DEFORMABLE CONTOURS: MODELING, EXTRACTION, DETECTION AND CLASSIFICATION

by

KOK FUNG LAI

A thesis submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

1994

Acknowledgement

I wish to express my sincerely gratitude to

- my advisor, professor Roland Chin, for his constant support and trust,
- Professors Yu Hen Hu and Charles Dyer, for their assistance in the course of the research,
- Professors John Gubner, James Bucklew, William Sethares and Leonard Uhr, for their valuable input, and
- my wife, for her patience and love.

In addition, I wish to acknowledge the following organizations for their support

- National Computer Board, Republic of Singapore, for the award of postgraduate research scholarship, and
- Sinu Software Research Centre, Hong Kong University of Science and Technology, for assitance rendered.

DEFORMABLE CONTOURS: MODELING, EXTRACTION, DETECTION AND CLASSIFICATION

Kok Fung Lai

Under the supervision of Professor Roland T. Chin

at the University of Wisconsin-Madison

Abstract

This thesis presents an integrated approach in modeling, extracting, detecting and classifying deformable contours directly from noisy images. We begin by conducting a case study on regularization, formulation and initialization of the active contour models (snakes). Using minimax principle, we derive a regularization criterion whereby the values can be automatically and implicitly determined along the contour. Furthermore, we formulate a set of energy functionals which yield snakes that contain Hough transform as a special case. Subsequently, we consider the problem of modeling and extracting arbitrary deformable contours from noisy images. We combine a stable, invariant and unique contour model with Markov random field to yield prior distribution that exerts influence over an arbitrary global model while allowing for deformation. Under the Bayesian framework, contour extraction turns into posterior estimation, which is in turn equivalent to energy minimization in a generalized active contour model. Finally, we integrate these lower level visual tasks with pattern recognition processes of detection and classification. Based on the Nearman-Pearson lemma, we derive the optimal detection and classification tests. As the summation is peaked in most practical applications, only small regions need to be considered in marginalizing the distribution. The validity of our formulation have been confirmed by extensive and rigorous experimentations.

Table of Contents

Acknowledgement ii

Abstract iii

Table of Contents iv

1 Introduction 1

1.1 Motivation 1

1.2 Contributions 2

1.3 Organization 2

2 On Regularization, Formulation and Initialization of the Active Contour Models (Snakes) 4

2.1 Introduction 4

2.2 Regularization 7

2.3 Energy Formulation 12

2.4 Initialization 14

2.5 Experiments and Discussions 15

2.5.1 Real Images 15

2.5.2 Synthetic Images 16

2.5.3 Experimental Protocols 16

2.5.4 Effect of Sawtooth Angles, Snaxel Spacing and Noise 19

2.5.5 Comparative Studies on Regularization 20

2.5.6 Comparative Studies on E_{ext} Formulation 27
2.5.7 Effect of Scale 28
2.6 Conclusions 28

3 Deformable Contours: Modeling and Extraction 32

3.1 Introduction 32
3.2 Modeling Deformable Contours 35
3.2.1 Global Deformations 37
3.2.2 Local Deformations 39
3.3 Contour Extraction 40
3.3.1 Rigid Template 40
3.3.2 Deformable Template 42
3.4 Generalized Active Contour Models 43
3.4.1 Formulation 43
3.4.2 Minimization Algorithm 44
3.4.3 Basic and Stratified Line Search 46
3.4.4 Regularization 48
3.4.5 Initialization 49
3.5 Applications and Experiments 50
3.5.1 Training 50
3.5.2 Initialization and Extraction 51
3.5.3 Affine Invariant Contour Model 53
3.5.4 Comparison to Snake 53
3.5.5 Handwritten Characters 53
3.5.6 Severe Clutter and Deformation 56
3.5.7 Matching Performance 56
3.6 Conclusions 58

4 Deformable Contours: Detection and Classification 62

4.1 Introduction 62
4.2 Contour and Image Models 64
4.2.1 Representing Shape 64
4.2.2 Rigid and Deformable Contours 65
4.2.3 Modeling Deformation 65
4.2.4 Characteristics of Contour Model 67
4.2.5 Image Model 68
4.3 Detection and Classification 68
4.3.1 Detection 69
4.3.2 Classification 70
4.4 Issues in Practical Implementation 72
4.4.1 Summation Algorithm 72
4.4.2 Peaked Summation 74
4.4.3 Normalizing Constant 74
4.5 Experiments 76
4.5.1 Quantitative Comparisons 76
4.5.2 Numeral Recognition 79
4.6 Conclusions 83

5 Conclusions and Future Works **84**
5.1 Conclusions 84
5.2 Future Works 85
5.2.1 Time Sequence Analysis 85
5.2.2 Deformable Surface Models 86

References **88**

Chapter 1

Introduction

1.1 Motivation

Under the Marr's paradigm [31], the problems of contour modeling, extraction, detection and classification have conventionally been treated as separate issues in computer vision. While enormous amount of literature exists in each of the fields, these works have typically been developed in isolation. The following is a representative list: contour modeling [13, 33, 35, 48], edge detection [8, 11, 17, 30, 43], edge linking [3, 32, 34], contour detection and classification [18, 20, 36, 45].

