un-clustered points. Now, we will discuss about the initialization of new level set.

We can randomly choose any un-clustered point. The new level set is initialized around the randomly chosen point.

Now, the new zero level set curve captures some set of points, which belongs to new class. If this new level set also does not cluster all the remaining points and satisfies the clustering tendency criteria then again on the same line a new level set is initialized. So this method can be seen as a divide and conquer approach in which new level set curves are initialized when all the existing level set curves stops evolving on satisfying the clustering tendency criteria.

We have shown results for supporting our heuristic on synthetic image having three classes (Figure 4.17). Initially, we have initialized only two curves but third level set curve is automatically initialized and captured the third class correctly.

In next section, we are describing the overall algorithm for segmenting different regions in the image.

3.5 Algorithm

Input:

1. Image to be segmented
2. Corresponding feature space represented as binary image
3. Number of classes k

Algorithm:

1. Initialize k zero level set curves as described in section 3.2.
2. Repeat until all points in the feature space are clustered.
3. Update the zero level set curves by equation (3.8).
4. Check for clustering tendency for every zero level set curve given by equation (3.9).
5. If clustering tendency test is true (p is greater than threshold) for some level set Φ_i, then stop evolving Φ_i.
6. If clustering tendency test is true for all zero level set curves and some feature points still remain to be clustered then randomly choose an un-clustered point. Initialize new zero level set curve around the randomly chosen point.
7. Continue with step 2.

Output:

1. Set of clustered points.
2. Corresponding segmented image.

Chapter 4

Results and Discussion

In this chapter we present some numerical results for the application of the proposed model to various synthetic and real images. In the experiments we have taken the value of parameter α of equation (3.1.8) as 1. The value of parameter β of equation (3.1.8) is not same for all the experiments. Generally, value of β is kept high, typically in the range of 10^2 to 10^5, so that the level set could capture the points, which are not near to the zero level set curve.

Initially, we are showing results on synthetic images. Synthetic image represents the feature space. We are evolving zero level set curve on the synthetic images representing feature space. Figure 4.1(a) shows a synthesized example of feature space having two classes, which are well separated. Also, Figure 4.1(a) shows two initial zero level set curves, one in red and another in green, initialized as described in section 3.2. The value of parameter β is taken as 100. Gradually, two level sets capture the two classes as shown in Figure 4.1(b)(c)(d). The zero level sets stops evolving when all the points are clustered in one of the two level sets.

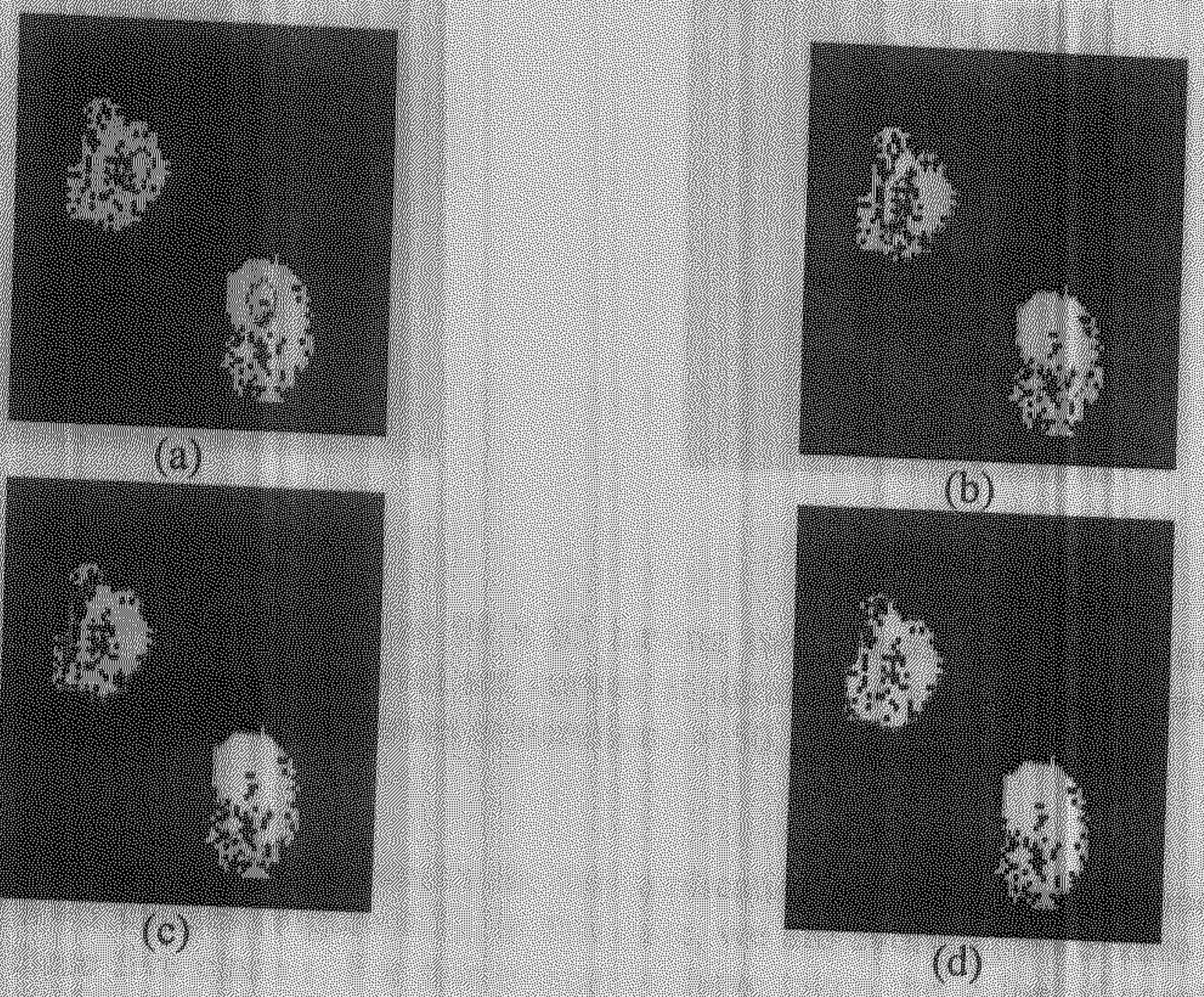


Figure 4.1: (a) Binary image of size 75 x 80 having two classes and two initial zero level set curves (red and green) (b) zero level set curves at iteration $t = 20$ (c) zero level set curves at iteration $t = 50$ (d) zero level set curves at iteration $t = 100$.

Figure 4.2(a) shows another synthetic image having two classes along with two initial zero level set curves initialized as described in section 3.2. Figure 4.2(b)(c)(d) shows the evolution of two curves with time. The value of parameter β is 100. Figure 4.2(d) shows the two zero level set curves capture the two classes correctly.

