國 立 交 通 大 學

資訊科學系

碩 士 論 文

等 位 函 數 法 在 影 像 切 割 之 研 究

Research on Image Segmentation Using Level Set Methods

研 究 生：何昌憲

指導教授：薛元澤　教授

中 華 民 國 九 十 四 年 六 月

等 位 函 數 法 在 影 像 切 割 之 研 究
Research On Image Segmentation Using Level Set Methods

研 究 生：何昌憲　　Student：Chang-Xian Ho

指導教授：薛元澤　　Advisor：Yuang-Cheh Hsueh

國 立 交 通 大 學
資 訊 科 學 系
碩 士 論 文

A Thesis

Submitted to Institute of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

等位函數法在影像切割之研究

學生：何昌憲　　　　　　　　　　　　　　　　　　　　指導教授：薛元澤

國立交通大學資訊科學學系（研究所）碩士班

摘要

近年來，隨著電腦科技的發展，數位化影像的處理與應用漸趨重要。在許多應用之前，影像切割是一個重要的基本處理。因此影像切割成爲影像處理上的一個重要課題。本研究的主要目的是研究使用等位函數法 (Level Set Methods) 的動態曲線 (Active Contour) 影像切割法。等位函數法是一個新穎的數值方法，可以很容易的表示出曲線或曲面。同時也具有容易處理偏微分運算的特性。我們整理和實做使用等位函數法的一些基本且重要的影像切割法，並討論他們在實際運作上會遇到的問題及在影像切割上的優缺點。

Research on Image Segmentation Using Level Set Methods

student : Chang-Xian Ho Advisor : Dr. Yuang-Cheh Hsueh

Department (Institute) of Computer and Information Science

National Chiao Tung University

ABSTRACT

In recent years, with the development of computer technology, digital image processing and its applications are more and more important. Image segmentation is an important and basic processing before other higher level applications. Image Segmentation has become one of the major subjects in image processing. The purpose of this thesis is to explore the image segmentation using Level Set Methods. Level Set Methods are novel numerical methods, they express curves or surfaces easily and have the convenient feature of dealing with PDE calculations. We implement several basic and important methods of image segmentation using level set methods. Then discuss the main problems encountered in practice and their pros. and cons. on image segmentation.

誌　　謝

首先要感謝的是我的指導教授 薛元澤教授, 在這兩年研究所生涯中於課業及研究上的悉心指導與諄諄教誨, 不僅讓我訓練獨立思考與研究之能力, 更讓我深刻體會求學處世的道理, 使本論文得以順利完成, 謹此致上最誠摯的謝意。

亦感謝實驗室學長聖博、永靖於研究上的寶貴建議；同窗逢軒、蕙綾、薇婷於課業上的討論與研究；學弟妹盈賢、仲庭、裕泉、佩君、慧縈、明志等使沉悶的實驗室充滿歡樂。此外特別感謝在外一起追求理想的同鄉摯友們的陪伴與鼓勵。

最後, 感謝我最敬愛的父母及家人, 多年來對我含辛茹苦的栽培與無怨無悔的付出, 使我能在安穩的生活中求學。謹將此篇論文獻給他們。

何昌憲

謹誌於 國立交通大學資訊科學研究所

中華民國九十四年六月

CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Motivation

With the develop of computer and digital equipment, image processing has become one of major subjects in developing computer vision. Segmentation is an important technique used in image processing to identify objects in an image. Many different methods are used to partition an image into meaningful regions. One of these methods is to use an active contour to deform its shape and lactation according to the relative data from the image. The idea behind active contours for image segmentation is quite simple. We specify an initial guess for the contour, which is then moved by image driven forces to the boundaries of the desired objects. And recently, the level set methods are used to implement the active contours, based on various image analysis methods. Since the level set methods are novel and known by few people and its applications on the image processing are also developing, it is interesting to know how the method is and processing segmentation differ to well-known methods. Our motivation is to organize several basic and important methods of image segmentation using level set methods, and gives better understanding of their pros, cons, improvement between each other, and defects of these methods.

1.2 Related Works

There are two deformable models. The traditional one is the Parametric Active Contours (PACs), also called snakes, which trace the contours by discretizing them to finite number marker points. The PACs restricts the degree of topological adaptability, especially if the deformation involves splitting or merging of parts. In contrast, the Geometric Active Contours

(GACs) utilizes the level set methods introduced by Osher and Sethian [1][2] recently.

The level set methods express the curve or surface as an implicit function, the topology changes naturally without any other dealing, and it is easy to calculate local properties. They give a smooth curve expression, and can be used in 3D models and simulation nature phenomena.

There are three forces considered, the curvature, defined from the curve itself, are designed to keep the curve smooth during the deformation process, the normal direction constant force which are used to decide the expanding or shrinking of a curve and force from external felicity field, which is computed from the underlying image data, are defined to move the curve toward an object boundary or other desired features within the image.

1.3 Organization of Thesis

In chapter 2 we introduce the basic level set methods. We first introduce the basic idea of level set methods and the function of three forces. Then we discuss each force and there effect of curve evolution. In chapter 3 we introduce our main interest of using level set methods on image segmentations. We first introduce basic method from snake followed by fixed form in level set methods of Geodesic Active Contour (GAC) [3]. These two method generate small capture vector fields which point to the boundary of objects. The extension method is Gradient Vector Flow (GVF) [4], it improves the capture ability of the vectors. Different to previously methods which utilize gradients of an image, the Edgeflow, purposed in [5] is based on Gabor filter and Difference of Offset Gaussian to prediction the vector points to the boundary of objects. Finally, we introduce the region stability method [6]. In the chapter 4 we will test these methods on artificial images and to observe the noise effect on these methods.

CHAPTER 2
LEVEL SET METHODS

In this chapter we will introduce the background knowledge of level set methods introduced by Osher and Sethian [1, 2]. Level set methods are simple numerical methods used to solve various PDE problems. Recently, they have been used to many areas, including optimal design, CAD, 3D shape modelling, motion simulation, etc.

We will introduce basic concepts of level sets in sections 1.1 and 1.2, then introduce evolving curve grounds on external velocity, mean curvature and normal direction velocity in section 1.3. Finally, we will discuss the advantages and disadvantage of using level set methods to implement curve evolution.

2.1 Implicit Functions

Consider a unit circle in two dimensions. We generally represent such a curve as a function $x^2+y^2=1$. A convenient way of approximating the representation is to parameterize the curve by discretizing it to a finite set of points s, $s \in [0,1]$, such that each point $\mathbf{x}$ is on the curve, $\mathbf{x} = \vec{v}(s) = (x(s), y(s))$. The curve is denoted as $C(s)\colon [0,1] \to \mathbb{R}^2$.

Alternatively, the idea of level set methods is to represent the interface in $\boldsymbol{R}^n$ in one higher-dimensional function. For example, the zero isocontour of function $\phi(x,y) = x^2+y^2-1$, i.e. $\phi(x,y) = 0$, also expresses the same unit circle as above. Note that ϕ is a three-dimensional function, and the unit circle is a two-dimensional curve.

For an implicit function $\phi(\mathbf{x})$, here $\mathbf{x} = (x_1, \ldots, x_n) \in \boldsymbol{R}^n$, define the interior region an open sets $\Omega^- = \{\,\mathbf{x} \mid \phi(\mathbf{x}) < 0\,\}$, and the exterior region $\Omega^+ = \{\,\mathbf{x} \mid \phi(\mathbf{x}) > 0\,\}$. The contour is defined by $\partial\Omega = \{\,\mathbf{x} \mid \phi(\mathbf{x}) = 0\,\}$.

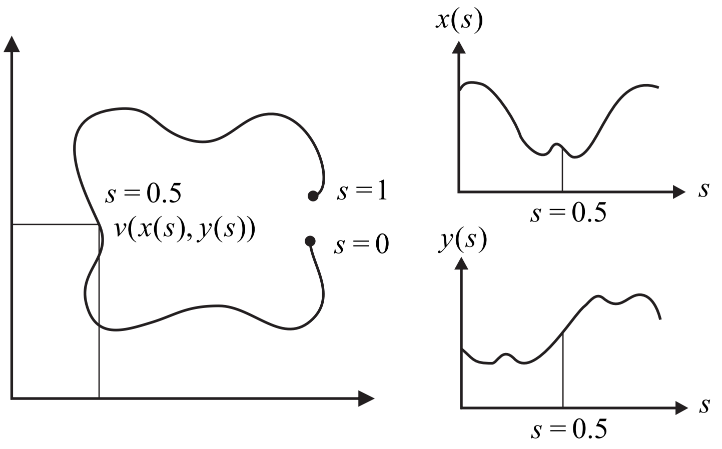


Fig. 2.1 parameter representation

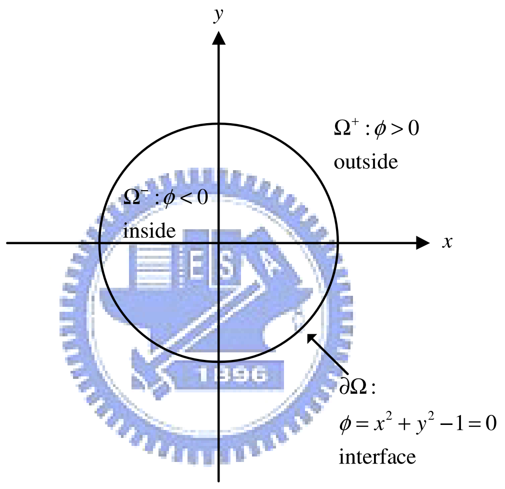


Fig. 2.2 Implicit representation of $x^2 + y^2 = 1$

2.2 Signed Distance Function

Given any interface Γ in $\boldsymbol{R}^n$, a signed distance function is used to generate the underly implicit function in a different way.

A *distance function* $d(\mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_n)$, is defined as

$$d(\mathbf{x}) = \min(|\mathbf{x} - \mathbf{x}_I|) \quad \text{for all} \quad \mathbf{x}_I \in \partial\Omega, \tag{2.1}$$

implying that $d(\mathbf{x}) = 0$ if $\mathbf{x} \in \Gamma$.

A *signed distance function* is an implicit function ϕ with $|\phi(\mathbf{x})| = d(\mathbf{x})$. The $\phi(\mathbf{x})$ is defined as

$$\begin{aligned} \phi(\mathbf{x}) &= -d(\mathbf{x}) && \text{for} \quad \mathbf{x} \in \Omega^- \\ \phi(\mathbf{x}) &= d(\mathbf{x}) && \text{for} \quad \mathbf{x} \in \Omega^+ \\ \phi(\mathbf{x}) &= 0 && \text{for} \quad \mathbf{x} \in \partial\Omega = \Gamma. \end{aligned} \tag{2.2}$$

In our example, we replace the implicit function $\phi(x, y) = x^2 + y^2 - 1$ with the signed distance function $\phi(x, y) = \sqrt{x^2 + y^2} - 1$ in order to represent the unit circle $\partial\Omega = \{\mathbf{x} \mid |\mathbf{x}| = 1\}$.

Between the implicit function and signed distance function, there are several common properties and advantages of geometry operations.

- **Region definition**

  The signed distance function gives the same boundary $\partial\Omega$, interior region Ω^-, and exterior region Ω^+, that $\phi(x, y) = x^2 + y^2 - 1$ does.

- **Local properties of the interface**

  - Gradient and Normal

    Define the gradient as

    $$\nabla\phi = \left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}\right). \tag{2.3}$$

    The gradient $\nabla\phi$ is perpendicular to the isocontours, and points to the direction of increasing ϕ. This indicates that if $\mathbf{x}$ is a point at one isocontour, $\nabla\phi(\mathbf{x})$ has the same direction as normal $\vec{N}$ at $\mathbf{x}$. Thus the *unit normal* is

    $$\vec{N} = \frac{\nabla\phi}{|\nabla\phi|}. \tag{2.4}$$

    Furthermore, the signed distance function has a new property that is

    $$|\nabla\phi| = 1 \quad \text{for all} \quad \mathbf{x} \neq 0. \tag{2.5}$$

– Mean Curvature

The *mean curvature* of the interface is defined as the divergence of the normal $\vec{N}$, that is

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right), \tag{2.6}$$

so that $\kappa > 0$ for *convex* regions, $\kappa < 0$ for *concave* regions, and $\kappa = 0$ for a plane.

- **Unity and Binary operations**

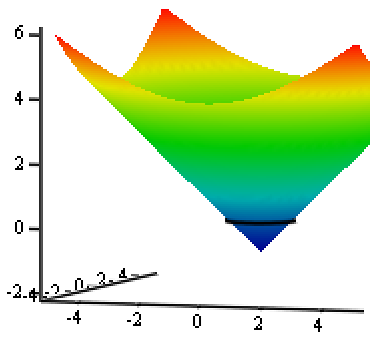
  For an implicit function $\phi(\mathbf{x})$, the *complement* of ϕ can be defined by $\phi^c(\mathbf{x}) = -\phi(\mathbf{x})$. If there are two implicit functions ϕ_1 and ϕ_2, the *union* of the interior regions of ϕ_1 and ϕ_2 can be obtained by $\phi(\mathbf{x}) = \min(\phi_1(\mathbf{x}), \phi_2(\mathbf{x}))$. Similarly, *intersection* of the interior regions of ϕ_1 and ϕ_2 can be obtained by $\phi(\mathbf{x}) = \max(\phi_1(\mathbf{x}), \phi_2(\mathbf{x}))$. It's easy to generate other operations of the interfaces like these do while using implicit functions.