This strict dichotomy of visual tasks has been widely adopted because it decomposes the problem of image analysis into independent and managable components. Unfortunately, the Marr's paradigm also results in unidirectional and irreversible flow of error from one component to the next. Moreover, due to image noise and the image projection process, the lower level visual tasks are generally ill-posed [37], i.e., they are underconstrained and so do not have unique solutions.

The classical and generalized Hough transforms [12, 4] are popular methods that combine modeling and extraction for *rigid* contours. Because these methods work by accumulating large number of votes, they are relatively insensitive to small vote fluctuation caused by noise and occlusion. In fact, if the superimposed noise is i.i.d. Gaussian, then the Hough transforms are simply efficient implementation of matched template, which is optimum in terms of detection error. These methods, however, do not scale very well to recognition in complex scenes where considerable outliers are present. In these cases, the probabilities of false positives can be very high [15, 16].

Furthermore, rigid templates used in Hough transforms cannot account for *deformations* which frequently arise from diversity and irregularity of shape. Since the degree of deformation is unknown in advance, a rigid template chosen *a priori* cannot produce satisfactory results for all cases. In fact, we will show in Chapter 3 of this thesis that its performance degrades with deformation.

Recently, numerous works in *deformable* templates have been reported, using weak models which deform in conformation to salient image features. Examples include the active contour model (snake) [21], parameterized templates [47], elliptic Fourier decompositions [41], implicit polynomials [42], handprinted character recognition [19] and affine invariant contour tracking [5]. In addition, several methods that classify deformable contours have also been reported [18, 45], but these ignore the problem of extraction. There have been relatively few methods that consider detection and classification of deformable contours directly from noisy images [19, 42].

1.2 Contributions

This thesis presents an integrated approach in modeling, extracting, detecting and classifying deformable contours directly from noisy images. We begin by conducting a close examinination on the active contour models of [21]. This yields enhanced understanding on the various aspects of deformable models which forms the corner stone of our subsequent works. We then proceed to formally develop the necessary theories that enable us to integrate the various visual tasks. The validity of our formulation has been confirmed through extensive experimentations on both real applications and synthetic images.

1.3 Organization

We organize this thesis as follows:

Chapter 2 presents a case study on regularization, formulation and initialization of the active contour models (snakes). In snake formulation, large regularization enhances the robustness against noise and incomplete data, while small values increase the accuracy in capturing boundary variations. We present a local minimax criterion which automatically determines the optimal regularization at every locations along the boundary with no added computation cost. In addition, we modify existing energy formulations to repair deficiencies in internal energy and improve performance in external energy. This yields snakes that contain Hough transform as a special case. We can therefore initialize the snake efficiently and reliably using Hough transform.

Chapter 3 considers the problem of modeling and extracting arbitrary deformable contours from noisy images. We propose a global contour model based on a stable and regenerative shape matrix, which is invariant and unique under rigid transformations. Combined with Markov random field to model local deformation, this yields prior distribution that exerts influence over a global model while allowing for deformation. We then cast the problem of extraction into posterior estimation and show its equivalence to energy minimization of a generalized active contour model. We discuss pertinent issues in shape training, minimax regularization, energy minimization, search strategies and initialization by generalized Hough transform. Finally, we present experimental results and compare its performance to rigid template matching.

Chapter 4 describes a formal strategy to detect and classify deformable contours directly from noisy images. Our formulation is based on the generalized active contour model developed in Chapter 3. The g-snake is suitable for these tasks because its shape representation is stable, unique, affine invariant and possesses metric properties. We derive the optimal detection and classification tests using the Nearmen-Pearson lemma. As the summation is peaked in most practical applications, only small regions need to be considered in marginalizing the distribution. Finally, we perform extensive experimentations and report significant improvement over matched template in handwritten numeral recognition.

Finally, Chapter 5 summarizes the work and suggests future research directions.

Chapter 2

On Regularization, Formulation and Initialization of the Active Contour Models (Snakes)

2.1 Introduction

Under the Marr's paradigm [31], boundary extraction has conventionally been treated as two independent problems: *edge detection* and *edge linking*. At the lowest level, an edge detector operates directly on the image intensities to produce an edge map marking the location, stength and direction of edge points. An edge linking algorithm then yields an ordering of successive edge points based on some predefined criterion functions such as continuity and connectivity.