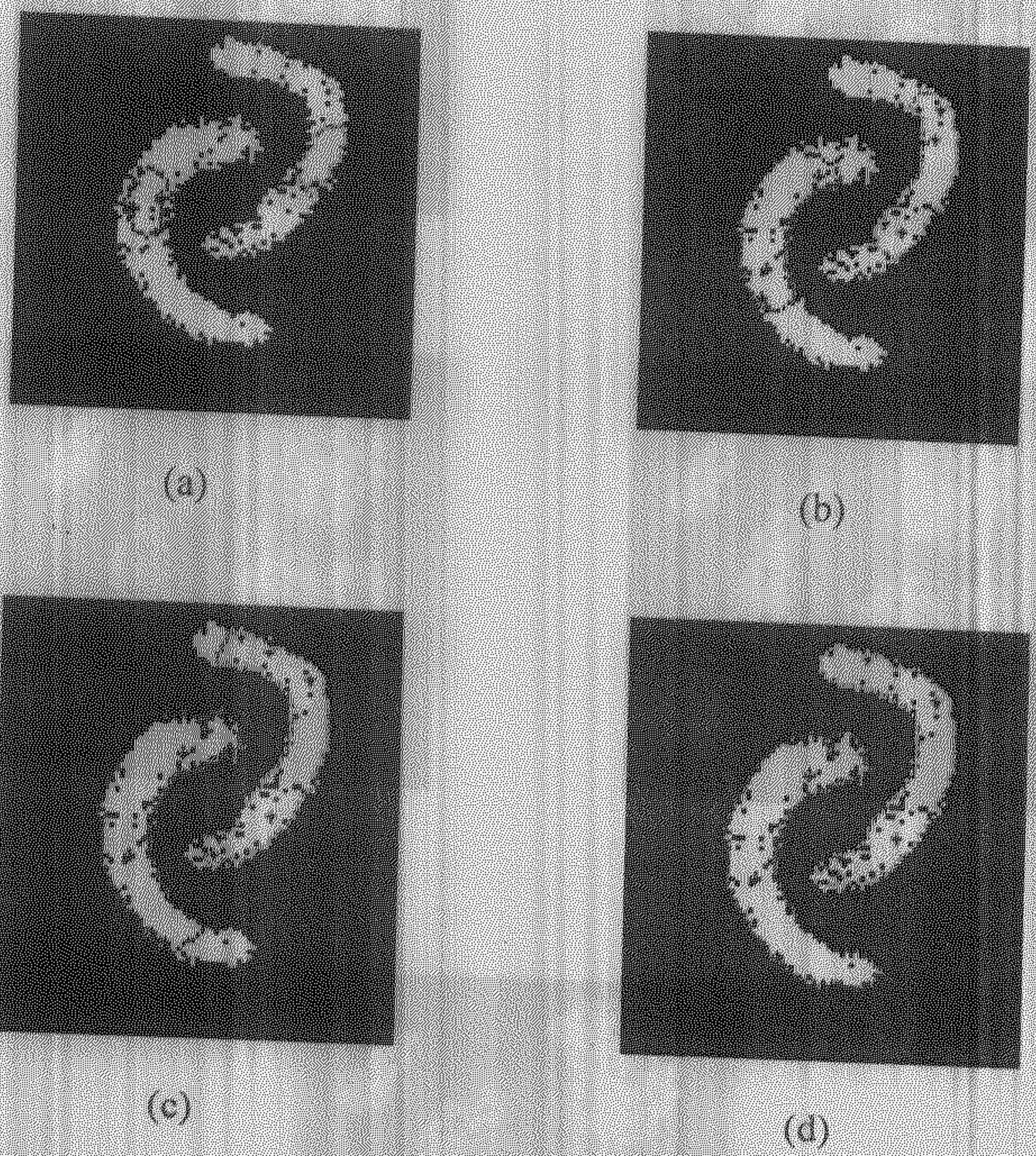
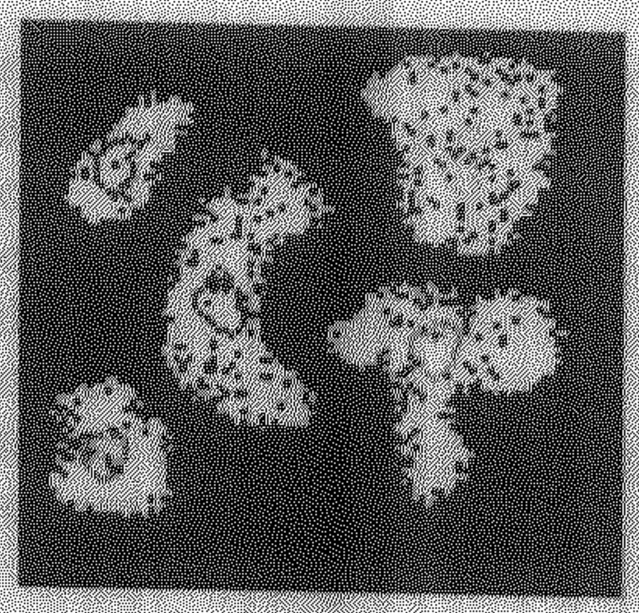
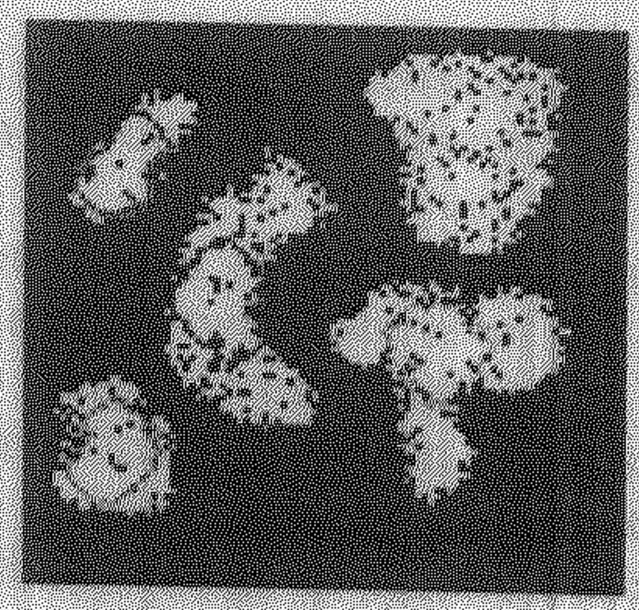


Figure 4.2: (a) Binary image of size 95 x 85 having two classes and two initial zero level set curves (red and green) (b) zero level set curves at iteration $t = 15$ (c) zero level set curves at iteration $t = 30$ (d) zero level set curves at iteration $t = 50$.

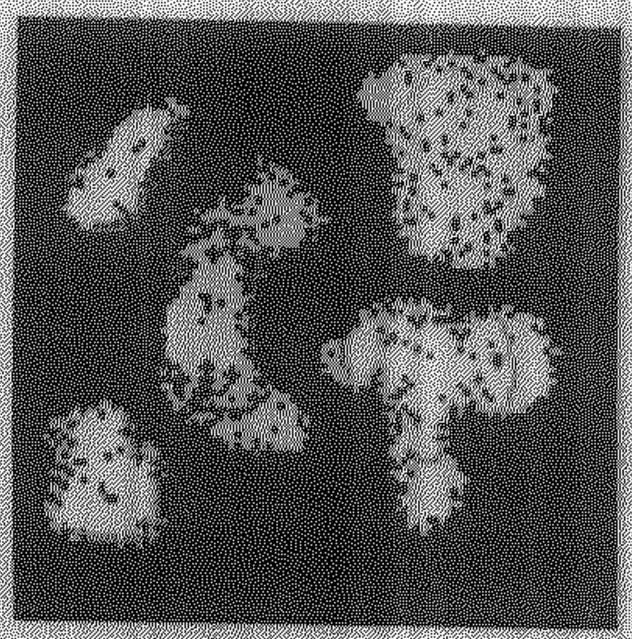
Next, we are showing the results for the application of our proposed model on synthetic image having $n = 5$ classes (Figure 4.3). In this case, we are evolving $n = 5$ zero level set curves for capturing these five classes. For this image we have taken the value of parameter β as 200. The five zero level set curves are represented by five different colors. Figure 4.3(e) shows each curve encompasses all the points, which belongs to one class, correctly.