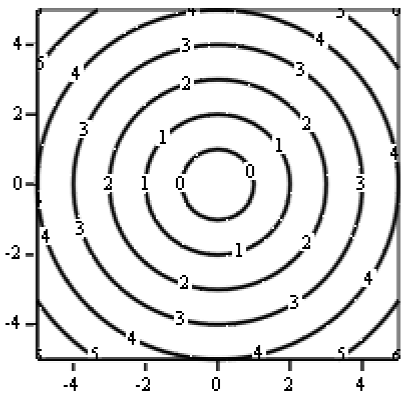
Fig. 2.3 is an example of using level set methods to express curve evolution. Fig. 2.3a is the distance function $\phi(\mathbf{x}) = \sqrt{x^2 + y^2} - 1$, Fig. 2.3b shows the contours of ϕ for each level set, we can see that the zero level set is a unit circle. If we suppose the unit circle evolves under the normal velocity of constant unit speed, after a time step $\triangle t$ the $\phi(\mathbf{x})$ evolves as shown in Fig. 2.3c. Fig. 2.3d shows the contours of Fig. 2.3c, we have seen that now the zero level set is a circle with radius two.

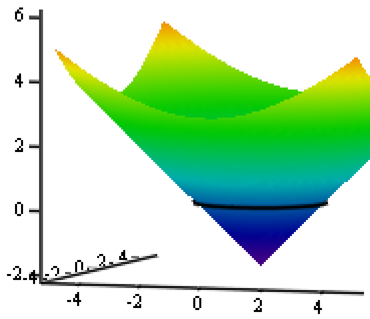
In fact, we can also let the ϕ be positive inside the circle and negative outside the circle, that will turn the cone upside down, and still gives the same zero level set of unit circle. At this situation, $\phi(\mathbf{x}) = 1 - \sqrt{x^2 + y^2}$.

2.3 Motion of Level Sets

The zero level set is generated to the corresponding curve that we are interested in. Then, evolving the $\phi(x, y)$ gives the curve evolution by tracking the zero level set at each time step.


(a) $\phi(\mathbf{x})$ at time t


(b) $\phi(\mathbf{x})$=0 at time t


(c) $\phi(\mathbf{x})$ at time $t + \triangle t$

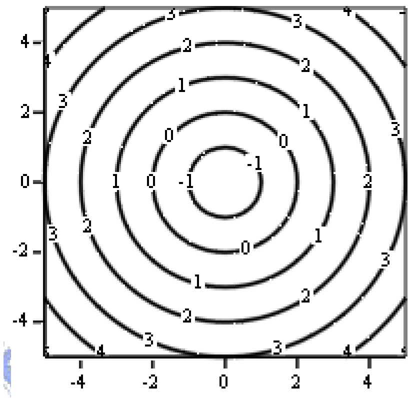

(d) $\phi(\mathbf{x})$=0 at time $t + \triangle t$

Fig. 2.3 Motion of a curve using level set methods

The level set methods can also be used in higher dimensional hyper-surface, for example the three-dimensional surface and thus the ϕ is a four-dimensional function. Since our main goal is to use level set methods to implement curve evolution and image segmentation, we will only discuss two-dimensional curves and three-dimensional signed distance functions.

2.3.1 External Velocity Field

Let $\phi(x, y, t)$ denote $\phi(x, y)$ at time t and $\Gamma(t)$ be the zero level set of $\phi(x, y, t)$. Suppose the external velocity of each point at Γ is given as $\vec{V}(x, y)$. Generating such external velocity filed $\vec{V} = (u, v)$, we wish to move Γ with this velocity. This gives the equation

$$\Gamma_t = \vec{V}(x, y). \tag{2.7}$$

Since $\Gamma(t)$ is the evolving zero level set of $\phi(x, y, t)$, this also gives $\phi(\Gamma(t), t) = 0$. Taking the derivative to t and applying chain rule gives

$$\phi_t + \vec{V} \cdot \nabla\phi = 0, \tag{2.8}$$

this is the basic equation that will be used to introduce other kinds of velocity fields. Since we are working on the images, we can generate cartesian grids with size of images, then the numerical methods are used to solve the ϕ forward in time.

A rather simple first-order accurate method is the *forward Euler* method given by

$$\frac{\phi^{n+1} - \phi^n}{\triangle t} + \vec{V}^n \cdot \nabla\phi^n = 0, \tag{2.9}$$

where $\triangle t$ is the discretized time step, and $t^{n+1} = t^n + \triangle t$, ϕ^{n+1}=$\phi(x, y, t^{n+1})$.

To write equation (2.9) in expanded form as

$$\frac{\phi^{n+1} - \phi^n}{\triangle t} + u^n\phi^n_x + v^n\phi^n_y = 0, \tag{2.10}$$

where (u, v) is the external velocity.

We look at one single point (x_i, y_i) with velocity (u_i, v_i) as example, if $u_i > 0$ the value of $\phi(x_i, y_i)$ moves from left to right, and the *method of characteristics* tells us to look to the left of (x_i, y_i) to determine what value of ϕ will land on the point (x_i, y_i). Similarly, if $u_i < 0$, we look to the right of (x_i, y_i) to determine what value of ϕ will be. Clearly, backward difference $D^{-x}_{i,j}\phi$ should be used to approximate ϕ_x when $u_i > 0$ and forward difference $D^{+x}_{i,j}\phi$ should be used to approximate ϕ_x when $u_i < 0$, where

$$D^{-x}_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{\triangle t} \tag{2.11}$$

$$D^{+x}_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\triangle t}. \tag{2.12}$$

The same is in approximating ϕ^n_y. This method of choosing an approximation to the spatial derivatives based on the sign of u and v is known as *upwinding*.

There are many other methods can get second order or third order accurate in approximating ϕ^n_x and ϕ^n_y, for example, Hamilton-Jacobin ENO, Hamilton-Jacobin WENO, and TVD RK (Total Variation Diminishing Runge-Kutta) method [7]. However, we use simple *upwind* method and *forward Euler* in this thesis. The reason is that we are interesting in segmentation using curve evolution not in numerical accurate of simulating curve evolution.

Fig. 2.4 shows the circle moving under the external velocity field $\vec{V} = (1, 0)$, this velocity field pushes the circle to move from left to right.

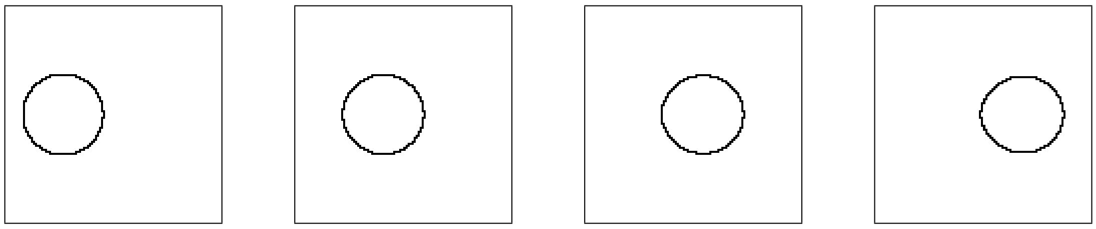


Fig. 2.4 Motion of a curve under external velocity field

2.3.2 Normal Direction with Mean Curvature

In the last subsection we discussed the motion of an interface in an external velocity filed. Now we consider the motion for a self-generated velocity. That is, the interface moves in normal direction with a velocity proportional to its curvature, which is generated from the ϕ itself. If we decompose $\vec{V} = V_n\vec{N} + V_t\vec{T}$, where $\vec{N}$ is the normal vector and $\vec{T}$ is tangent vector to the interface, the level set equation (2.8) is equivalent to

$$\phi_t + (V_n\vec{N} + V_t\vec{T}) \cdot \nabla\phi = 0, \tag{2.13}$$

since $\vec{N}$ and $\nabla\phi$ point in the same direction, $\vec{T} \cdot \nabla\phi = 0$. Then equation (2.13) becomes

$$\phi_t + V_n\vec{N} \cdot \nabla\phi = 0. \tag{2.14}$$

Here $V_n = -b\kappa$, where $b > 0$ and using (2.4), finally we have

$$\phi_t - b\kappa\,|\nabla\phi| = 0. \tag{2.15}$$

The mean curvature has the following effects. At the convex region, the curvature $\kappa > 0$ such that the velocity in the normal direction $V_n < 0$. This results in shrinking curve to the direction opposite to normal and trying to smooth curve. Similarly, at the concave region the

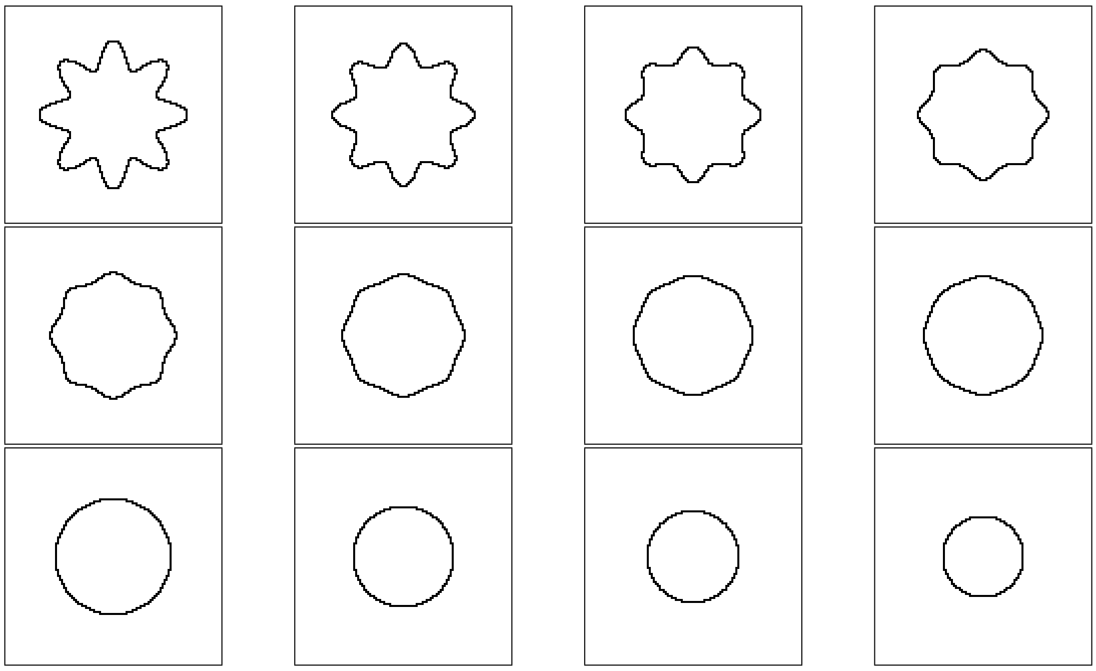


Fig. 2.5 Motion of a curve under curvature driven velocity

curvature $\kappa < 0$ and $V_n > 0$, the curve expands as to smooth curve. This motivation lets any curve evolving into a circle and shrinking to disappear. Fig. 2.5 shows a star-shaped contour moving under curvature driven velocity. The tips, that are convex, move inward while the gaps, that are concave, move outward.

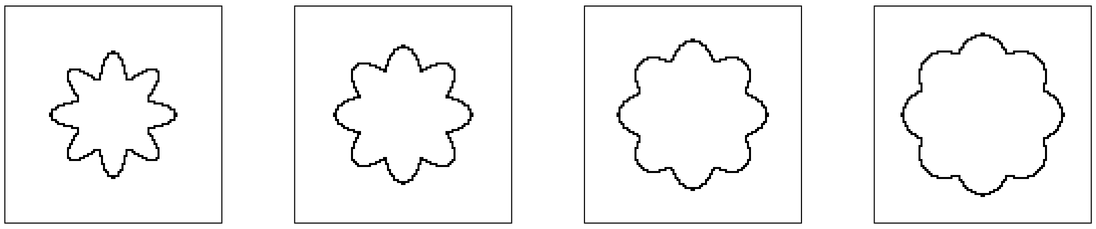
2.3.3 Normal Direction with Constant Velocity Field

In the last subsection the normal directional velocity of the interface is given by the mean curvature of the interface that determined by local geometric information. Now, we discuss the normal directional velocity of interface given by a constant velocity field $\vec{V} = a\vec{N}$. Just directly replacing $-b\kappa$ by a in the (2.15) yields the corresponding level set equation

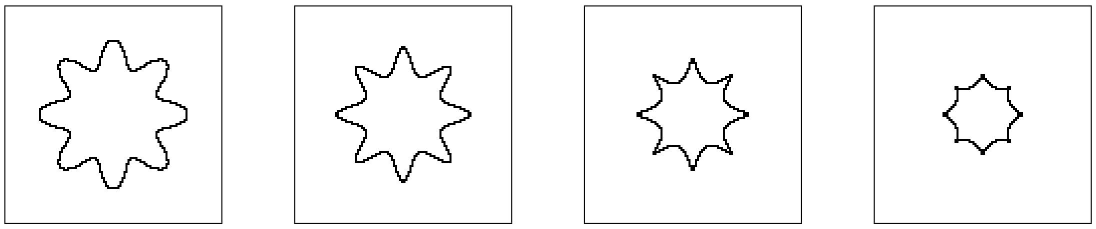
$$\phi_t + a\,|\nabla\phi| = 0. \tag{2.16}$$

When $a > 0$ the interface moves in the normal direction, and when $a < 0$ the interface moves in the opposite direction. When $a = 0$, the interface stays still. Since ϕ is a signed distance function we have $|\nabla\phi| = 1$, this reduces equation to $\phi_t = -a$.

By Forward Euler, $\phi^{n+1} = \phi^n - a\triangle t$, this equation gives a hint of the behavior of a curve.