The enormous amount of research in edge detection can be broadly classified into one of several categories: optimal filtering [8, 11], scale space [30], surface fitting [17] and cost minimization [43]. A great deal of literature also exists on edge linking [3, 32, 34]. Despite this tremendous effort, there is little consensus that the problem of boundary extraction has been solved. A problem with this approach is that errors made in edge detection propagate to edge linking without opportunity for correction. Moreover, the lower level visual tasks are generally ill-posed [37], i.e., they are underconstrained and so do not have unique solutions.

The *active contour models*, or *snakes*, were originally proposed by [21] as a regularization approach [44] to the ill-posed edge detection problem. A snake is an ordered set of points, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$ where each *snaxel* $\mathbf{v}_i$ is defined on the finite grid: $\mathbf{v}_i \in \mathbb{E} = \{(x, y) : x, y = 1, 2, \ldots, M\}$. A snake can be either *closed* or *open*, depending on whether the end points are connected.

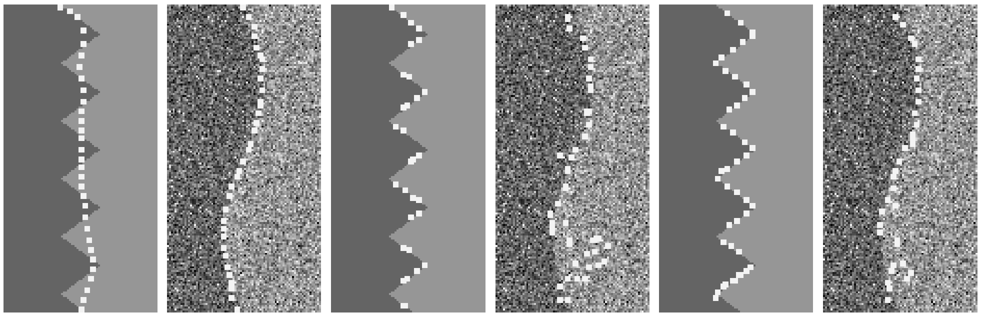
We denote two energy functionals $E_{int}, E_{ext} : \mathbb{E} \rightarrow [0, H]$, where $H < \infty$. While the internal energy E_{int} imposes continuity and smoothness constraints, the external energy E_{ext} attracts the snake to salient image features. The snake then turns boundary extraction into an energy minimization problem. Under this framework, one seeks to obtain $\mathbf{V}_\Lambda$ where

$$\mathbf{V}_\Lambda = \arg\min_{\mathrm{V}} \sum_{i=1}^{n} \lambda_i E_{int}(\mathbf{v}_i) + (1 - \lambda_i) E_{ext}(\mathbf{v}_i) \tag{2.1}$$

$\lambda_i \in [0, 1]$ are the regularization parameters that significantly influence the solution to (2.1). Setting $\lambda \gg (1 - \lambda)$ emphasizes regularization, yielding strongly model-driven solutions which are robust to noise. In contrast, small values of λ enable the snakes to effectively capture boundary discontinuities, but they also make the solutions very noise sensitive. Figure 2.1 illustrates these conflicting goals. Despite its significance, regularization selection were either ignored [6, 21, 29, 49], or based on empirical observations [2, 9] and heuristics [46].

Another factor which directly affects the solutions is energy formulation. The original E_{int} formulation [21] causes a snake to shrink around strong edge points. Ballooning force [9] and hard constraint on snaxel proximity [2] have been suggested, but these require prior information in setting the weighing factor or constraint. Moreover, existing curvature measure on E_{int} is not rotational and scale invariant, causing erroneous imposition of the smoothness constraint.

Snaxel initialization is the next factor with significant impact. For satisfactory results, most existing methods [2, 21, 46] require an operator to judiciously place the initial snaxels near desired boundaries. Alternatively, short snakes are initialized at locations of high intensity gradient and they either overlap [49] or extend in length [6] to cover a longer boundary. However, the latter strategies cause the snake to lose the globalities that enable it to interpolate subjective contours across boundary gaps and



(a) $\lambda = 0.8$ (b) $\lambda = 0.2$ (c) minimax

Figure 2.1: Effects of Regularization. (a) Snake fails to capture corners but robust to noise; (b) Snake capture corners but noise sensitive; (c) good compromise.

noisy segments.

This chapter tackles the fundamental issues on regularization, formulation and initialization of the active contour models. We made a number of contributions which are summarized below:

- *Automatic and implicit selection of local regularization parameters.* We present a parameter selection strategy based on the *minimax* criterion [14]. The resulting *local minimax criterion* automatically determines the optimal tradeoff at every location along the boundary of interest without any user input or heuristics. Moreover, this criterion can be implicitly incorporated into the energy computation process with no increase in computational cost.

- *Formulation of energy functionals.* We propose modifications in the formulations of the energy functionals to repair some existing deficiencies. We also generalize E_{ext} to incorporate the gradient direction. This ensures consistency between a snake's orientation to the underlying intensity profile.