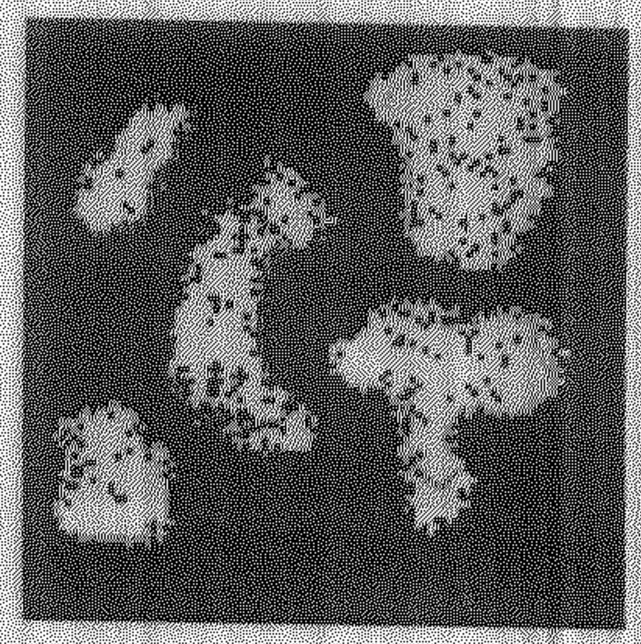
(a)



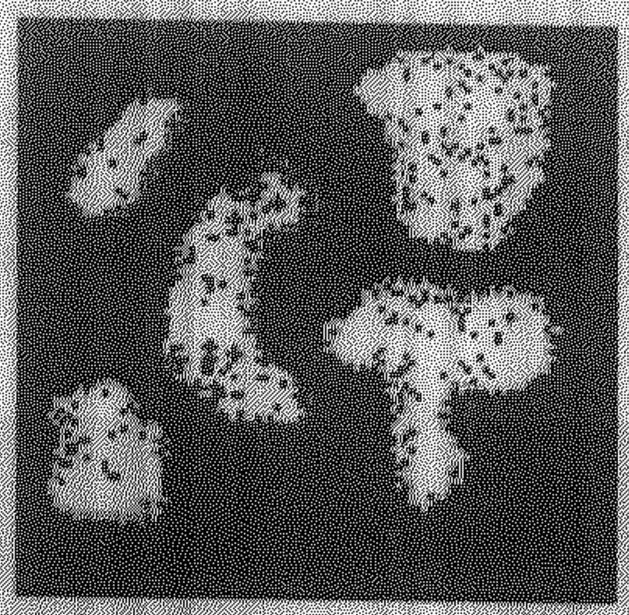
(b)



(c)



(d)



(e)

Figure 4.3: (a) Binary image of size 100 x 100 having five classes and five initial zero level set curves initialized properly (red, green, blue, cyan and yellow) (b) zero level set curves at iteration $t = 8$ (c) zero level set curves at iteration $t = 15$ (d) zero level set curves at iteration $t = 25$ (e) zero level set curves at iteration $t = 58$.

After showing results on synthetic images, now we will discuss results on real images. For each image, we also show the misclassification by our proposed method against the manual segmentation.

Since we are evolving the zero level set curves in feature domain, first we have to transform the image into feature space. So, we extract two features of the image and plot the pixels of image in feature space to get the binary image representing feature space. Figure 4.4(a) shows a satellite image having two regions. The two features chosen to transform the image into feature space are green channel information and average intensity of three-channel information (red, green and blue). Figure 4.4(b) shows the binary image of feature space. x-axis represents the green channel information feature and y-axis represents the average intensity of three-channel information feature.

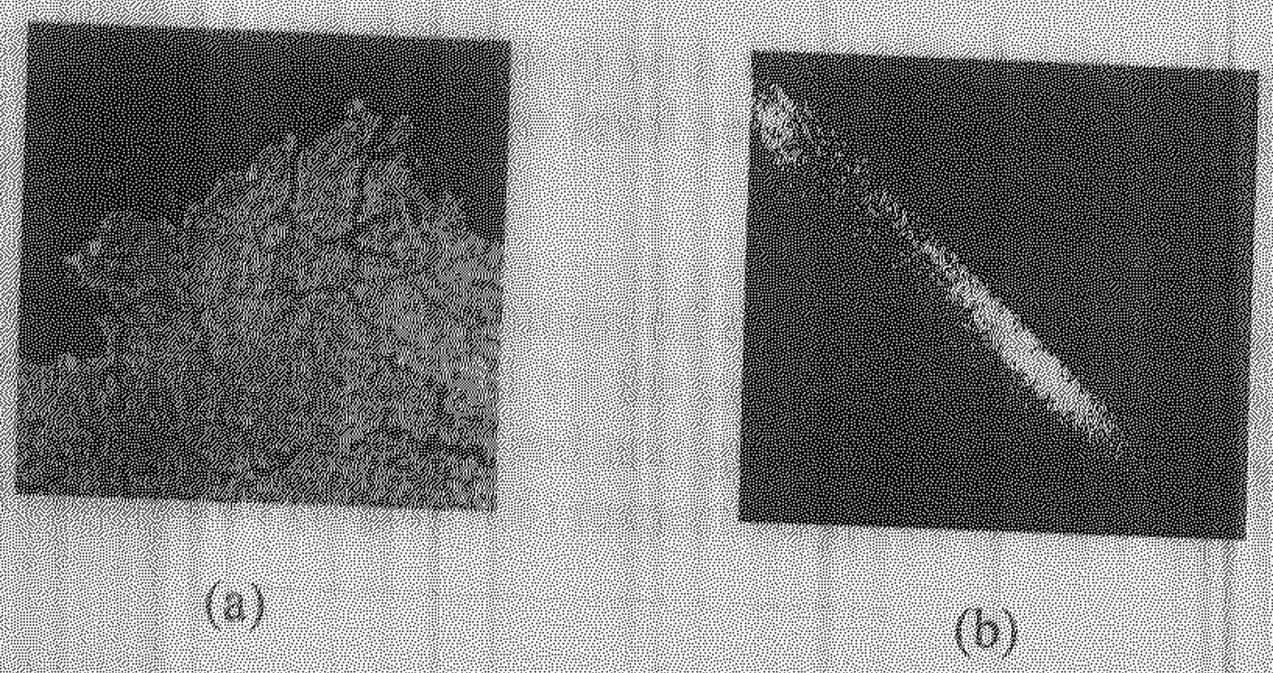
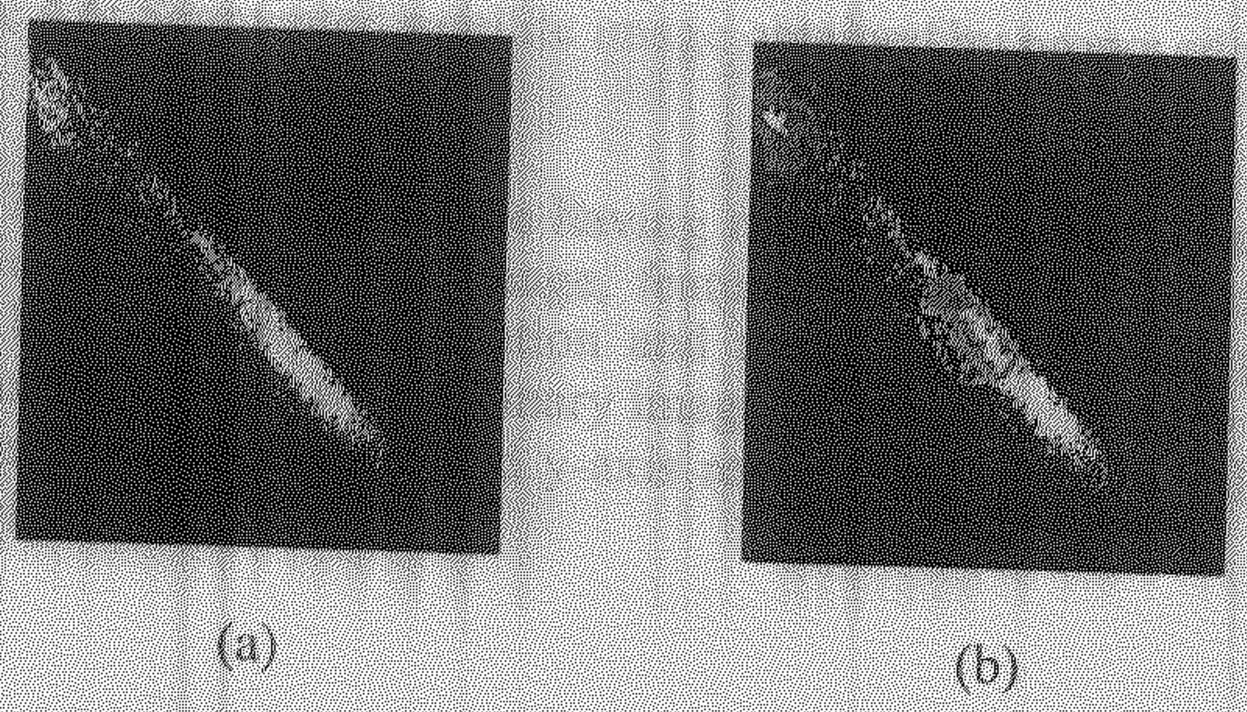
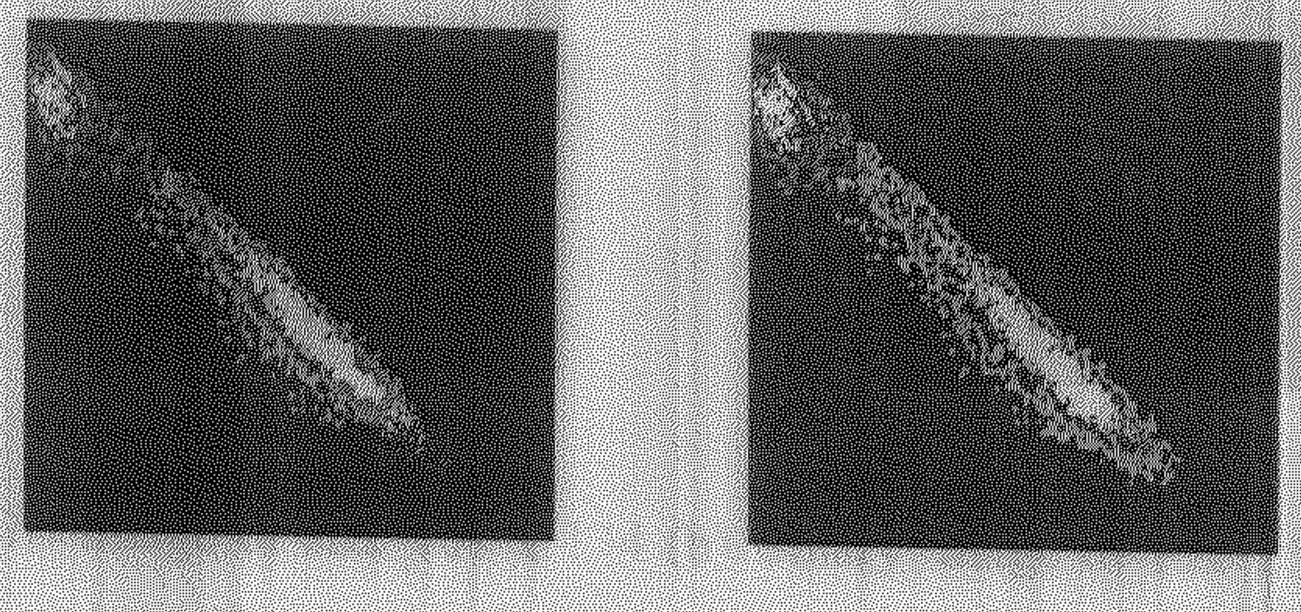