(a) $a > 0$



(b) $a < 0$

Fig. 2.6 Motion of a curve under constant normal velocity

Suppose $a > 0$, if at some points $\mathbf{x_0}$ and $\mathbf{x_1}$, which $\phi(\mathbf{x_0}) = 0$ and $\phi(\mathbf{x_1}) = a\triangle t$, after a time step $\triangle t$, the $\phi(\mathbf{x_0}) = -a\triangle t$ and $\phi(\mathbf{x_1}) = 0$. This means that the zero level set moves to the normal direction with distance $a\triangle t$. Similarly, if $a < 0$ the zero level set moves to the opposite direction with distance $a\triangle t$. If we let our distance function ϕ be positive outside the circle and negative inside the circle, then the whole curve will expand outward for $a > 0$ and shrink inward for $a < 0$.

Fig. 2.6a is an example of a star-shaped moving under the constant normal velocity $a = 1 > 0$, and Fig. 2.6b is under the constant normal velocity $a = -1 < 0$. Compare to the motion under the curvature driven velocity, where only the curves of the convex regions are shrinking, the motions of convex regions and concave regions under the constant normal velocity are either both shrinking or both expanding.

2.3.4 Overall of Level Set Methods

We have discussed the concept of using level set methods to express curves and their motions under three kinds of velocities. Combine these three kinds of velocities to get,

$$\phi_t - b\kappa\,|\nabla\phi| + a\,|\nabla\phi| + \vec{V}\cdot\nabla\phi = 0. \tag{2.17}$$

Let F be the function of normal directional constant velocity of grid points. Weight each kind of velocity to yield the level set equation

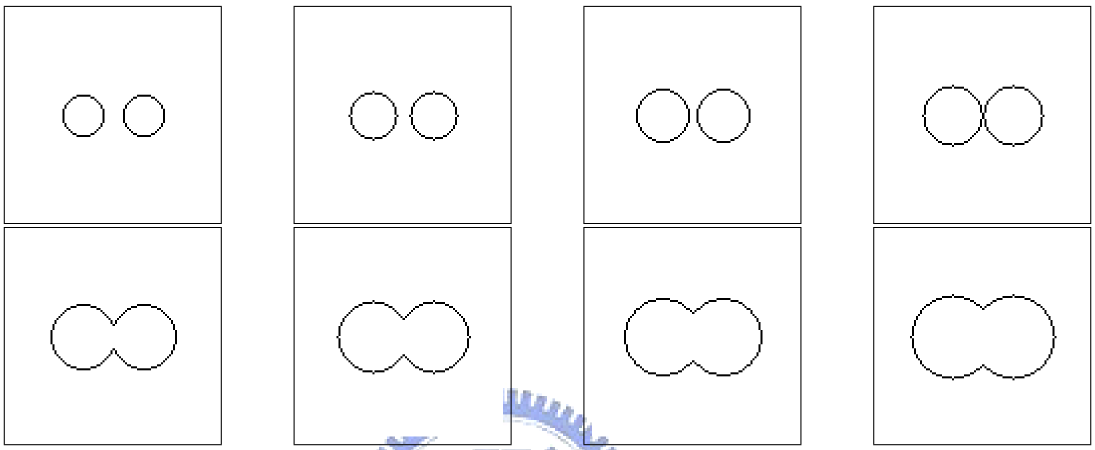
$$\phi_t = \alpha\kappa\,|\nabla\phi| + \beta F\,|\nabla\phi| - \gamma\vec{V}\cdot\nabla\phi. \tag{2.18}$$

By Forward Euler method, we have

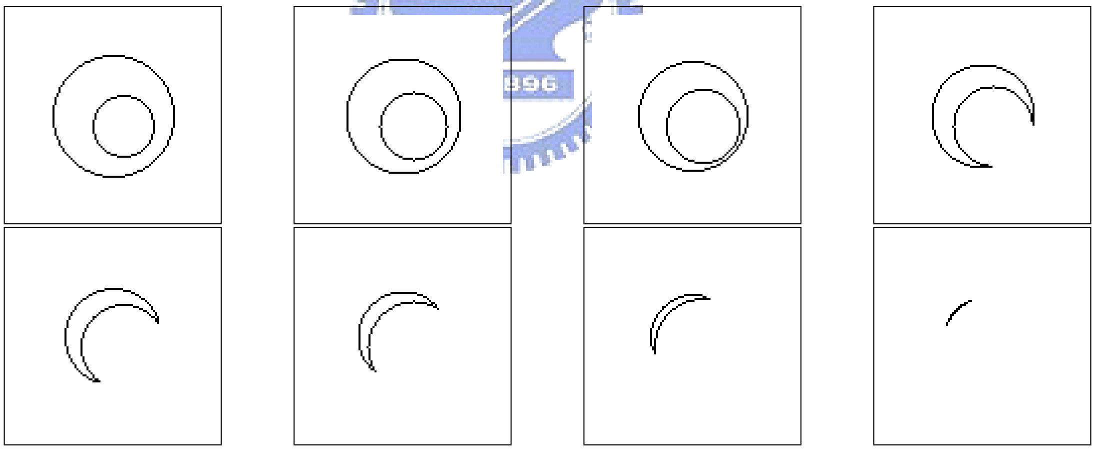
$$\phi_{n+1} = \phi_n + \triangle t(\alpha\kappa\,|\nabla\phi| + \beta F\,|\nabla\phi| - \gamma\vec{V}\cdot\nabla\phi). \tag{2.19}$$

The Courant-Friedreichs-Lewy condition (CFL condition) asserts that if want to have a stability approximation solution of (2.19), the numerical waves should propagate at least as fast as the physical waves. In other words, $(\alpha\kappa\,|\nabla\phi| + \beta F\,|\nabla\phi| - \gamma\vec{V}\cdot\nabla\phi) < \triangle x/\triangle t$ over all grid points. Since we work with images, the grid size is set to the pixel size. Thus $\triangle x = 1$ pixel. Our time step must be chosen to satisfy $\triangle t < 1/\max(\alpha\kappa\,|\nabla\phi| + \beta F\,|\nabla\phi| - \gamma\vec{V}\cdot\nabla\phi)$.

The first advantage of using level set methods is that we don't need to care about the merging or separating of curves. The topological changes of curve evolution is not problematic as traditional curve tracking methods, which have to remove or add tracking points and reconnect if two curves are merging or separating. The level set methods automatically handle topological changes. Fig. 2.7a shows two single circles that are expanding to connect each other to generate a single curve. Fig. 2.7b shows two circles, the outside circle is shrinking while the inside expanding, then merge to a horn shape. The second advantage is that it is easy to calculate the local properties of the curve, such as normal and curvature discussed in Section 2.2. Third is the generality for arbitrary dimensions and no self-intersection problem which results in swallowtail in PACs. And the most disadvantage is its computational expensiveness.



(a) Merge of two expanding circles



(b) Merge of two circles which outside is shrinking and inside is expanding

Fig. 2.7 Topological changes of curves

CHAPTER 3
IMAGE SEGMENTATION USING LEVEL SET METHODS

In this chapter we will introduce our survey of image segmentation using level set methods. Section 3.1 gives a preview of classification and skeleton of image segmentation using level set methods. Section 3.2, 3.3 and 3.4 provide the detail of Geodesic Active Contour, Gradient Vector Flow, and EdgeFlow, respectively. Rather than previous edge-based methods, Section 3.5 introduce the region-based method to evolve contours via level set methods.

3.1 Preview

Image segmentation is a basic preprocessing of many image analysis applications. Most methods process image segmentation based on various attributes such as texture, color, and gray level intensity. Most previous approaches to image segmentation can be classified as follows:

- Filtering-based methods : detect edges followed by edge linking.
- Region growing and merging.
- Global optimization based on energy functions and bayesian criteria.
- Graph partitioning and clustering.
- Curve evolution and active contour models.

Being different to the other ways which give a fragment result needed further processing, curve evolution methods usually result in closed contours and give a whole complete object segments. The two different methods to represent and implement curve evolutions are parametric active contours (PACs) and geometric active contours (GACs).

The major problem of using active contours to approach image segmentation is how to drive the curve to move to the boundaries of objects. In this thesis we will discuss several well-known

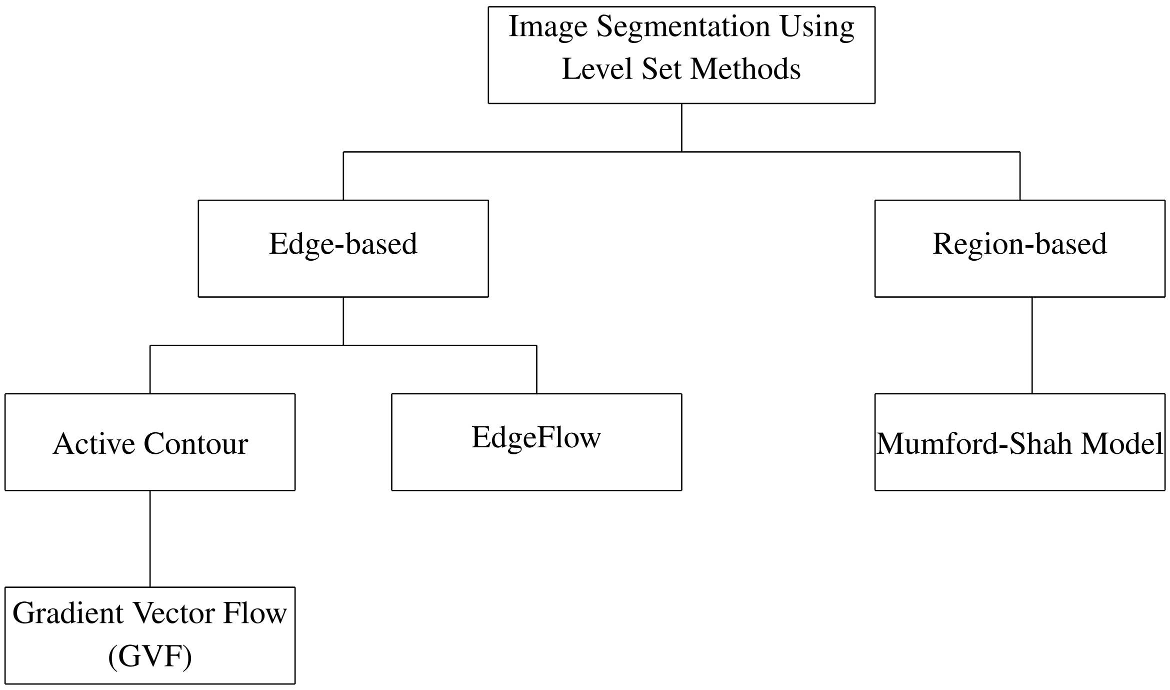


Image Segmentation Using Level Set Methods
Edge-based
Region-based
Active Contour
EdgeFlow
Mumford-Shah Model
Gradient Vector Flow (GVF)

Fig. 3.1 Image Segmentation using level set methods

level set methods of image segmentation using active contours.

3.2 Active Contours with Edge Stopping

Caselles et al. [8] and Malladi et al. [9] are among the first to use level set methods to extract objects from an image in their parallel works. They suppose the curve is either expanding or shrinking and stop evolving at the object boundary. This gives the idea to design a constant velocity field in the normal direction such that the velocities are positive or negative when the curve is at the smooth region and zero while at the object boundary.

Suppose $C(s)\colon\ [0,1] \to \mathbb{R}^2$ is a parametrization of a 2-D closed curve. The edge stopping is designed with the following edge function

$$g(x,y) = \frac{1}{1 + |\nabla G_\sigma(x,y) * I(x,y)|}, \tag{3.1}$$

where G_σ is a Gaussian function with standard deviation σ. In equation (3.1), it is obvious that at smooth regions we have $|\nabla G_\sigma * I| \to 0$ implies $g \to 1$ and at edges we have $|\nabla G_\sigma * I| \to \infty$

implies $g \to 0$. In fact, $|\nabla G_\sigma * I|$ as known is a simple method to find edges in an image. Another edge function which falls to zero faster on edges can be defined as:

$$g(x, y) = e^{-|\nabla G_\sigma(x,y)*I(x,y)|} \tag{3.2}$$

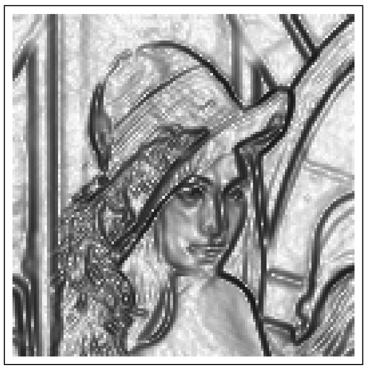


Fig. 3.2 $g(x, y) = 1/(1 + |\nabla G_\sigma(x, y) * I(x, y)|)$ of Lenna, normalize to $0 \sim 255$, where black means small value while white means large value.

Modify the curve evolution PDE as:

$$C_t = g(b\kappa + F)\vec{N}, \tag{3.3}$$

the corresponding level set equation is

$$\phi_t = g(\kappa + F)\,|\nabla\phi|\,. \tag{3.4}$$

The curve will expand if F is always positive and shrink if F is always negative. With the effects of g, the curve will slow down when progressing near to the edges, then stops while reaching the edges.

The problem of this method is that in the numerical calculation if the curve propagates beyond the desired boundary, it continues propagating, there is no other mechanism to push the curve back to the object boundary.