- *Automatic initialization.* Using the reformulated energy functionals, we show that Hough transform [12] is a special case of a snake with large regularization parameters. Consequently, the Hough transform provides an obvious choice as the initial snake for optimization.

2.2 Regularization

We begin by considering global regularization where a uniform parameter λ is applied over an objective function with two terms E_{int} and E_{ext}. Denote $E_{int}(\mathbf{V}) = \sum_{i=1}^{n} E_{int}(\mathbf{v}_i)$ and $E_{ext}(\mathbf{V}) = \sum_{i=1}^{n} E_{ext}(\mathbf{v}_i)$, we seek $\mathbf{V}$ to minimize

$$e(\mathbf{V}, \lambda) = \lambda E_{int}(\mathbf{V}) + (1 - \lambda) E_{ext}(\mathbf{V}) \tag{2.2}$$

Suppose that in (2.2) the parameter λ is unknown because no prior information on the underlying variations on the boundary and the level of noise is available. Since values chosen *a priori* are unlikely to be satisfactory for all cases, it becomes necessary to define a criterion to determine λ. A viable criterion is to seek a $\mathbf{V}$ that minimizes, over all λ, the maximum of $e(\mathbf{V}, \lambda)$. This yields the *minimax* [14] criterion :

$$e(\mathbf{V}^*, \lambda^*) = \min_{\mathbf{V}} \max_{\lambda} \lambda E_{int}(\mathbf{V}) + (1 - \lambda) E_{ext}(\mathbf{V}) \tag{2.3}$$

where $\mathbf{V}^*$ and λ^* are, respectively, the minimax snake and regularization parameter.

As a function of λ, $e(\mathbf{V}, \lambda)$ is a straight line from $e(\mathbf{V}, 0) = E_{ext}(\mathbf{V})$ to $e(\mathbf{V}, 1) = E_{int}(\mathbf{V})$, as shown in Figure 2.2(a). Thus, for a fixed $\mathbf{V}$, the maximum value of $e(\mathbf{V}, \lambda)$ occurs at either $\lambda = 0$ or $\lambda = 1$ and this maximum value is $\max(E_{int}(\mathbf{V}), E_{ext}(\mathbf{V}))$. For each $\lambda \in [0, 1]$, let $\mathbf{V}_\lambda$ be the *optimum snake* that minimizes equation (2.2) corresponding to the chosen λ, i.e.,

$$\mathbf{V}_\lambda = \arg\min_{\mathbf{V}} e(\mathbf{V}, \lambda) \tag{2.4}$$

and let $\mathcal{U}(\lambda) = e(\mathbf{V}_\lambda, \lambda)$. Note that since $\mathbf{V}_\lambda$ are optimum, $\mathcal{U}(\lambda) = \min_{\mathbf{V}} e(\mathbf{V}, \lambda)$ is the minimum possible energy for λ. Moreover

Proposition 1 *$\mathcal{U}(\lambda)$ is a concave function.*

Proof. **Write**

$$\begin{aligned} e(\mathbf{V}, \alpha\lambda_1 + (1-\alpha)\lambda_2) &= \alpha \left[\lambda_1 E_{int}(\mathbf{V}) + (1-\lambda_1)E_{ext}(\mathbf{V})\right] + \\ &\quad (1-\alpha)\left[\lambda_2 E_{int}(\mathbf{V}) + (1-\lambda_2)E_{ext}(\mathbf{V})\right] \\ &= \alpha e(\mathbf{V}, \lambda_1) + (1-\alpha)e(\mathbf{V}, \lambda_2) \\ &\geq \alpha\mathcal{U}(\lambda_1) + (1-\alpha)\mathcal{U}(\lambda_2) \end{aligned} \tag{2.5}$$

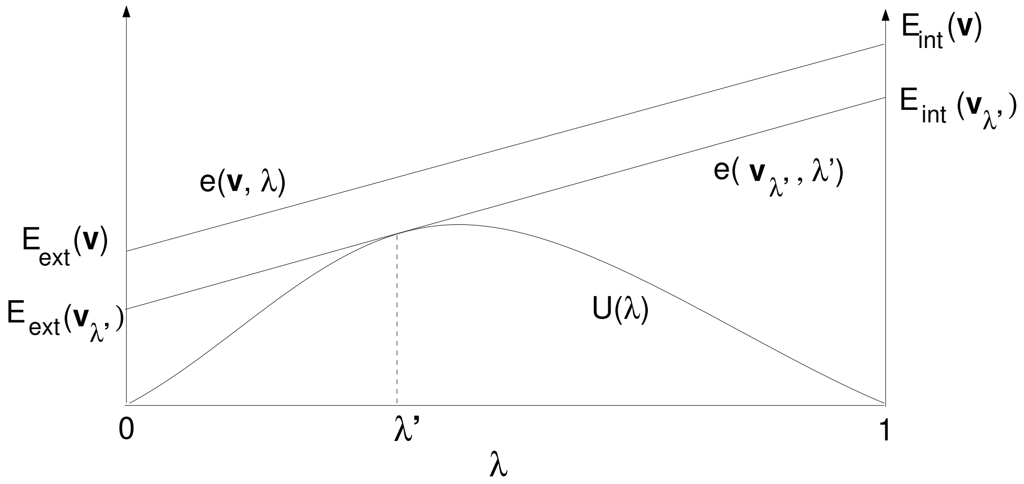
The inequality in (2.5) is true for all $\mathbf{V}$, hence