Figure 4.4: (a) Satellite image (RGB) of size 120 x 120 having two regions (b) Binary image of size 256 x 256 representing feature space with x-axis as green channel feature and y-axis as average intensity of three-channel feature.

Figure 4.4(b) is of size 256 x 256 because range for x-axis and y-axis information is from 0 to 255. Now, two zero level set curves are evolved on Figure 4.4(b) and the results are shown in Figure 4.5. Since, the two classes are not well separated, there is competition between the level sets as one can see in Figure 4.5.





(b) (d)

Figure 4.5: (a) Same image as Figure 4.4(b) with two initial level sets (red and green) (b) zero level set curves at iteration $t = 8$ (c) zero level set curves at iteration $t = 15$ (d) zero level set curves at iteration $t = 30$.

After the convergence of the evolution of zero level set curves, we classify the original satellite image based on points classified by the zero level set curves. The segmented image is shown in Figure 4.6. The portion under red and green zero level set curves (Figure 4.5(d)) are shown in white and black color (Figure 4.6) respectively.

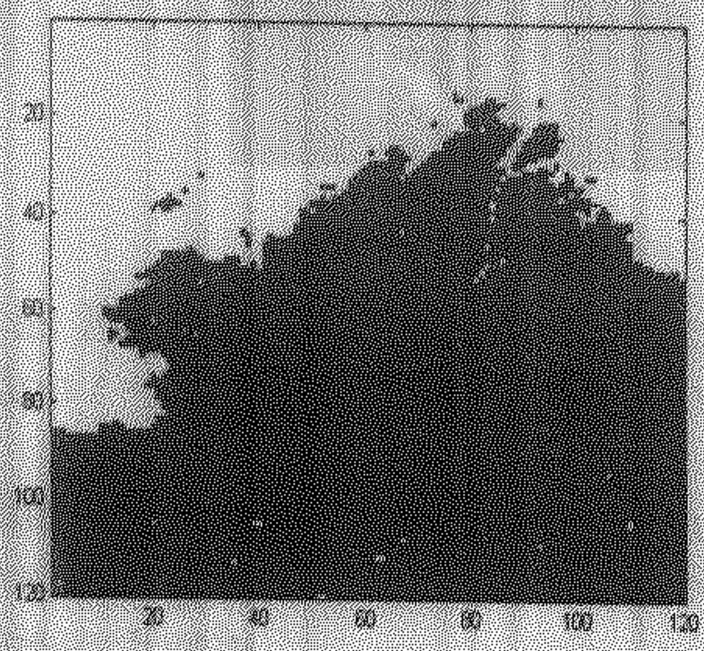
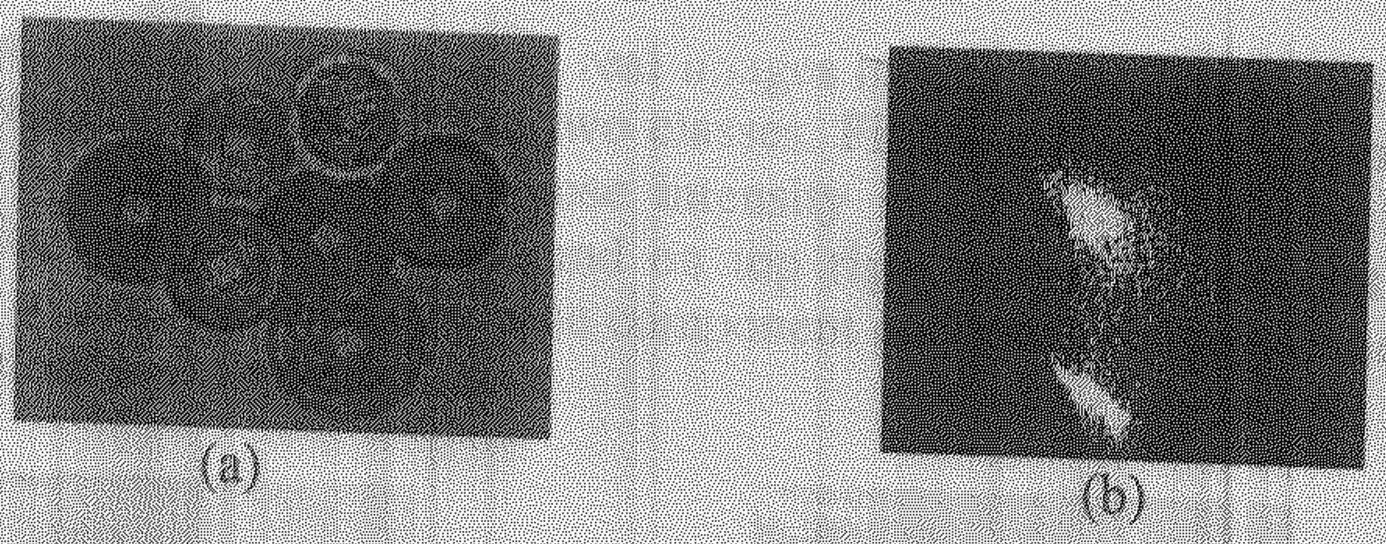


Figure 4.6: Segmented satellite image obtained by proposed method.