To overcome this problem, Cassela et al. [3] introduce Geodesic Active Contour (GAC) method , which uses minimization of the energy function

$$Min \int_0^1 g(C)\,|C'(s)|\;ds \tag{3.5}$$

and finds the gradient descend equation

$$C_t = g(\kappa + F)\vec{N} - (\nabla g \cdot \vec{N})\vec{N}. \tag{3.6}$$

The corresponding level set equation is

$$\phi_t = g(\kappa + F)\,|\nabla\phi| + \nabla g \cdot \nabla\phi. \tag{3.7}$$

The term ∇g generates an external velocity vector field $\vec{V} = -\nabla g$. The corresponding vector is almost zero at the smooth region, and points to the nearest edge at the neighborhoods of this edge, see Fig. 3.3a. As the contour moves beyond the edge, the vector pushes the contour back to the edge.

3.3 Gradient Vector Flow

Since the ∇g is almost zero at smooth region, the initial contour must be close to the true boundary or else it will evolve to ill result. In the papers [4] [10], Xu and Prince use an energy function to generate vector flow filed which is propagated from certain image generating vector field.

Define

$$\begin{aligned} f^1(x,y) &= |\nabla I(x,y)|^2\,, \\ f^2(x,y) &= |\nabla G_\sigma(x,y) * I(x,y)|^2\,. \end{aligned} \tag{3.8}$$

The field ∇f has vectors pointing toward the edges, but as mentioned above this kind of field has a narrow capture range. Xu an Prince generate vector filed $\vec{\mathbf{v}}(x,y) = (u(x,y), v(x,y))$, called the *gradient vector flow* (GVF) field, from minimizing the energy functional

$$\varepsilon = \iint \mu({u_x}^2 + {u_y}^2 + {v_x}^2 + {v_y}^2) + |\nabla f|^2 \, |\vec{\mathbf{v}} - \nabla f|^2 \, dx\, dy, \tag{3.9}$$

where f is chosen as f^1 or f^2 and μ is a parameter governing the tradeoff between the first and the second terms.

We see that when $|\nabla f|$ is small, the vector is dominated by partial derivatives of the vector field, when $|\nabla f|$ is large, the second term dominates the integrand, and is minimized by setting $\vec{\mathbf{v}} = \nabla f$. This says that the minimum solution will hold the original vector at neighborhoods of edges and propagate the vector for the smooth region next to the edges.

The *calculus of variations* gives the minimal solution of energy function in Euler equation forms. Given

$$E[u] = \iint_D F(x, y, u, u_x, u_y) dx\, dy, \tag{3.10}$$

the solution minimizing $E[u]$ satisfies

$$\frac{\partial F}{\partial u} = \frac{\partial}{\partial x}\frac{\partial F}{\partial u_x} + \frac{\partial}{\partial y}\frac{\partial F}{\partial u_y}. \tag{3.11}$$

Solving equation (3.9), the solution can be found from

$$\begin{aligned} \mu\nabla^2 u - (u - f_x)({f_x}^2 + {f_y}^2) = 0 \\ \mu\nabla^2 v - (v - f_y)({f_x}^2 + {f_y}^2) = 0. \end{aligned} \tag{3.12}$$

Now, we turn to solve (3.12) by treating u and v as functions of time

$$\begin{aligned} u_t(x,y,t) = \mu\nabla^2 u(x,y,t) - (u(x,y,t) - f_x(x,y)) \cdot (f_x(x,y)^2 + f_y(x,y)^2) \\ v_t(x,y,t) = \mu\nabla^2 v(x,y,t) - (v(x,y,t) - f_y(x,y)) \cdot (f_x(x,y)^2 + f_y(x,y)^2) \end{aligned} \tag{3.13}$$

where $u(x, y, 0)$ and $v(x, y, 0)$ are initialized as ∇f. In practice, the more iterations to solve (u, v) the more range vectors diffuse and the larger capture ability the result has.

C. Xu and J. L. Prince use the GVF in the parametric curve model

$$C_t(s,t) = \alpha C''(s,t) + \beta C''''(s,t) + \vec{\mathbf{v}}. \tag{3.14}$$

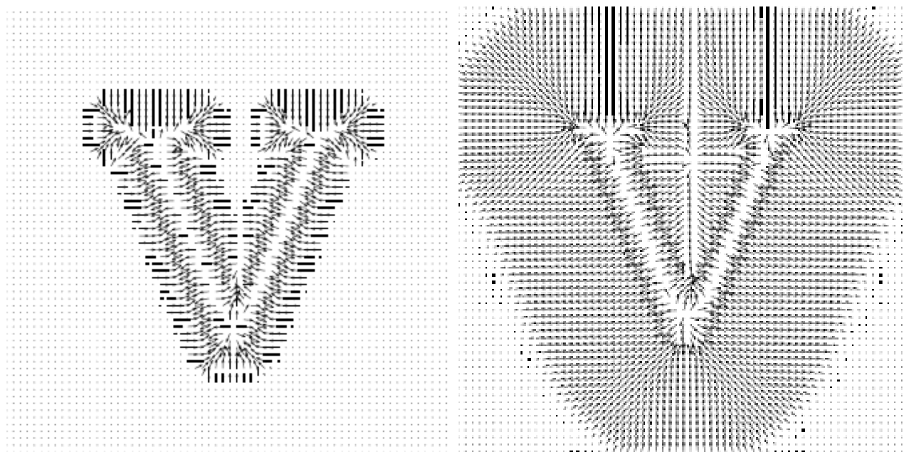

(a) $-\nabla g$ (b) GVF $\mathbf{v}$

Fig. 3.3 Compare vector field of GAC with GVF of V-shape.

In this thesis we replace f with $-\nabla g$, in last subsection, to generate GVF, and use GVF as external velocity field. This results in modifying equation (3.7) as

$$\phi_t = g(\kappa + F)\,|\nabla\phi| - \vec{\mathbf{v}} \cdot \nabla\phi. \tag{3.15}$$

Since $\mathbf{v}$ has large capture rang, the curve evolution can be driven by field rather than constant speed F. The curve will shrink at somewhere and expand at others.

3.4 EdgeFlow

Previous active contour methods aim to identify an object by utilizing the local discontinuities such as edges. The weak connection between contours and images is only with edge function or vector flow generated from edge function. Obviously, it's bad on nature images or texture images. As we will discuss in our experiment result, the segmentation is based on the founded edges.

Alternatively, Wei et al. [11] generate the *EdgeFlow* from the intensity information of the pixel and its neighborhoods, then Sumengen et al. [5] use *EdgeFlow* in active contours.

Decide the number of directions n and $\theta = \{\pi i/n \mid i \in 0, \ldots, n-1\}$. Consider a location s in the image, define a tuple measurement $F(s,\theta) = [E(s,\theta), P(s,\theta), P(s,\theta+\pi)]$, where

$E(s,\theta)$	edge energy at s along the orientation θ.
$P(s,\theta)$	probability of finding an image boundary in the direction θ from s.
$P(s,\theta+\pi)$	probability of finding an image boundary in the direction $\theta+\pi$ from s.

Sumengen et al. generate these measurements based on intensity and texture properties. For vector-valued image the intensity measurement can be calculated at each layer, RGB color image as example, we will have $F_R(s,\theta)$, $F_G(s,\theta)$, $F_B(s,\theta)$ and $F_{TEX}(s,\theta)$ for each direction θ from each location s. Finally, combining each orientation for each location s generates final *EdgeFlow*.

3.4.1 Useful Tools

We introduce several useful tools used to generate $F(s,\theta)$. The two-dimensional Gaussian function is defined as

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\sigma} e^{\dfrac{-(x^2+y^2)}{2\sigma^2}}. \tag{3.16}$$

The first derivative of Gaussian (GD) along the x-axis is

$$GD_\sigma(x,y) = \frac{\partial G_\sigma(x,y)}{\partial x} = -\frac{x}{\sigma^2} G_\sigma(x,y) \tag{3.17}$$

and the difference of offset Gaussian (DOOG) along the x-axis with distance is defined as

$$DOOG_\sigma(x,y) = G_\sigma(x,y) - G_\sigma(x+d,y), \tag{3.18}$$

where d is the offset between centers of two Gaussian kernels. To generate the derivative of Gaussian and difference of offset Gaussian along some orientation θ, we simply rotate the coordinates with $-\theta$ and calculate GD and DOOG. From linear algebra, we know that the new coordinates are

$$\begin{aligned} x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \tag{3.19}$$

thus

$$\begin{aligned} GD_{\sigma,\theta}(x,y) &= GD_{\sigma}(x',y') \\ DOOG_{\sigma,\theta}(x,y) &= DOOG_{\sigma}(x',y'). \end{aligned} \tag{3.20}$$

3.4.2 Intensity EdgeFlow

The GDs are generally used to determine if s lies near the edge and the edge lies in direction θ from s, i.e. the edge direction is around perpendicular to θ. If so, $|GD_{\sigma,\theta}(x,y) * I(x,y)|$ will have a large value. In fact, this is one dimensional derivative form different to $|\nabla G_{\sigma}(x,y) * I(x,y)|$.

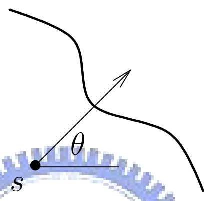


Fig. 3.4 Edge lies on direction θ from s

For two orientation θ_1 and θ_2, if $|GD_{\sigma,\theta_1}(x,y) * I(x,y)| > |GD_{\sigma,\theta_2}(x,y) * I(x,y)|$, then the edge may lies from s in direction θ_1 than θ_2. The function $E(s,\theta)$ will be used to express the strength of vectors. Thus, we define edge energy as

$$E(s,\theta) = \left|GD_{\sigma,\theta}(x,y) * I(x,y)\right|. \tag{3.21}$$

Another prediction method utilities DOOGs, the simple idea is that if there is an edge lies within the distance d in direction θ from s, the intensity different between $(x+d\cos\theta, y+d\sin\theta)$ and (x,y) should be large. The error in prediction is defined to be

$$\begin{aligned} Error(s,\theta) &= \left|I_{\sigma}(x+d\cos\theta, y+d\sin\theta) - I_{\sigma}(x,y)\right| \\ &= \left|DOOG_{\sigma,\theta}(x,y) * I(x,y)\right|. \end{aligned} \tag{3.22}$$

Therefore, the probabilities of EdgeFlow direction is

$$P(s,\theta) = \frac{Error(s,\theta)}{Error(s,\theta) + Error(s,\theta+\pi)}. \tag{3.23}$$

Note that σ is the only parameter needed to control.

3.4.3 Texture EdgeFlow

The texture features are extracted based on a Gabor wavelet decomposition scheme [12]. A two dimensional Gabor function $g(x,y)$ and its Fourier transform $G(u,v)$ are

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) e^{\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jWx\right]}, \tag{3.24}$$

$$G(u,v) = e^{-\frac{1}{2}\left(\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right)}, \tag{3.25}$$

where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$. By dilations and rotations of $g(x,y)$ through

$$\begin{aligned} g_{mn}(x,y) &= a^{-m}G(x',y'),\ a > 1,\ m,n \in \mathbb{Z} \\ x' &= a^{-m}(x\cos\varphi + y\sin\varphi) \\ y' &= a^{-m}(-x\sin\varphi + y\cos\varphi). \end{aligned} \tag{3.26}$$

this generates a class of function decompose frequency domain.

Give U_l and U_h to be the lower and upper center frequencies of interest. Let K be the number of orientations and S the number of scales. Then design

$$\begin{aligned} a &= \left(\frac{U_h}{U_l}\right)^{-\frac{1}{S-1}} \\ \sigma_u &= \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}} \\ \sigma_v &= \tan\left(\frac{\pi}{2k}\right)\left[U_h - 2\ln\left(\frac{\sigma_u^2}{U_h}\right)\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-\frac{1}{2}} \end{aligned} \tag{3.27}$$

using (3.26) with $W = U_h$, $m = 0, 1\ldots, S-1$, and $k = 0, 1\ldots, K-1$. The total number of decomposed frequencies are $N = S \cdot K$, see Fig. 3.5 as an example.

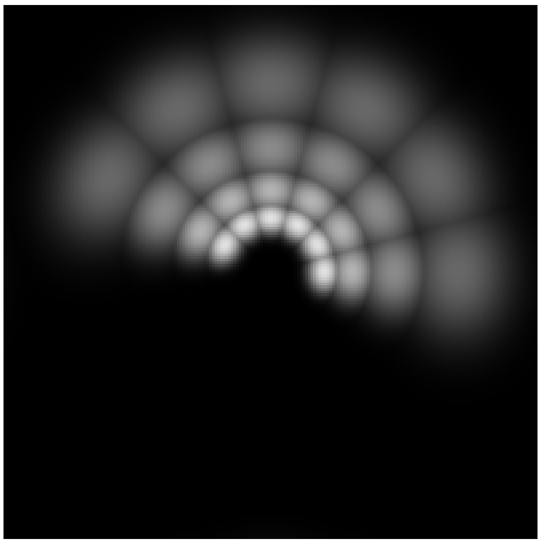


Fig. 3.5 Filter responses in the Gabor filter dictionary. The filter parameters used are $U_h = 0.35$, $U_l = 0.1$, $K = 6$ and $S = 4$.

Ma et al. define $U_l = 1/4\sigma$ and $U_h = 0.45$, as in the intensity edges, the σ is the only parameter needed to control. Applying filters on the image gives the Gabor filtered images

$$O_i(x, y) = g_i(x, y) * I(x, y) = m_i(x, y)\, e^{[\varphi_i(x, y)]} \tag{3.28}$$

where $1 \le i \le N$.

For each location (x, y), now we have a texture feature vector

$$\Psi(x, y) = [m_1(x, y), m_2(x, y), \ldots, m_N(x, y)] \tag{3.29}$$

The texture edge energy is defined as

$$E(s, \theta) = \sum_{1 \le i \le N} |GD_{\sigma,\theta}(x, y) * m_i(x, y)| \tag{3.30}$$

and the error in prediction is defined as

$$Error(s, \theta) = \sum_{1 \le i \le N} |DOOG_{\sigma,\theta}(x, y) * m_i(x, y)| \tag{3.31}$$

The probabilities $P(s, \theta)$ can be estimated using (3.23).