$$\begin{aligned} \mathcal{U}(\alpha\lambda_1 + (1-\alpha)\lambda_2) &= \min_{\mathbf{V}} e(\mathbf{V}, \alpha\lambda_1 + (1-\alpha)\lambda_2) \\ &\geq \alpha\mathcal{U}(\lambda_1) + (1-\alpha)\mathcal{U}(\lambda_2) \end{aligned}$$

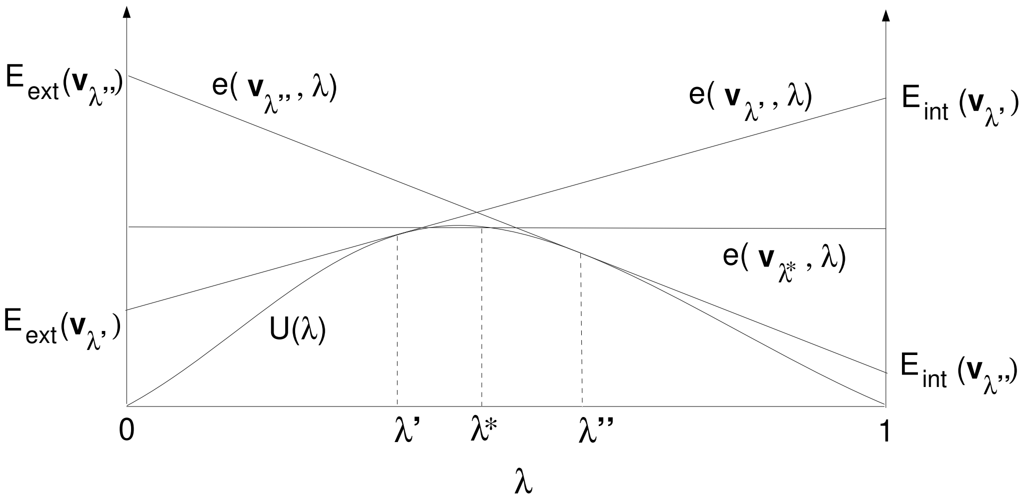
Suppose that $\mathcal{U}(\lambda)$ assumes a typical form as depicted in Figure 2.2(a). The line labeled $e(\mathbf{V}, \lambda)$ is both parallel to $e(\mathbf{V}_{\lambda'}, \lambda')$ and tangent to $\mathcal{U}(\lambda')$. Note that $\mathbf{V}$cannot be a minimax solution because the line $e(\mathbf{V}, \lambda)$ lies completely above $e(\mathbf{V}_{\lambda'}, \lambda')$ and thus has a larger minimum value. Since $e(\mathbf{V}_{\lambda'}, \lambda')$ touches $\mathcal{U}(\lambda)$ at $\lambda = \lambda'$, $\mathbf{V}_{\lambda'}$ is the optimum snake for parameter λ'. Because no snake can minimize (2.2) below $\mathcal{U}(\lambda)$, given any $\mathbf{V}$, we can always find a tangent line that has the smallest $e(\mathbf{V}, \lambda)$. Consequently only an optimum snake can be the minimax solution for this figure. Moreover, by examination of Figure 2.2(b), we see that the minimax solution is an optimum snake $\mathbf{V}_{\lambda^*}$ for a parameter λ^* that maximizes $\mathcal{U}(\lambda)$ over $\lambda \in [0, 1]$:

$$\min_{\mathbf{V}} \max_{\lambda} e(\mathbf{V}, \lambda) = \max_{\lambda} \min_{\mathbf{V}} e(\mathbf{V}, \lambda) \tag{2.6}$$

Since λ^* maximizes the minimum snake energy $\mathcal{U}(\lambda)$, a minimax snake is the optimum snake that minimizes the worst case energy. Note that in this case it also equalises the



(a)



(b)

Figure 2.2: (a) Illustration of functions $e(\mathbf{V}, \lambda)$ and $\mathcal{U}(\lambda)$; (b) Illustration of the minimax criterion when $\mathcal{U}$ has an interior maximum.

value of internal and external energy.