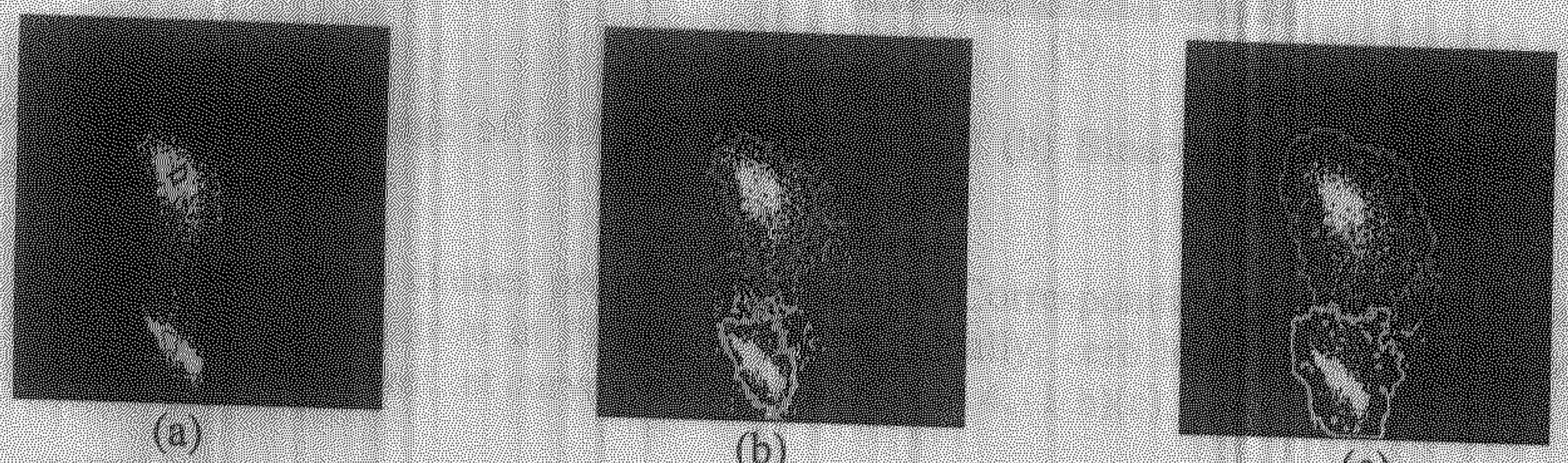
In Figure 4.7(a) we have an image having round objects and a background. Clearly, this image also has two classes. For this image, features selected for transforming the image into feature space are blue channel information and mean value of each pixel around its 3 x 3 neighborhood. Figure 4.7(b) shows the binary image representing feature space. x-axis represents the blue channel information feature and y-axis represents the mean value feature around each pixel in its 3 x 3 neighborhood.



(a) (b)

Figure 4.7: (a) RGB image of size 101 x 134 (b) Binary image of size 256 x 256 representing feature space.

The two zero level set curves are evolved on Figure 4.7(b). The evolution of level set curves is shown in Figure 4.8. When the two zero level sets encompass all the points the evolution of curve stops.



(a) (b) (c)

Figure 4.8: (a) Same image as Figure 4.7(b) with two initial zero level sets (red and green) (b) zero level set curves at iteration $t = 20$ (c) zero level set curves at iteration $t = 40$.

Figure 4.9 displays the segmented image for Figure 4.7(a) obtained by the proposed method. The portion under red and green zero level set curves (Figure 4.8(c)) are shown in white and gray color (Figure 4.9) respectively.

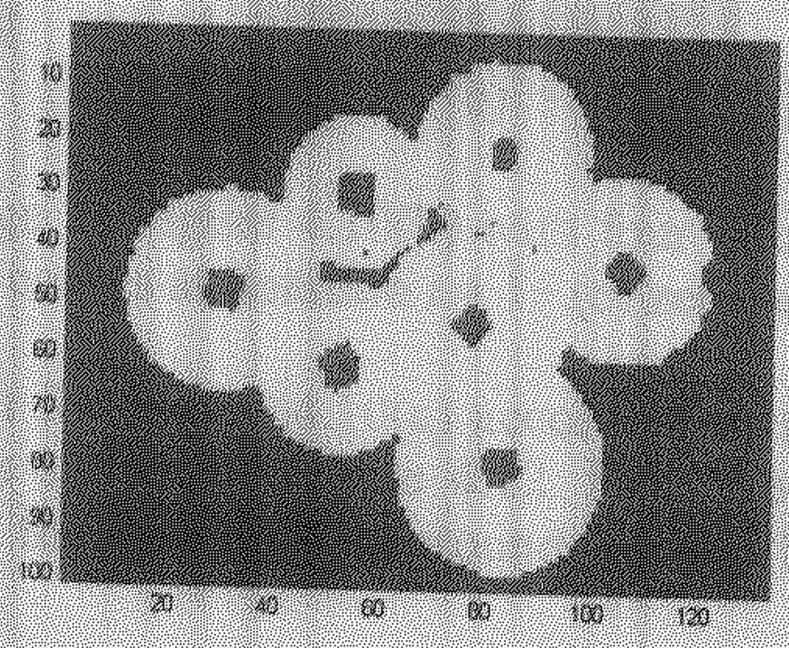


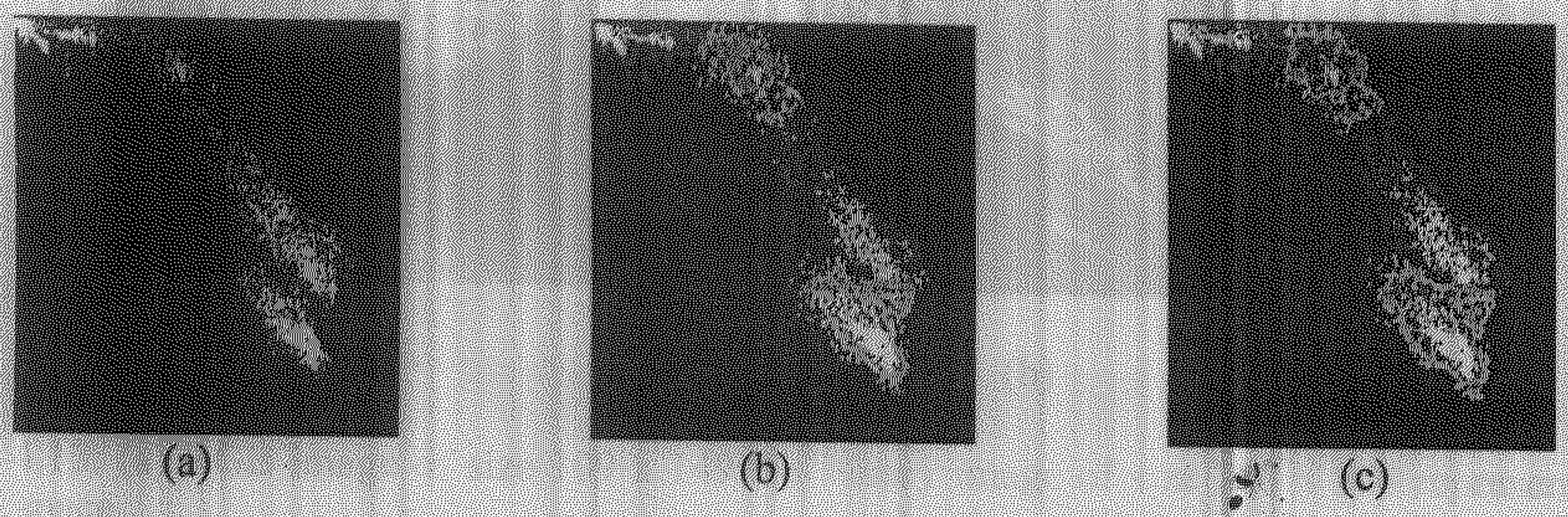
Figure 4.9: Segmented image by proposed method.