3.4.4 EdgeFlow Vector

For each location s, we had generated intensity and texture edge energy and probabilities in different orientation. The edge energy and probabilities can be combined together to form a single EdgeFlow field. Consider

$$\begin{aligned} E(s,\theta) &= \sum_{a\in A} E_a(s,\theta)\cdot\omega(a) \\ P(s,\theta) &= \sum_{a\in A} P_a(s,\theta)\cdot\omega(a), \qquad \sum_{a\in A}\omega(a)=1 \end{aligned} \tag{3.32}$$

where $A = \{intensity/color, texture\}$, the ω is the weighting coefficient with different image attribute $a \in A$. For example, EdgeFlow of the RGB color model image could be obtained with $A = \{red, green, blue, texture\}$ and maybe let $\omega(texture) = 0.4$ and $\omega(red) = \omega(green) = \omega(blue) = 0.2$. After combine various edge energy and probabilities, there are only one edge energy and one probability in each orientation at location s. The following work is to deicide flow direction at each lactation s form $\{[E(s,\theta), P(s,\theta), P(s,\theta+\pi)] \mid 0 \le \theta < \pi\}$ Define the continuous range of flow direction which maximize the sum of probabilities in a corresponding half plane

$$\Theta(s) = \arg\max_{\theta}\left\{\sum_{\theta\le\theta'<\theta+\pi} P(s,\theta')\right\}. \tag{3.33}$$

The EdgeFlow vector is then defined by

$$\vec{F}(s) = \sum_{\Theta(s)\le\theta<\Theta(s)+\pi} E(s,\theta)\cdot e^{j\theta}. \tag{3.34}$$

The $\vec{F}(s)$ is a complex number with magnitude representing edge energy and angle representing the flow direction. In our implementation the direct flow vector is

$$\vec{V}(s) = \left(\sum_{\Theta(s)\le\theta<\Theta(s)+\pi} E(s,\theta)\cos\theta, \sum_{\Theta(s)\le\theta<\Theta(s)+\pi} E(s,\theta)\sin\theta\right). \tag{3.35}$$

After this vector field is generated, the original method is followed by propagating vectors. The propagation ends and edges are made by checking the opposite direction vectors. If x or y component of two connect vectors change signs, the edges are marked. The boundaries are found by linking the edges. Segmentation is further processed with region merging.

Sumengen et al. [5] use EdgeFlow as external vector field in front propagation. They utilize EdgeFlow to produce the normal direction constant velocity F as the edge function. The edge

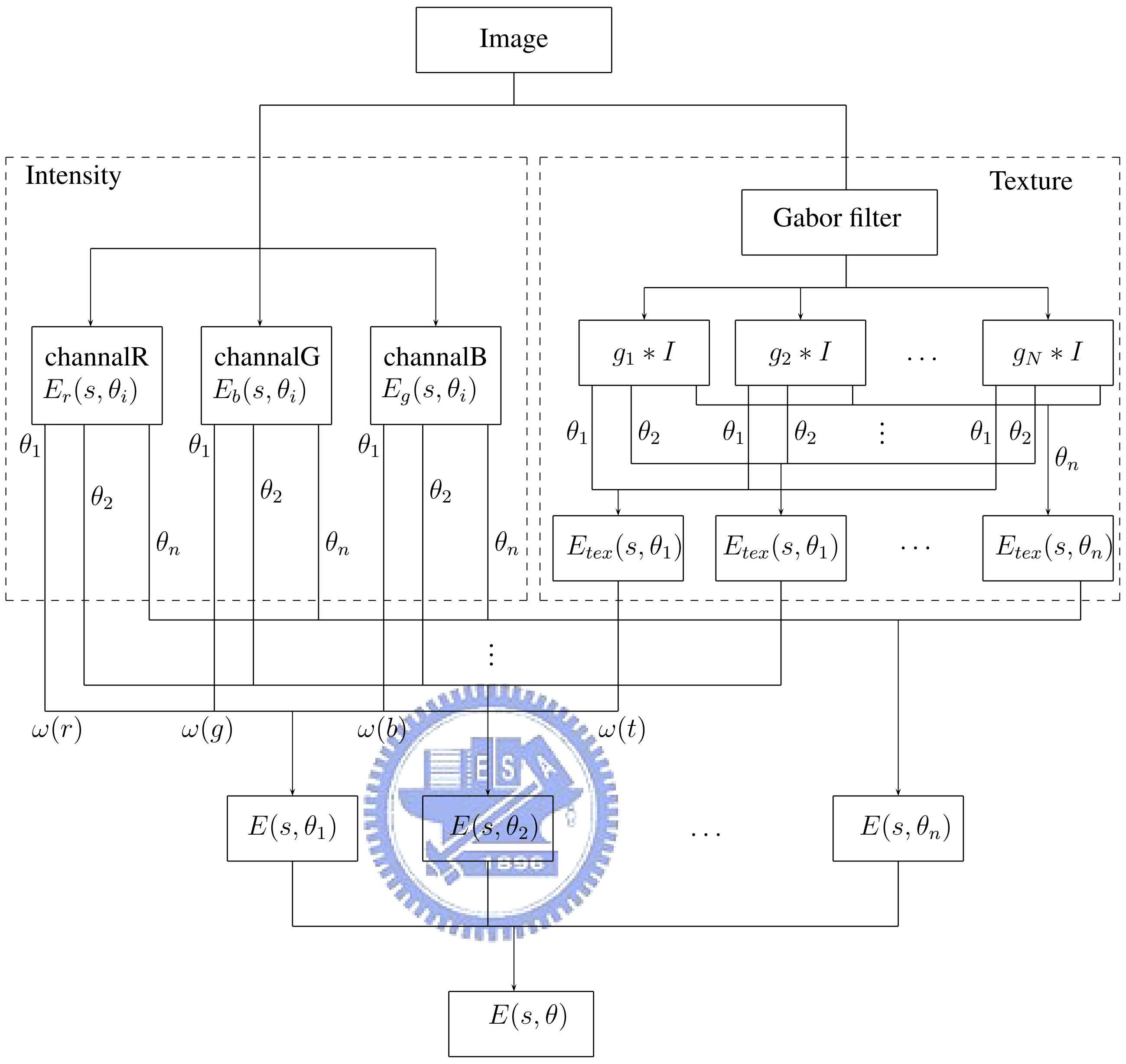


Fig. 3.6 The procedure to generate *EdgeFlow*

function F is derived as follows

$$F = \frac{1}{1 + \left|\vec{V}\right|}. \tag{3.36}$$

This is the reverse procedure of gradient vector flow method, which generates edge function first and then external velocity. We summarize the procedure as follows

3.5 Region Based Image Segmentation

Most methods of pushing the contour to the edge utilize vector flow field and stopping edge function, as we had surveyed prior, with level set active contour model (2.18). Another method is to generate level set functions based on minimization of an energy based-segmentation. Chan et al. [6] proposed a model using Mumford-Shah functional for segmentation [13].

3.5.1 Description of model

Suppose $u_0 \colon \bar{\Omega} \to \mathbb{R}$ is the given image and C is the curve in Ω. The Mumford-Shah functional for segmentation is

$$\begin{aligned} F^{MS}(u, C) = & \mu \cdot Length(C) \\ & + \lambda \int_{\Omega} |u_0(x,y) - u(x,y)|^2 \, dx \, dy \\ & + \int_{\Omega \backslash C} |\nabla u(x,y)|^2 \, dx \, dy, \end{aligned} \tag{3.37}$$

where μ and λ are positive parameters.

The solution of image u is obtained by minimizing this functional and is formed by smooth regions R_i with sharp boundaries, denoted here by C. Mumford and Shah reduced F^{MS} with restriction to piecewise constant functions $u = c_i$ where c_i is constant on each connected component R_i of $\Omega \backslash C$. The model proposed by Chan et al. is a particular case of the Mumford and Shah's model, which the $\Omega \backslash C$ is decomposed according to only two piecewise components $R_1 = inside(C)$ and $R_2 = outside(C)$.

The idea of Chan et al. is that assume the image u_0 is formed by two regions of piecewise-constant intensities, generally one object and one background. Then consider the fitting energy function

$$F_1(C) + F_2(C) = \int_{inside(C)} |u_0(x,y) - c_1|^2 \, dx \, dy + \int_{outside(C)} |u_0(x,y) - c_2|^2 \, dx \, dy, \tag{3.38}$$

where $c_1 = average(inside(C))$ and $c_2 = average(outside(C))$. If C is fitting to the boundary of the object, the $F_1(C) + F_2(C) \approx 0$. Therefor in active contour model the curve C will minimize this function, and we add regularizing terms the length of C and the area inside C. This energy function is

$$\begin{aligned} F(C, c_1, c_2) =& \mu \cdot Length(C) + \nu \cdot Area(inside(C)) \\ &+ \lambda_1 \int_{inside(C)} |u_0(x,y) - c_1|^2 \, dx \, dy \\ &+ \lambda_2 \int_{outside(C)} |u_0(x,y) - c_2|^2 \, dx \, dy. \end{aligned} \tag{3.39}$$

3.5.2 The Level Set Formulation of the Model

Represent the curve C with level set methods, here we inverse the ϕ to $\phi > 0$ inside the curve and $\phi < 0$ outside the curve. The function becomes:

$$\begin{aligned} F(\phi, c_1, c_2) =& \mu \cdot Length\{\phi = 0\} + \nu \cdot Area\{\phi > 0\} \\ &+ \lambda_1 \int_{\phi > 0} |u_0(x,y) - c_1|^2 \, dx \, dy \\ &+ \lambda_2 \int_{\phi < 0} |u_0(x,y) - c_2|^2 \, dx \, dy. \end{aligned} \tag{3.40}$$

Note integral terms in the function are over inside and outside regions of the curve respectively. Besides, after evolved with error of accuracy the zero level set lying between $\phi > 0$ and $\phi < 0$ may not exactly zero. In order to set the integral over the entire image domain Ω and measure the length of the curve we utilize the following two special functions:

- Heaviside function

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x \leq 0. \end{cases} \tag{3.41}$$

- Dirac measure δ

$$\delta(x) = \frac{d}{dx} H(x). \tag{3.42}$$

Express the function F in

$$\begin{aligned} F(\phi, c_1, c_2) =& \mu \int_\Omega \delta(\phi) \left|\nabla \phi\right| dx\, dy + \nu \int_\Omega H(\phi) dx\, dy \\ &+ \lambda_1 \int_\Omega \left|u_0(x,y) - c_1\right|^2 H(\phi) dx\, dy \\ &+ \lambda_2 \int_\Omega \left|u_0(x,y) - c_2\right|^2 (1 - H(\phi)) dx\, dy. \end{aligned} \tag{3.43}$$

Minimizing $F(\phi, c_1, c_2)$ using a gradient descent method, yields the associate Euler equation for ϕ

$$\phi_t = \delta(\phi) \left[\mu \cdot \text{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2 \right]. \tag{3.44}$$

The additional length term is a regularizing term and has a scaling role. If μ is large, only lager objects are detected, while for small μ, much and smaller objects are also detected. Chan et al. set $\nu = 0$ and $\lambda_1 = \lambda_2 = 1$. This is different to general level set active contour model. Actually, the term $-\nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2$ can be seen as the constant speed in the normal direction but is dependent on the position and time.

3.5.3 Model of Vector-Valued Images

In the continuous paper of Chan et al. [14], they extended the method to vector-valued images. The model lets the curve minimize the energy function according to each channel $u_{0,i}$ of an image, with $i = 1, \ldots, N$ channels.

Let $\overline{c^+} = (c_1^+, \ldots, c_N^+)$ and $\overline{c^-} = (c_1^-, \ldots, c_N^-)$ be the vectors of the averages of inside and outside curve on each channel respectively. The extension of model is

$$\begin{aligned} F(\phi, \overline{c^+}, \overline{c^-}) =& \mu \cdot Length(C) \\ &+ \int_{inside(C)} \frac{1}{N} \sum_{i=1}^N \lambda_i^+ \left|u_{0,i}(x,y) - c_i^+\right|^2 dx\, dy \\ &+ \int_{outside(C)} \frac{1}{N} \sum_{i=1}^N \lambda_i^- \left|u_{0,i}(x,y) - c_i^-\right|^2 dx\, dy. \end{aligned} \tag{3.45}$$

Rewriting it in level set form, we have

$$\begin{aligned} F(\phi, c^+, c^-) = & \mu \int_\Omega \delta(\phi)\, |\nabla \phi|\, dx\, dy \\ & + \int_\Omega \frac{1}{N} \sum_{i=1}^{N} \lambda_i^+ \left| u_{0,i}(x,y) - c_i^+ \right|^2 H(\phi) dx\, dy \\ & + \int_\Omega \frac{1}{N} \sum_{i=1}^{N} \lambda_i^- \left| u_{0,i}(x,y) - c_i^- \right|^2 H(1-\phi) dx\, dy \end{aligned} \tag{3.46}$$

and the Euler equation for ϕ

$$\phi_t = \delta(\phi) \left[\mu \cdot \mathrm{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \frac{1}{N} \sum_{i=1}^{N} \lambda_i^+ (u_{0,i} - c_i^+)^2 + \frac{1}{N} \sum_{i=1}^{N} \lambda_i^- (u_{0,i} - c_i^-)^2 \right]. \tag{3.47}$$

The μ is regularizing term as above, λ_i^+ and λ_i^- are used to set different weighing according each layer.