We give below a formal proof for (2.6) which includes the case when $\mathcal{U}(\lambda)$ does not assume the typical form i.e., $\max_\lambda \mathcal{U}(\lambda)$ occurs at $\lambda = 0$ or $\lambda = 1$.

Proposition 2 *Suppose that λ^* is a solution to $\mathcal{U}(\lambda^*) = \max_\lambda \mathcal{U}(\lambda)$ where $\lambda \in [0,1]$. Suppose further that either $\lambda^* = 0$ or $\lambda^* = 1$, or $E_{int}(\mathbf{V}_{\lambda^*}) = E_{ext}(\mathbf{V}_{\lambda^*})$. Then $\mathbf{V}_{\lambda^*}$ is a snake that satisfies the minimax criterion.*

Proof We consider first the case where $E_{int}(\mathbf{V}_{\lambda^*}) = E_{ext}(\mathbf{V}_{\lambda^*})$. For the parameter λ^*, we have

$$\max_{\lambda} \min_{\mathrm{V}} e(\mathbf{V}, \lambda) = e(\mathbf{V}_{\lambda^*}, \lambda^*) = e(\mathbf{V}_{\lambda^*}, \lambda) \tag{2.7}$$

where the first equality is by definition of $\mathcal{U}$, and the second equality follows from the fact that $e(\mathbf{V}_{\lambda^*}, \lambda)$ is constant in λ. Thus we have

$$\begin{aligned} \max_{\lambda} \min_{\mathrm{V}} e(\mathbf{V}, \lambda) &= \max_{\lambda} e(\mathbf{V}_{\lambda^*}, \lambda) \\ &\geq \min_{\mathrm{V}} \max_{\lambda} e(\mathbf{V}, \lambda) \end{aligned} \tag{2.8}$$

However, for each λ, we always have

$$\max_{\lambda} e(\mathbf{V}, \lambda) \geq \max_{\lambda} \min_{\mathrm{V}} e(\mathbf{V}, \lambda) \tag{2.9}$$

which implies

$$\min_{\mathrm{V}} \max_{\lambda} e(\mathbf{V}, \lambda) \geq \max_{\lambda} \min_{\mathrm{V}} e(\mathbf{V}, \lambda) \tag{2.10}$$

Combining (2.8) and (2.10), we have

$$\min_{\mathrm{V}} \max_{\lambda} e(\mathbf{V}, \lambda) = \max_{\lambda} \min_{\mathrm{V}} e(\mathbf{V}, \lambda) \tag{2.11}$$

and the left-hand equality of (2.7) implies

$$e(\mathbf{V}_{\lambda^*}, \lambda^*) = \min_{\mathrm{V}} \max_{\lambda} e(\mathbf{V}, \lambda) \tag{2.12}$$

which proves the proposition.

Now suppose that $\lambda^* = 0$. In this case it is straightforward to show that $\max_\lambda e(\mathbf{V}_{\lambda^*}, \lambda) = E_{ext}(\mathbf{V}_{\lambda^*}) = e(\mathbf{V}_{\lambda^*}, \lambda^*)$. This fact and the argument of equations (2.9) through (2.12) imply that λ^* is a minimax parameter. A similar argument holds for the case $\lambda^* = 1$. This completes the proof.

Note that the concavity of $\mathcal{U}(\lambda)$ ensures that its derivative[1], when exists, can only equal zero at the maximum. Thus $E_{int}(\mathbf{V}_{\lambda^*}) = E_{ext}(\mathbf{V}_{\lambda^*})$ whenever $\mathcal{U}(\lambda)$ has an interior maximum. Consequently, the above proposition covers all cases, except when $\mathcal{U}(\lambda)$ has an interior maximum at which it is not differentiable. Such case is rare in most applications; a complete proof can be found from page 22 to 27 in [38].

The results obtained thus far illustrate the intuitively pleasing characteristics of minimax regularization λ^* which minimizes the worst case energy in a *global* sense. However, global regularization sometimes causes over-smoothing of discontinuities in some parts of the boundary and excessive interpolation of noisy boundary in the rest, as evidenced in Figure 1. Hence we seek an alternative solution that will provide sufficient regularization to suppress the effect of noise while preserving boundary discontinuities, i.e., we wish to find $\mathbf{V}$ to minimize

$$e(\mathbf{V}, \Lambda) = \sum_{i=1}^{n} [\lambda_i E_{int}(\mathbf{v}_i) + (1 - \lambda_i) E_{ext}(\mathbf{v}_i)] \tag{2.13}$$

where $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ are the unknown *local* regularization parameters. We apply the arguments presented above and seek a *local* minimax solution $\mathbf{V}^*$ that minimizes,

[1] $\mathcal{U}'(\lambda) = E_{int}(\mathrm{V}_\lambda) - E_{ext}(\mathrm{V}_\lambda)$

over every λ_i, the maximum of $e(\mathbf{v}_i, \lambda_i)$. This yields the *local minimax criterion* :

$$e(\mathbf{V}^*, \Lambda^*) = \min_{\mathrm{V}} \sum_{i=1}^{n} \max(E_{int}(\mathbf{v}_i), E_{ext}(\mathbf{v}_i)) \tag{2.14}$$

where $\mathbf{V}^*$ and Λ^* are the solutions. Note from (2.14) that this criterion completely bypasses the explicit selection of the *nuisance* parameters Λ^*. It can therefore be incorporated into discrete energy minimization algorithms [2, 46] with no added computational cost.