Till now we have shown results for real images having two classes. Now, we will discuss results on real images having more than two classes. Figure 4.10(a) shows image of a child. In this image child is wearing a dress whose intensity is different from its face. Also, image contains two backgrounds. The background on top right corner is different from the general background. Here again, we are taking blue component and mean value around each pixel in 3 x 3 neighborhood as our two features.

(a) (b)

Figure 4.10: (a) RGB image of child of size 130 x 132 (b) Binary image of size 256 x 256 representing feature space.

We are assuming 4 classes present in the image. Correspondingly, we initialize our four zero level set curves as shown in Figure 4.11(a). Gradually, zero level set curves evolve with time. Evolution of zero level set curves is shown in Figure 4.11(b)(c).



(a) (b) (c)

Figure 4.11: (a) Same image as Figure 4.10(b) with four initial zero level sets (red, green, blue and cyan) (b) zero level set curves at iteration $t = 15$ (c) zero level set curves at iteration $t = 35$.

The segmented result for Figure 4.10(a) according to final convergence of zero level set curves in Figure 4.11(c) is shown in Figure 4.12. Red, green, blue and cyan level set curve (Figure 4.11(c)) represents the general background, second background, face and dress of the child respectively.

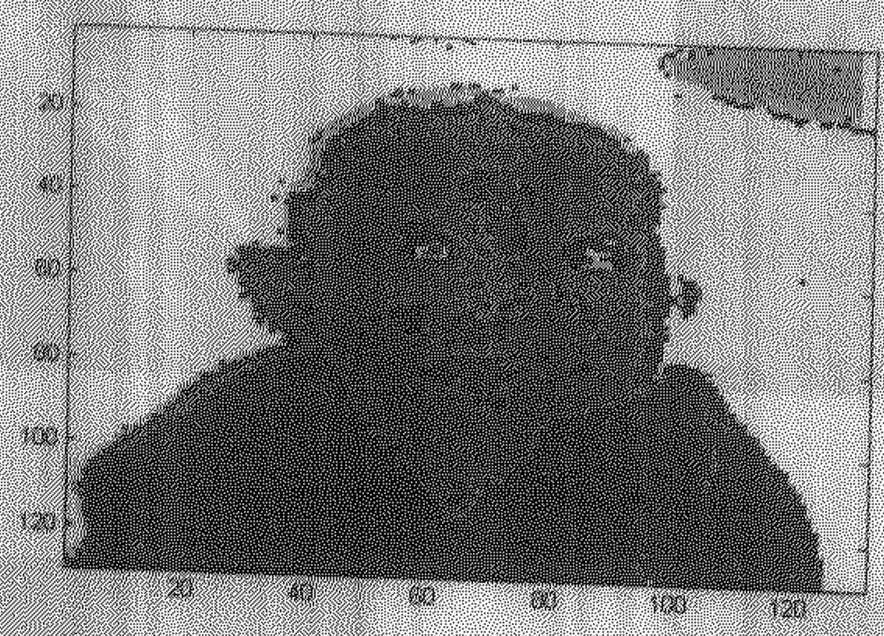
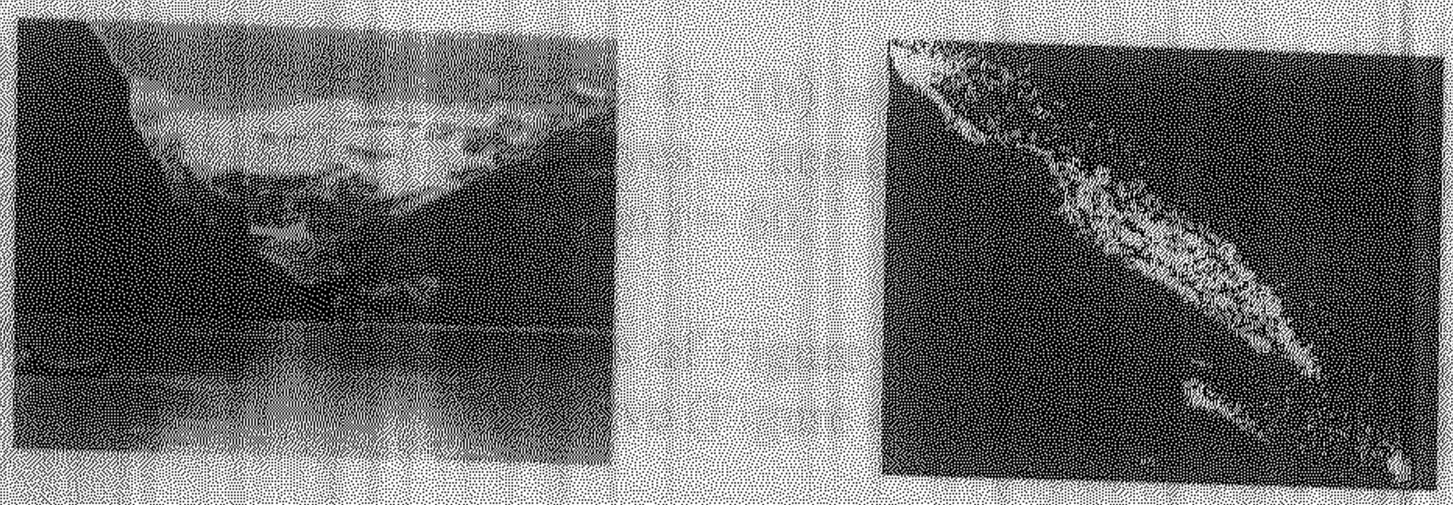


Figure 4.12: Segmented image by proposed method.

The proposed method could clearly segment face, dress and background regions. The two backgrounds come out to be different in Figure 4.12. Some part of the face is treated as the dress part because final cyan and blue level sets (Figure 4.11(c)) do not converges correctly.

Consider another image shown in Figure 4.13(a). Figure 4.13(b) shows the binary image representing feature space with x-axis having feature as blue channel information and y-axis having feature as mean values of each pixel around 3 x 3 neighborhood.



(a) (b)

Figure 4.13: (a) RGB image of size 109 x 149 (b) Binary image of size 256 x 256 representing feature space.

For this image we are assuming five classes and evolving five zero level set curves as shown in Figure 4.14.