CHAPTER 4
EXPERIMENT RESULTS

In this chapter we will discuss the experiment results using the level set segmentation methods. First, we will discuss the effects of parameters when using level set methods on image segmentation. Then some artificial images will be the sample of image segmentation in order to compare the noise effects of these methods. In our experiment we can organize these methods from the view point of how they drive the curves. The first part is driven by generated vector field with edge function, this part including Geodesic Active Contour, Gradient Vector Flow and Edge Flow. The vector field $\vec{V}$ and edge function F are calculated before the curve evolution and fixed at all time. Another part is driven by energy balance which the curve evolution is according to the region stability which changes with time to time. We will discuss the improvement progress of these methods and observe the practical situation in our experiment.

4.1 Vector Driven Methods

The general level set equation of curve evolution is of the form

$$\phi_t = \alpha\kappa\,|\nabla\phi| + \beta F\,|\nabla\phi| - \gamma\vec{V}\cdot\nabla\phi.$$

We process the image segmentation with procedure as in Fig. 4.1 to generate the curve evolution segmentation. With the several methods we have introduced, each term can be summarized as

- κ based on the curve itself,
- F the edge function used to decide whether the curve is still expanding or shrinking or stops at the edges,
- $\vec{V}$ the velocity field pushes the curve to move to the edges.

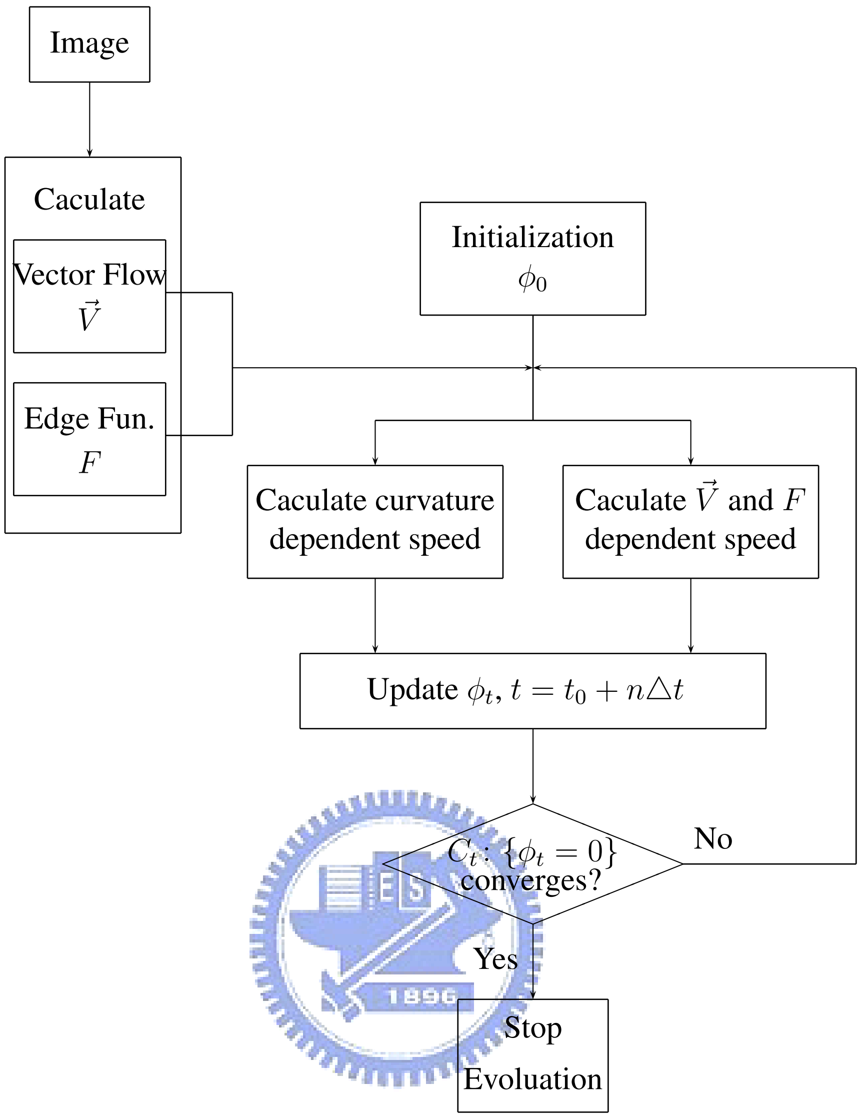


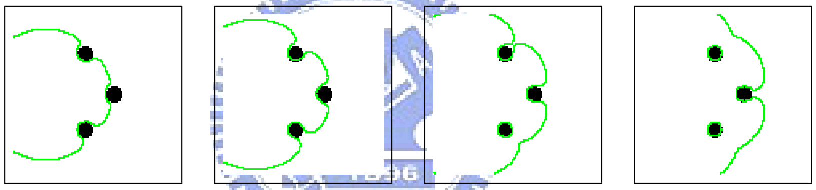
Image
Caculate
Vector Flow $\vec{V}$
Edge Fun. F
Initialization ϕ_0
Caculate curvature dependent speed
Caculate $\vec{V}$ and F dependent speed
Update ϕ_t, $t = t_0 + n\triangle t$
$C_t : \{\phi_t = 0\}$ converges?
No
Yes
Stop Evoluation

Fig. 4.1 The procedure of image segmentation using level set methods

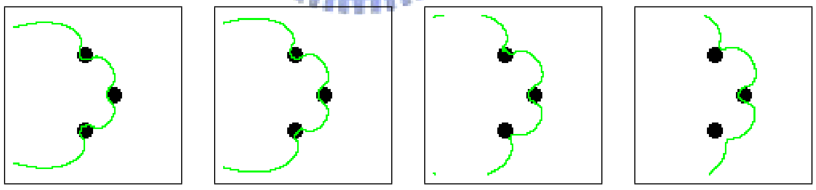
The total amount of these three forces decide the curve evolution. Besides, α, β, and γ are weights to each term and σ is used in calculating Gaussian filtered image. In the experiment we manual set $\gamma = 1$ and $\alpha < 0.3$ for each method. How to set a value for β is the most critical problem in curve evolution for segmentation. Too small β will make curve move slowly and may stick if there is no external velocity vector. Too large β will let curve move across the weak edges even there have external velocity pushing curve back to edges, but it is not enough to balance the force given by F. We first test the effect of each velocity as follows.

4.1.1 Curvature Term

In fact, the κ is set to $-\kappa$ as we discussed in section 2.3.3. The κ keeps the curve evolving smoothly since it can suppress the high curvature location. In the segmentation, the κ also effects the sensitives of curve when moving pass the noise. Fig. 4.2 shows an example when a curve is moving across the noise, In this example, the size of each dot is about 10 pixels and $\beta = 0.3$. Fig. 4.2a is curve evolving with $\alpha = 0.1$, Fig. 4.2b is shows the curve evolved with $\alpha = 0.5$. The difference is that when the curve meets the noise with small α value, it will be stopped according the relative $F \approx 0$, although κ velocity should make small expansion of the curve it is not large enough to overcome the influence of $\vec{V}$ which pushes the curve back to the boundary of dots, and with large α value, the κ term of velocity will be able to overcome the influence of $\vec{V}$ near the boundary.



(a) $\alpha = 0.1$



(b) $\alpha = 0.5$

Fig. 4.2 The curve evolution under different κ weighting

4.1.2 Edge Function Term

Next, we experiment the Edge Function term $F = g$, we find that this term controls the segmentation results in methods GAC and GVF. There are two variables that generate edge function and affect the curve evolution. First, the σ of the Gaussian smooth function decides the pixel rang of an edge to be found. Fig. 4.3b and Fig. 4.3c show $\sigma = 0.75$ and $\sigma = 2$, respectively. With small σ values the Gaussian smooth function is sensitive to the noises and the resultant curve is hard to evolve because at most ranges $F \approx 0$. We try to increase the α and β weights of κ and F, respectively, to overcome this situation in high noise images. We find that although this makes curve to easily pass through the noise in theory, but in practice, it does not work. Especially the new problem occurs in the hight curvature corner such as branch corner of a vessel, the high α value will let curve to expand and leave the corner.

Use large σ value can blur the noise but it will also extend the small F area around the noises, in this situation, some weak but border noise will result extension area of $F \approx 0$. We can see this at top left branch in Fig. 4.3c, there is a blurred edge across the vessel, when curve enter the narrow vessel it will stop.

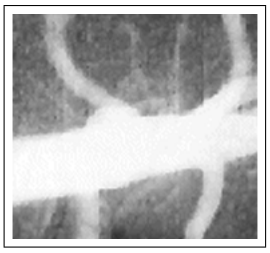
We use the following fixed edge function to nonlinearly enhance the F values.

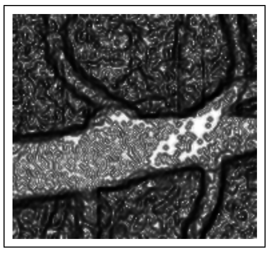
$$F = g = \frac{1}{1 + a * |\nabla G_\sigma * I|}. \tag{4.1}$$

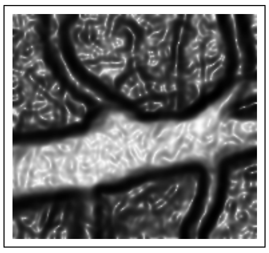
Observe that F will have a larger value if $0 < a < 1$. This means the F value of noise will not approximate zero, but this also increases the F value of real edges. We can use a threshold d to cutoff the F, if $F < d$ we set F value equal to zero. And if $a > 0$ it can used to reduce the F in the weak edges in noiseless images, which have smooth areas but the edge between objects are not strong. We can chose a according to the original edge function to look if an image is filled with noise or filled with smooth regions with weak edges between them.

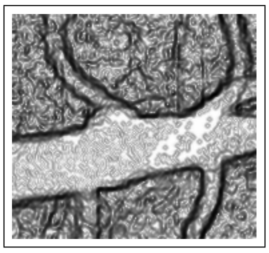
The edge linking is another problem in F, if there are small gaps in the edge the curve will expand out from this gap. We use a threshold as above to link the edge, this method also set the F value of edges absolutely equal to zero. Because we work on the gray images, the maximum $|G_\sigma * I|$ is about $255\sqrt{2}$. Thus the smallest $F = 1/(1 + 255\sqrt{2}) \approx 0.002$, after several time steps, the curves will leave the edges. If we set the F values of edges to be zero, we can avoid this situation. But this improvement only works for gaps small enough. For large discontinuity, it is really the problem of edge detection.

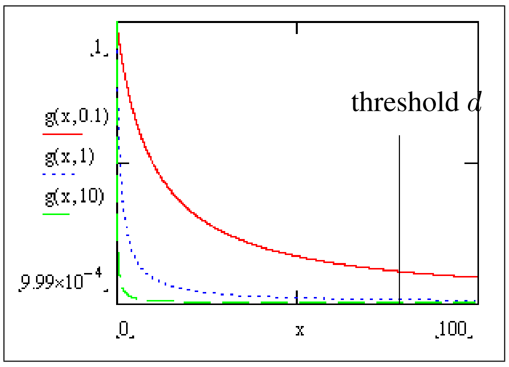
These problems are caused by that the edge functions are generated roughly from gradient


(a) $\sigma = 0.75$


(b) $\sigma = 0.75$


(c) $\sigma = 2$


(d) scale with $a = 0.1$



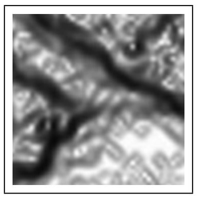
(e) $g(x,a) = 1/(1 + a\,|x|)$

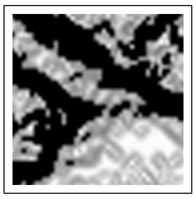
Fig. 4.3 Different control of generating edge function

of gaussian smoothed image. The methods GAC and GFV design the curve to stop according to the edge function, that means where the final curve is, where the edges are found from gradient of gaussian smoothed image.

Summarize the adjustment for F in the experiment

- select suitable σ
- select a to enhance edge function according the original edge function
- select the threshold d to set $F = 0$ if F is less than the threshold


(a) No threshold


(b) threshold with 0.5

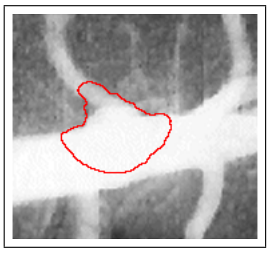
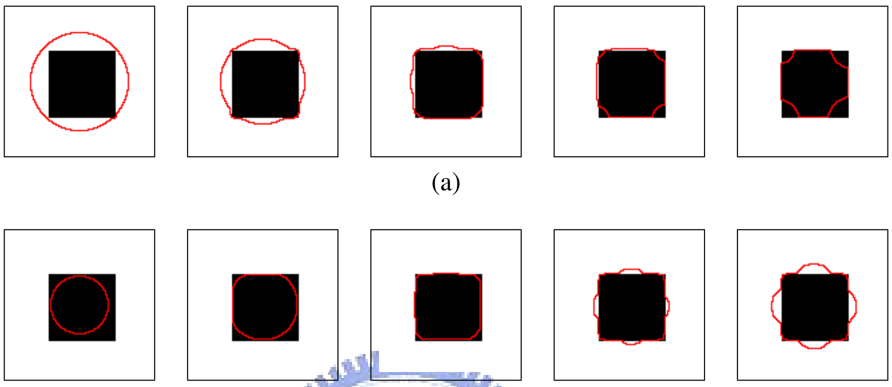

(c) curve expand over the edge

Fig. 4.4 Edge linking by threshold

4.1.3 Vector Field Term

The main difference between vector driven methods is the vector field generation. Before test the vector field generated by these methods, we explain the basic edge stopped method without external vector field. The basic method uses only κ and edge function F, and the initial curve is set to cover the objects or entirely inside an object. Then the curve expands or shrinks until reaches the edges. But after several time steps, the curves will leave the edges, as we have



(a)

(b)

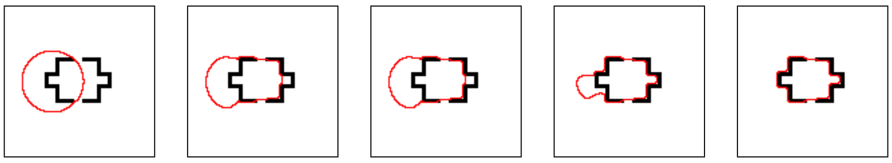
Fig. 4.5 Simplest segmentation by active curve

discussed in last subsection. The improvement of GAC is to add the external vector field around the edges. This let the curve stop at the edges. We can see the same effect in the GVF too, since the GVF is extended from GAC. The drawback is that the curve still need to be initialized to cover the object or to be inside the object. The GVF propagates the vector field and emphasizes three improvements

- the curve can be set across the object,
- avoid the curve across the gap of edges,
- curve can move to enter concave regions.