Finally, we generalize the arguments to cases with multiple constituent terms:

$$e(\mathbf{V}, \Lambda) = \sum_{i=1}^{n} \Lambda_i^T \mathbf{E}(\mathbf{v}_i) \tag{2.15}$$

where $\mathbf{E}(\mathbf{v}_i) = [E_1(\mathbf{v}_i), E_2(\mathbf{v}_i), \ldots, E_p(\mathbf{v}_i)]^T$ and $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{ip}]^T$ are p-dimensional energy functionals and regularization parameters respectively. Let $\mathbf{1} = [1, 1, ..., 1]^T$. If Λ_i satisfies $\Lambda_i^T \mathbf{1} = 1$ for $i = 1, 2, \ldots, n$, then the local minimax criterion equalizes the energies and satisfies

$$e(\mathbf{V}^*, \Lambda^*) = \min_{\mathrm{V}} \sum_{i=1}^{n} \max_{j}(E_j(\mathbf{v}_i)) \tag{2.16}$$

As energies are equalized at local minimax, the criterion requires each energy functional to be properly normalized. The next section presents normalized energy formulations which have also undergone some enhancements.

2.3 Energy Formulation

In the original formulation of [21], the internal energy E_{int} was defined as the first and second derivatives of along the boundary, giving the snake rubber-sheet and thin-plate like behavior respectively. On the discrete grid, E_{int} was approximated as

$$E_{int}(\mathbf{v}_i) = \|\mathbf{v}_i - \mathbf{v}_{i-1}\|^2 + \|\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}\|^2 \tag{2.17}$$

This discrete approximation presents several problems in practical applications. Firstly, E_{int} vanishes when $\mathbf{v}_i = \mathbf{v}$ for all i, often causing a snake to shrink around strong edge points. To counter the shrinkage tendency, ballooning force [9] and hard constraint on snaxel proximity [2] have been suggested, but these require prior information in setting the weighing factor or constraint. Secondly, E_{int} in (2.17) is not invariant to scale change and rotation, thus often causing error in constraining smoothness on the snake.

It is important to realize that the primary purpose of E_{int} is to impose smoothness contraint. Therefore, in the absence of external force, E_{int} should force the snake to take the form of a *line* for an *opened* snake, or a *circle* for a *closed* snake. The objective can be achieved by defining, for $i = 2, 3, \ldots, n-1$,

$$E_{int}(\mathbf{v}_i) = \frac{1}{l(\mathbf{V})}\|\mathbf{v}_i - \alpha(\mathbf{v}_{i-1} + \mathbf{v}_{i+1})\|^2 \tag{2.18}$$

where $\alpha = 0.5$ or $\alpha = 0.5\cos^{-1}(2\pi/n)$ for opened and closed snakes respectively. For end points, we simply take $E_{int}(\mathbf{v}_1) = E_{int}(\mathbf{v}_2)$ and $E_{int}(\mathbf{v}_n) = E_{int}(\mathbf{v}_{n-1})$. The internal energy is normalized using the averaged snaxel distance $l(\mathbf{V})$:

$$l(\mathbf{V}) = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{v}_i - \mathbf{v}_{i-1}\|^2 \tag{2.19}$$

which makes E_{int} invariant to scale change and rotation.

In addition, we also generalize the external energy E_{ext} to include intensity gradient direction if necessary. Define image $\mathbf{f}$ as the function $\mathbf{f} : \mathbb{E} \to \mathbb{D}$. Depending on the data type, either $\mathbb{D} = \mathbb{R}$ (intensity image or edge magnitude) or $\mathbb{D} = \mathbb{R}^2$ (2 × 1 intensity gradient vector). We formulate the external energy as follows:

$$E_{ext}(\mathbf{v}_i) = 1 - \mathbf{h}_i^T \mathbf{f}(\mathbf{v}_i) \tag{2.20}$$

where $\mathbf{h}_i$ is the unit normal vector at $\mathbf{v}_i$, and $\|\mathbf{h}_i\| = 1$. For $\mathbb{D} = \mathbb{R}^2$, $\mathbf{h}_i$ can be adaptively determined by rotating the tangent direction of the contour by 90^o, i.e., let $\mathbf{t}_i$ represent the *tangent* vector at $\mathbf{v}_i$:

$$\mathbf{t}_i = \frac{\mathbf{v}_i - \mathbf{v}_{i-1}}{\|\mathbf{v}_i - \mathbf{v}_{i-1}\|} + \frac{\mathbf{v}_{i+1} - \mathbf{v}_i}{\|\mathbf{v}_{i+1} - \mathbf{v}_i\|} \tag{2.21}$$

then $\mathbf{h}_i$ is a unit vector which is normal to $\mathbf{t}_i$.