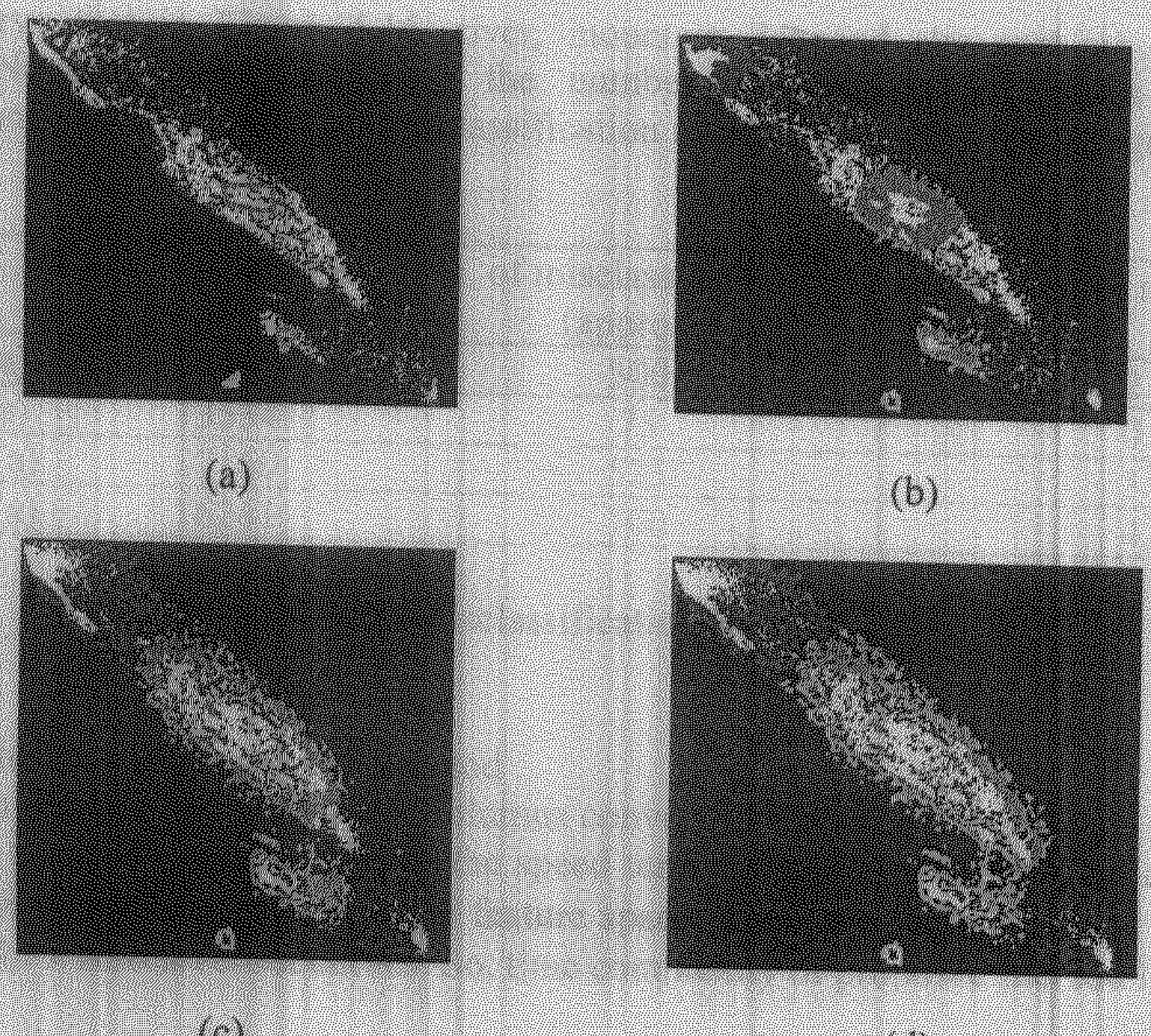


Figure 4.14: (a) Same image as Figure 4.13(b) with five initial zero level sets (red, green, blue, cyan and yellow) (b) zero level set curves at iteration $t = 11$ (c) zero level set curves at iteration $t = 33$ (d) zero level set curves at iteration $t = 60$.

Corresponding segmented image of Figure 4.13(a) using 4.14(d) is shown in Figure 4.14. Red, green, blue and cyan level set curve (Figure 4.14(d)) represents mountain, river, ice and sky respectively.

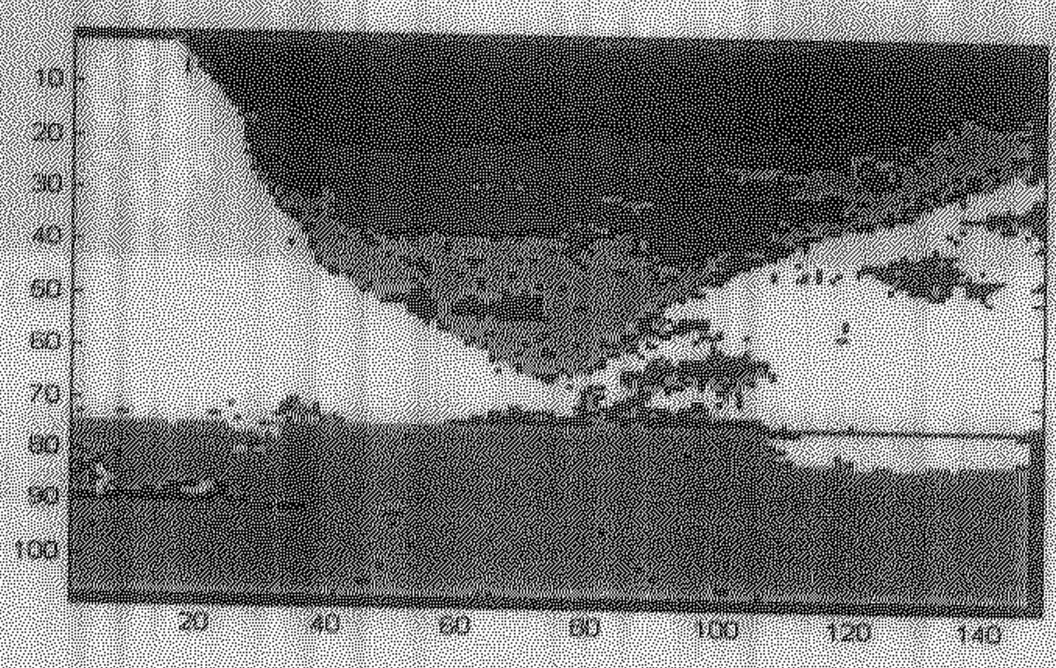


Figure 4.15: Segmented image obtained by proposed method.

Observe that in Figure 4.15, river, mountain, ice and sky part are separated almost correctly. Now, we are describing the number of misclassified pixels against manual segmentation. Table 4.1 gives the misclassification results on the four real images.

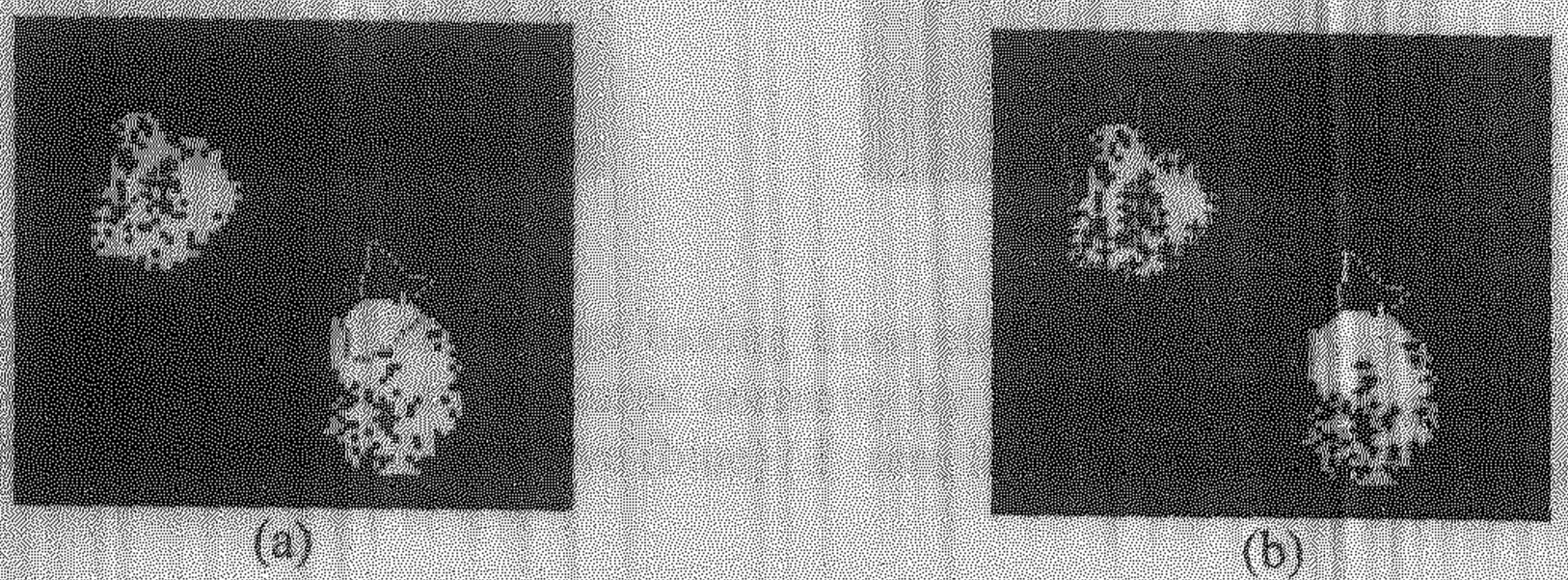
Image	Total no. of pixels	No. of misclassified pixels (approx.)	Percentage of misclassification
Figure 4.4(a)	14400	300	2.08
Figure 4.7(a)	13534	75	0.56
Figure 4.10(a)	17160	525	3.05
Figure 4.13(a)	16241	700	4.31

Table 4.1: Misclassification Results.