We show this in Fig. 4.6, which the curve is initialized to across the object and is influenced by the vector field to surround the object without crossing top and bottom gaps.

The firs improvement is because the vectors around the gaps will be generated after vector propagation and points to the opposite of edges as Fig. 4.6b shows. The second one can be used


(a)

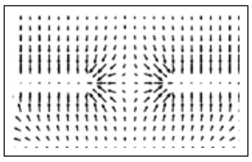

(b)

Fig. 4.6 Improvement by GVF

to overcome the edge gaps problem as we have mentioned above. Unfortunately, the improvement only suits to the situation when there is only one object we want. In our experiment, we observe when the curve movement there are multi-object. The curve stops between the parallel corners or edges of objects.

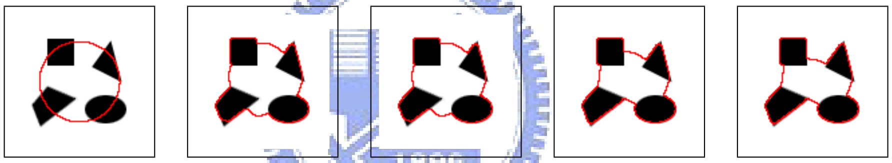
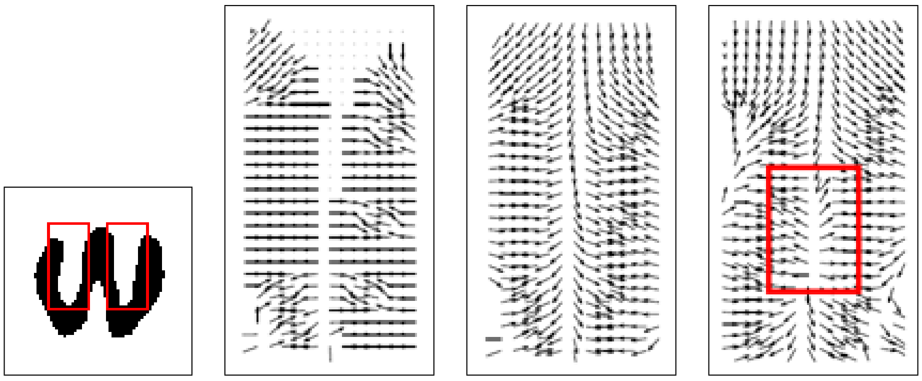


Fig. 4.7 Multi-Object curve stop at gaps

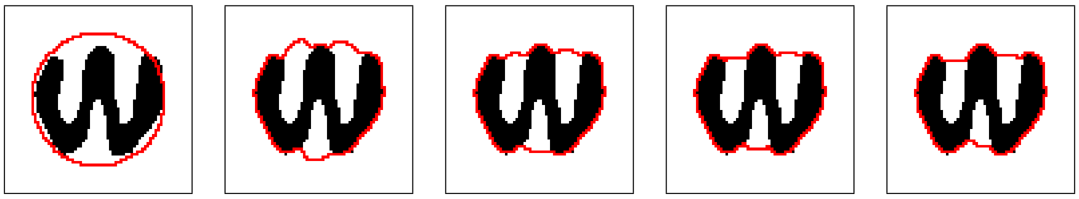
The last improvement is that the curve can move into the concave regions of an object. Fig. 4.8 shows an example about this problem, the original $-\nabla g$ in the concaves of w-shaped point horizontally and parallelly to left and right edges as shows in Fig. 4.8b.

When the curve enters the concaves, it will be stopped. After propagation using GVF method, the vectors in the concave turn to point into concave as shown in Fig. 4.8c. Fig. 4.9 shows the curve evolution with $-\nabla g$ and GVF respectively. But in our experiment we observe that this only works well if the concave region is open wide and reduce to narrow or almost the same width. If the concave region is from narrow to wide or with some bulge, the vectors after propagation will point outward the concave and the curve stays still. We can see this in Fig. 4.8d. Fig. 4.10 shows a clearer example, where vectors in the front part of concave point out

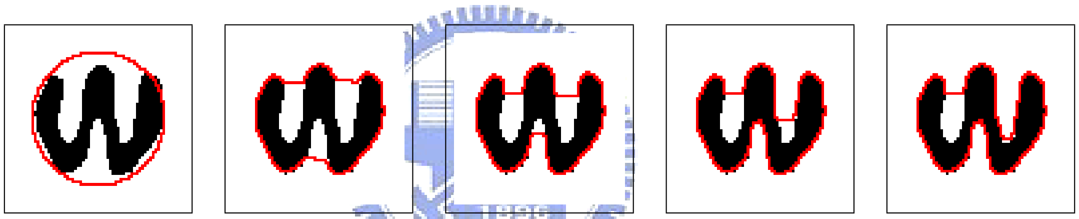


(a) w shape (b) right concave of $-\nabla g$ (c) right concave of GVF (d) left concave of GVF

Fig. 4.8 Vector Field on the concave region



(a) with $-\nabla g$



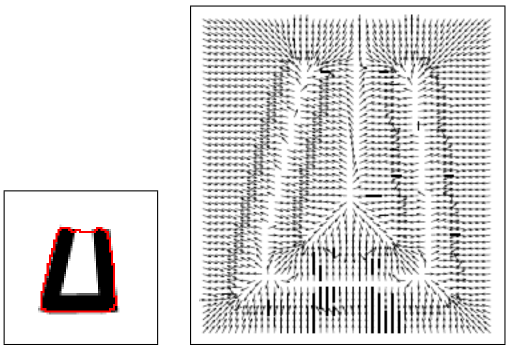
(b) with GVF

Fig. 4.9 The curve evolution under different vector field

and vectors in the rear area of concave point into concave.

The GAC and GVF generate vector field based on gradient of edge function, i.e. $F = g$ and $\vec{V} = -\nabla g$, this means there are limits to the texture images.

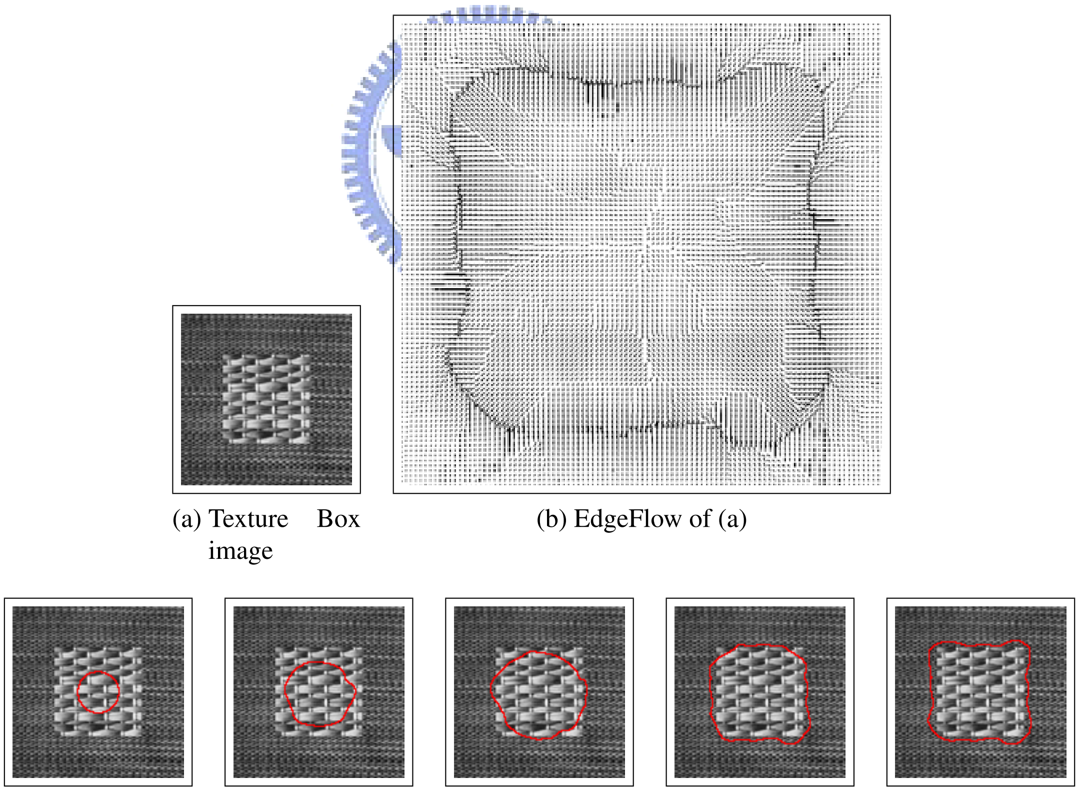
In contrast to generate vector field from edge function, EdgeFlow uses Gabor filter to give a better vector flow of texture images. Fig. 4.11 shows a texture box with similar intensity. In the experiment we utilizes the streamlines to compare the vector field generated by different parameters. The streamlines are the paths over which a dense number of free particles move under the influence of external forces when laced in the external force field. From the experiment, in the texture images, the σ has to set large, in order to avoid the disorderly vector groups which obstruct the curve evolution. We show this in the Fig. 4.12, with $\sigma = 3$ there are a little textures



(a) u shape (b) GVF

Fig. 4.10 U-shape Concave region

in the inside box which are regarded as edges and results the vectors point to them. And we compare the EdgeFlows generated with intensity or texture only to look the effect of the prediction using Gabor filter. The most drawback of EdgeFlow is that it is very time consuming.



(a) Texture Box image (b) EdgeFlow of (a)

(c) Segmentation with EdgeFlow

Fig. 4.11 EdgeFlow generate with 12 orientation, $\sigma = 5$, $S = 5$, $K = 6$, texture EdgeFlow only.

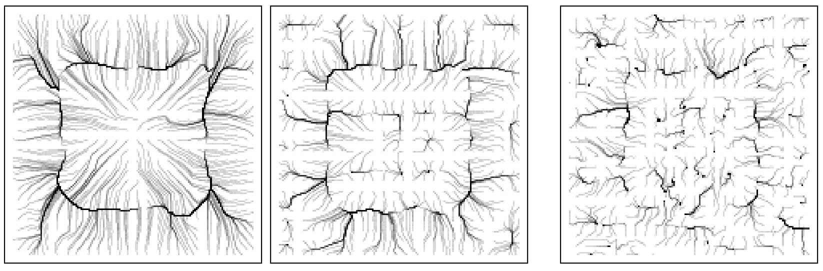
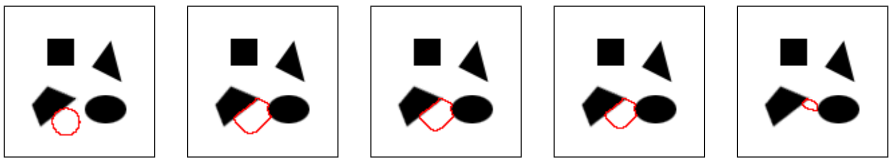

(a) $\sigma = 5$ texture only (b) $\sigma = 3$ texture only (c) $\sigma = 3$ intensity only

Fig. 4.12 Streamlines of EdgeFlow vector field

4.1.4 Force Balance Problem

We have tested the each vector driven method in terms of each force. In these methods we set the curve to cover, inside or across the objects. While the initial curve is far from the objects or edges, there are two methods to evolve curves. First method is to suppose that the vector field $\vec{V}$ is able to move the curves to the objects. But in the multi-object images or nature images, the vectors between the objects are disarray. And the curve will be ill evolved. Second is to set the curve expanding by F and will around the objects, this means in the smooth region the effect of $\alpha\kappa + \beta F$ is must larger than $\gamma\vec{V}$, otherwise the curve will have ill-evolution as above. We show our experiment in Fig. 4.13


(a) Ill evolution

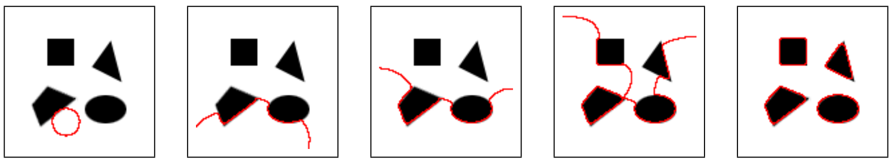
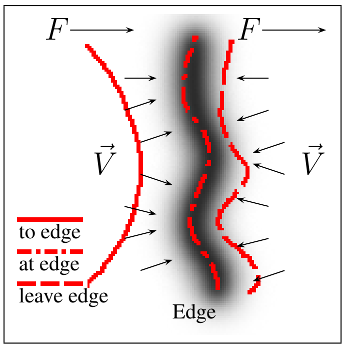

(b) Expanding around the objects

Fig. 4.13 The curve initialize outside the objects

But $\alpha\kappa + \beta F$ should also be less than $\gamma\vec{V}$ near the edges. Otherwise, the little values of $\alpha\kappa + \beta F$ will let the curve leave the edge after several time steps, the $\vec{V}$ is not able to push the curves back to the edges. And the curves will keep leave the edges.



Summarize each parameter setting from experiment as follow

- $\alpha\kappa + \beta F > \gamma\vec{V}$ let curves expand to edges at smooth region.
- $\alpha\kappa + \beta F$ small enough at edges avoid curve leave the edges.
- $\alpha\kappa + \beta F < \gamma\vec{V}$ let curves back to the edges when curves try to leave edges.

4.2 Energy Balance Methods

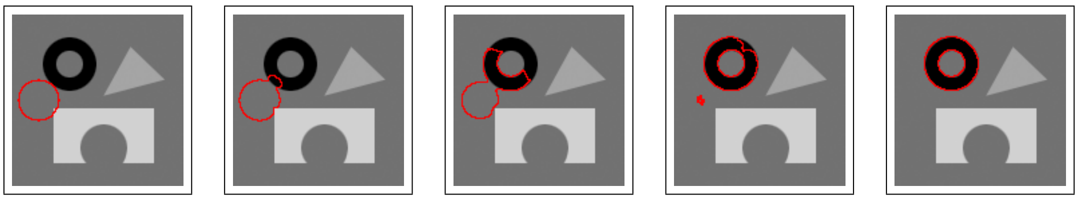
Instead of vector driven methods, we test on the region based methods. The region based equation is

$$\phi_t = \delta(\phi)\left[\mu \cdot \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \frac{1}{N}\sum_{i=1}^{N}\lambda_i^+(u_{0,i} - c_i^+)^2 + \frac{1}{N}\sum_{i=1}^{N}\lambda_i^-(u_{0,i} - c_i^-)^2\right]$$

The algorithm of region based method is:

```
**Initialize** $\phi_0$, $t = t_0$
**for** $n$ less than maximum iterations
    **Compute** $c_i(\phi_t)$
    **Compute** div$\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$
    **Update** $\phi_t$, $t = t_0 + n\triangle t$
**end**
```

The initial curve locations affect the segmented objects in region based method. Fig. 4.14 shows the segmentation progress of two initial curves. It is worthy to notice that the inside hole of the circle object is also segmented out by the curve. The time cost in region based method are much less than the vector driven methods, i.e. much fewer iterations of solving ϕ are needed.


(a)

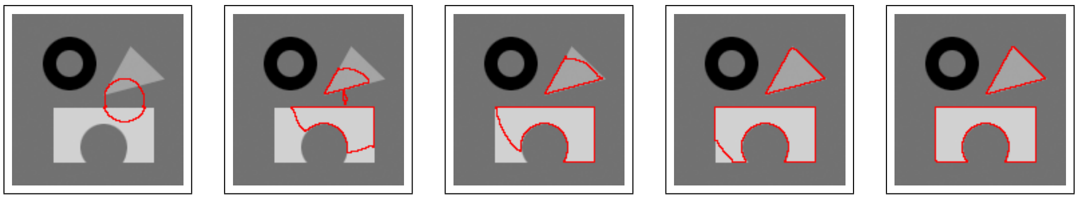

(b)

Fig. 4.14 Region based method initial in different location

4.3 Artificial Images

We test the image segmentation on artificial images. In all tests the parameters are manually set to have best results. In Fig. 4.15 , we use GVF to segment the three objects shown. In this example, we also can see the curve merging and splitting. Then we add 15%, 30% and 45% Gaussian noise respectively, In the 15% Gaussian noise the GVF still can cut out the objects, as shown in Fig. 4.16, but with more noise it is hard to set a suitable parameter to find out the objects, since the methods are based on the edge and the edge function are noise sensitively. An ill-improvement is to set an initial curve which covers all objects then shrink it by a vector field. We use EdgeFlow to generate a under vector filed around edges. The segmented result with 45% gaussian noise added are shown in Fig. 4.17. With noise image, the region based method gives good results on Fig. 4.18a and Fig. 4.18b.

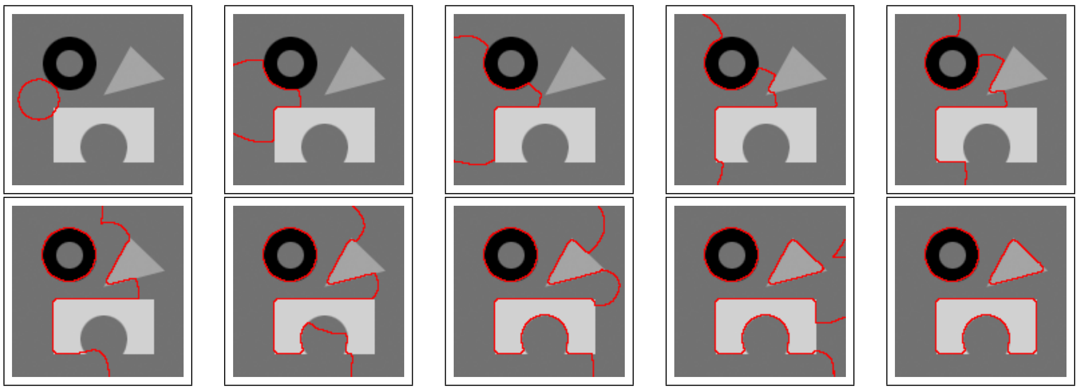


Fig. 4.15 Segment three objects by Gradient Vector Flow, $\sigma = 0.5$, $\mu = 0.1$ with 60 iteration, $\alpha = 0.05$, $\beta = 0.2$

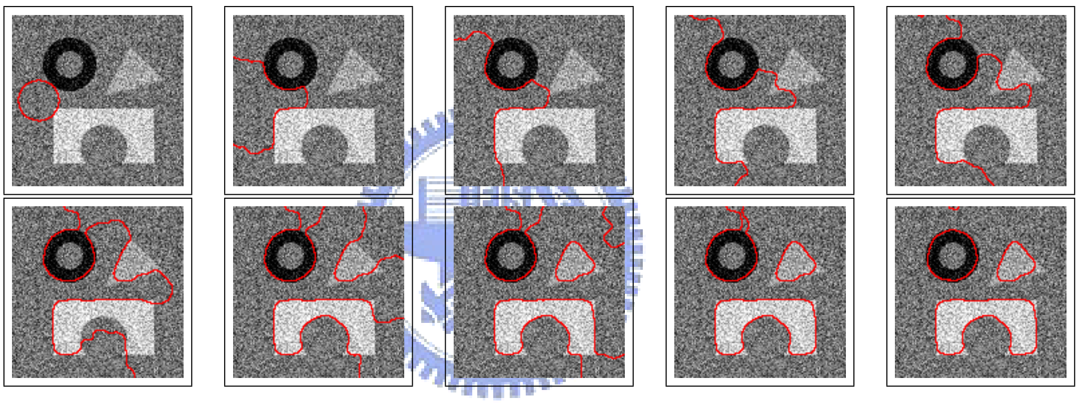


Fig. 4.16 Added 15% Gaussian noise, $\sigma = 2$, $\mu = 0.1$ with 60 iteration, $\alpha = 0.05$, $\beta = 0.3$

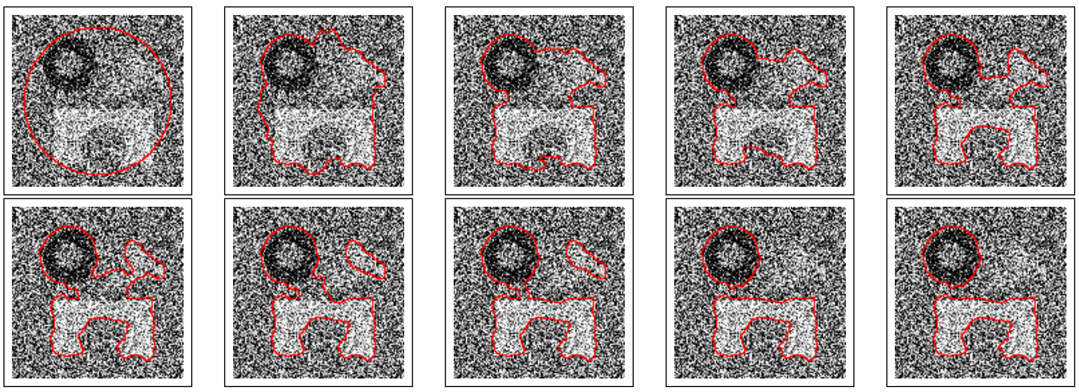
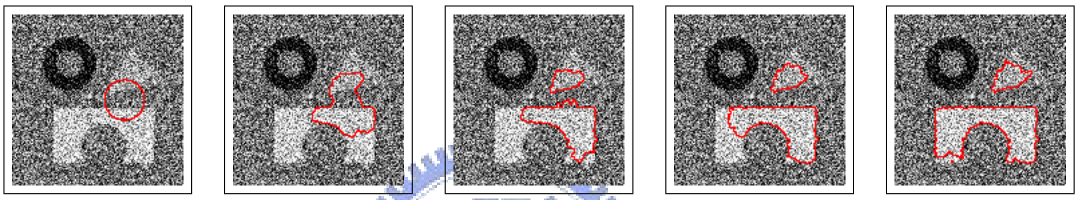
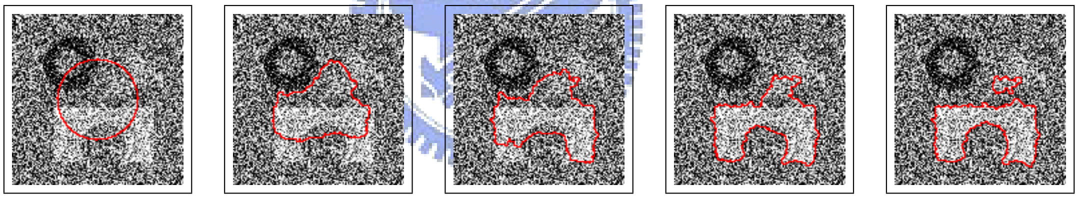


Fig. 4.17 Added 45% Gaussian noise, using EdgeFlow, 8 orientation $\sigma = 3$, $\alpha = 0.1$, $\beta = 0.2$



(a) Gaussian noise 30%



(b) Gaussian noise 45%

Fig. 4.18 Segmentation of the 30% and 45% Gaussian noise image using region based method

CHAPTER 5
CONCLUSIONS AND FUTURE WORKS

In this thesis, we survey the level set methods and their applications on image segmentation. We separate the conclusion discuss in two ways

level set methods

The level set methods are simpler to express the surfaces or curves, but they have the drawback of high time consuming on numerical PDE calculation. Since each implicit function ϕ can result in only one piece of segmentation, to get more segmentations we need more implicit functions to be initialized in different parts of an image. This also increases calculation time.

Image segmentation

The advantage we can find is its convenient for cutout the region after segmentation and the result of segmentation by curve is a complete region. The drawbacks of the level set methods in image segmentation are that:

First, the level set is an active contour implementation method and the segmentation results are based on the methods of the image attribute analysis, for instance, edge detection uses gradient of Gaussian to smooth images, texture analysis utilizes Gabor filter.

Second, the generated vector field and the driven curve are hardly to balance among the different velocities κ, F (Edge function) and $\vec{V}$ (External velocity field). The parameters need to be set differently case by case. Ideally, we hope we do not need to care about the parameters, α, β, of the level set curve evolution function and only need to care about the underlying designs of F and $\vec{V}$. For this reason, the region based method seems to give an idea of obtaining another form of the level set function of curve evolution.

Furthermore, to solve the parameter setting, designing a training system to get reasonable setting maybe workable.

REFERENCE

[1] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.

[2] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science*. Cambridge University Press, 1988.

[3] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vision*, vol. 22, pp. 61–79, 1997.

[4] C. Xu and J. L. Prince, "Gradient vector flow: A new external force for snakes," in *IEEE Pro. Conf. on Comp. Vis. Patt. Recog.*, 1997, pp. 66–71.

[5] B. Sumengen, B. S. Manjunath, and C. Kenney, "Image segmentation using curve evolution," in *Proc. Asilomar Conf. Signals, Syst. Computers*, vol. 2, 2001, pp. 1141–1145.

[6] T. F. Chan and L. A. Vese, "An active contour model without edges," in *Scale-Space Theories in Computer Vision*, 1999, pp. 141–151.

[7] C. W. Shu and S. Osher, "Efficient implementation of essentially non-oscillatory shock capturing schems," *J. Comput. Phys.*, vol. 77, pp. 439–471, 1988.

[8] V. Caselles, F. Catte, T. Coll, and F. Dibous, "A geometric model for active contours in image processing," *Numer. Math.*, vol. 66, no. 1, pp. 1–31, 1993.

[9] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Evolutionary fronts for topology-independent shape modeling and recovery," in *Proc. European Conf. Computer Vision*, 1994, pp. 3–13.

[10] C. Xu and J. L. Prince, "Snaks, shapes and gradient vector flow," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[11] W. Y. Ma and B. S. Manjunath, "Edgeflow: A technique for boundary detection and image segmentation," *IEEE Trans. Image Processing*, vol. 9, no. 8, pp. 1375–1388, 2000.

[12] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 8, pp. 837–842, 1996.

[13] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math*, vol. 42, pp. 577–685, 1989.

[14] T. F. Chan, B. Y. Sanberg, and L. A. Vese, "Active contours without edges for vector-valued images," *J. Visual Commun. and Image Rep.*, vol. 11, pp. 130–141, 2000.

[15] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[16] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002.