Time Complexity of the proposed method

For an image of m rows and n columns conversion between image space and feature space takes $O(mn)$ time. We need to update the speed term for all the zero level set curves with time. There are mn numbers of points in the feature space. Then, we require $O(kmn)$ time to calculate speed term for k zero level set curves. So, time complexity of proposed method is $O(kmn)$.

Figure 4.16 shows the effect when length-shortening term of equation (3.1.8) is not incorporated. In this case, the contour is initialized as shown in Figure 4.16(a). After convergence final
contour doesn't tightly bound the cluster of binds because of absence of length shortening term.



(a) (b)

Figure 4.16: (a) Binary synthetic image of size 75 x 80 having two classes and two initial zero level set curves (red and green) (b) zero level set curves after convergence.

As we have discussed in last section of Chapter 3, we proposed a heuristic for clustering tendency. This heuristic is useful when number of classes is not known a priori. Figure 4.17 shows the results of applying the heuristic given in equation (3.1.9) on synthetic image. Initially, two zero level set curves are initialized shown in red and green in Figure 4.17(a). When the curves captures the two classes correctly, the test for clustering tendency becomes true for both the zero level set curves, hence, a new zero level set curve is automatically initialized at some random un-clustered point as shown in Figure 4.17(c). The third zero level set (shown in blue) correctly clusters the points (Figure 4.17(d)), which belongs to third class.

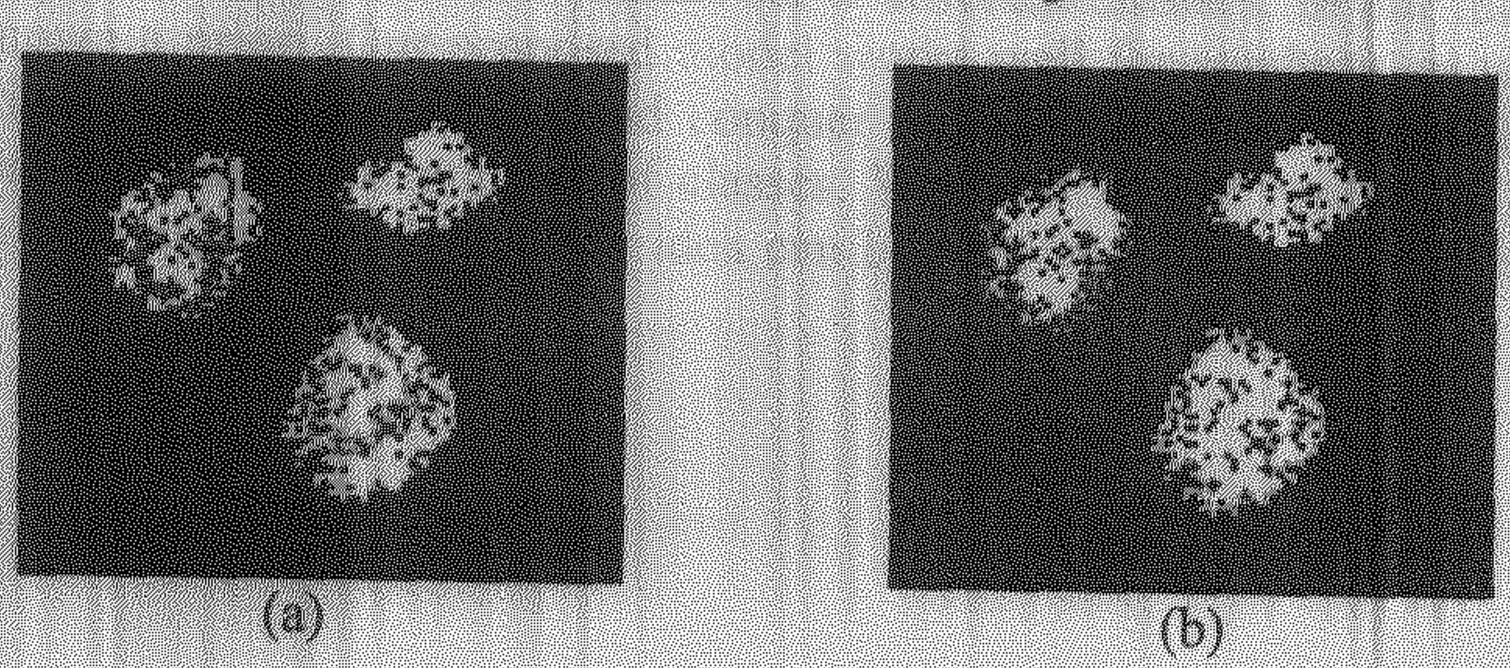


Figure 4.17: (a) Binary synthetic image having three classes but only two zero level set curves are initialized (b) two classes are captured correctly by two zero level set curves (c) Automatic initialization of third zero level set curve (blue) (d) zero level set curves after convergence.

Chapter 5

Conclusion

We have extended the level set theory for image clustering by treating the problem of segmenting multiple regions as a clustering problem in feature space. We have proposed new generalized approach for n curve evolution for n different clusters. In our model the number of classes is known a priori. Our proposed model works well for synthetic as well as real images. Also, we tried to develop a divide and conquer model based on a heuristic, which does not require priori information of number of classes. We have shown results on synthetic image to describe the idea of our heuristic. This idea can be explored further for further research in this area.

References

[1] M. Kass, A. Witkin, and D. Terzopoulos, "*Snakes: active contour models*," International Journal Computer Vision, vol. 1, no. 4, pp. 321-331, 1987.

[2] V. Casselles, F.Catte, T. Coll, and F.Dibos, "*A geometric model for active contours*," Numerische Mathematik, vol. 66, pp. 1-31, 1993.

[3] R. Malladi, J. A. Sethian, and B. C. Vemuri, "*Shape modelling with front propagation: a level set approach*," IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 17, no. 2, pp. 158-175, 1995.

[4] G. Sapiro and A. Tannenbaum, "*Affine invariant scale space*," International Journal Computer Vision, vol. 11, no. 1, pp. 25-44, 1993.

[5] S. Osher and J. A. Sethian, "*Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*," Journal Computational Physics, vol. 79, pp. 12-49, 1988.

[6] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science.* Cambridge, UK: Cambridge University Press, 2nd edition, 1999.

[7] T. F. Chan and L. A. Vese, "*Active Contours without Edges*," IEEE Transactions on Image Processing, vol. 10, no. 2, pp. 266-275, 2001.

[8] S. Tiwari and S. Agrawal, "*Active Contour Model for Multiple Diffused Object Segmentation*," Proceedings of the fifth International Conference on Advances in Pattern Recognition ICAPR, Kolkata, India, pp. 369-374, December 2003.

[9] A. Yezzi, A. Tsai and A. Willsky, "*A Statistical Approach to Snakes for Bimodal and Trimodal Imagery*," Proceedings of the International Conference on Computer Vision, vol. 2, no. 2, pp. 898, 1999.

[10] A Tsai, "*Curve Evolution and Estimation-Theoretic Techniques for Image Processing*", MIT Ph.D. Thesis, August 2000.

[11] A. Gilles and Kornprobst Pierre, *Mathematical Problems in Image Processing*. New York, Springer-Verlag, 2002.

[12] Duda R. O., Hart P. E., Stork D. G., *Pattern Classification*. New York, John Wiley, 2nd edition, 2000

[13] A. Rosenfield, *Digital Picture Processing*. New York, Academic Press, 2nd edition 1982.

[14] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge, Cambridge University Press, 2001.

[15] M. A. Grayson, "*Shortening embedded curves*," Annals of Mathematics, vol. 129, pp. 71-111, 1989.