Unlike methods [6] which use gradient direction to discard spurious edge points, the proposed E_{ext} continuously weigh the gradient magnitude against any orientation inconsistency. Consequently, besides a non edge point, an edge point whose orientation disagrees with that of the overlaying snake may also yield large external energy. Hence E_{ext} is able to discriminate against phantom lines while allowing legitimate corners to develop.

Based on these reformulated energy functionals, we derive an efficient and reliable initialization strategy using the Hough transform, which is discussed next.

2.4 Initialization

Using the reformulated energy functionals, the optimum snake is a line for opened snake, and a circle for closed snake when λ is sufficiently closed to 1. Snakes that satisfy this structural constraints have $E_{int} = 0$. We can therefore ignore E_{int} in energy minimization to yield

$$\mathbf{V}^* = \arg\min_{\mathrm{V}} \sum_{i=1}^{n} E_{ext}(\mathbf{v}_i)$$

$$= \arg\max_{V} \sum_{i=1}^{n} \mathbf{h}_i^T \mathbf{f}(\mathbf{v}_i) \qquad (2.22)$$

The problem is then equivalent to finding a line or circle that maximizes the cross-correlation between $\mathbf{V}$ and the image. This suggests a counting procedure that can be implemented using Hough transform [12].

Since $e(\mathbf{V}, \lambda)$ is continuous in λ, the optimum snake $\mathbf{V}^*_{\lambda_1}$ must be located near $\mathbf{V}^*_{\lambda_2}$ if the difference $|\lambda_1 - \lambda_2|$ is small. Because the initialized snake using Hough transform has $\lambda \approx 1$, we can bring the snaxels into nearby optimal positions by relaxing the regularization parameters to their minimax values.

This initialization strategy exploits the efficiency and globality of Hough transform in the presence of noise and boundary gaps. Moreover, since the snaxels are initialized near optimal positions, only small search regions are needed in energy minimization.

2.5 Experiments and Discussions

In this section, we examine the performance of the proposed minimax regularization, energy formulation and method of initialization on a series of real and synthetic images. We employ a point-wise minimization algorithm for energy minimization which ensures convergence to local minima. This algorithm is a modification of the greedy algorithm [46]: it moves a snaxel $\mathbf{v}_i$ to a location that minimizes the sum of energy at $\mathbf{v}_{i-1}$, $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$, rather than just the energy in $\mathbf{v}_i$.

2.5.1 Real Images

Figure 2.3 shows the results obtained by applying the proposed realization on an image of body cells, an aerial photograph, and an image of soil particles. After initializing using Hough transform, regularization was selected implicitly using the local mininax criterion. These examples demonstrate that the snake successfully accomplished the given tasks of initializing the snaxels and converging to the desired boundaries, even when high degree of deformations are present.

2.5.2 Synthetic Images

The synthetic images used in these experiments consist of dark and bright portions separated by a sawtooth boundary as shown in Figure 2.4. The sawtooth angle, ϑ, controls the sharpness of the sawtooth boundary. To these perfect images we add zero mean gaussian white noise of variance σ. We then apply opened snakes on these images to detect the boundary. Using Hough transform, the initial snaxels lie on a straight line bisecting the sawtooth. The snaxels then converge to their final positions through energy minimization. From the converged results, we reconstruct the boundaries and compute the following two performance measures:

- $e_{avg}(\sigma)$: mean squared error between the ideal and reconstructed boundaries as a function of σ. A small $e_{avg}(\sigma)$ indicates that the reconstructed boundary replicates the ideal boundary with high accuracy.

- $e_{cor}(\sigma)$: mean squared error between the ideal and reconstructed boundaries at the *corners* as a function of σ. A small $e_{cor}(\sigma)$ indicates that the reconstructed boundary captures the discontinuities on the ideal boundary with high accuracy.

The following sections describe experiments that examine the effect of regularization, sawtooth angles, snaxel spacing and energy formulation on these performance measures. We begin by formally presenting the experimental protocols.

2.5.3 Experimental Protocols

Let $\mathbf{B} = \{(x, g(x))\}$ contains the set of points on the ideal boundary, and let $\mathbf{C} = \{(x_c, g(x_c))\} \in \mathbf{B}$ be the set of points that form the *corners* in $\mathbf{B}$. For the sawtooth boundary with parameters L and ϑ as shown in Figure 2.4, the sawtooth function $g(x)$

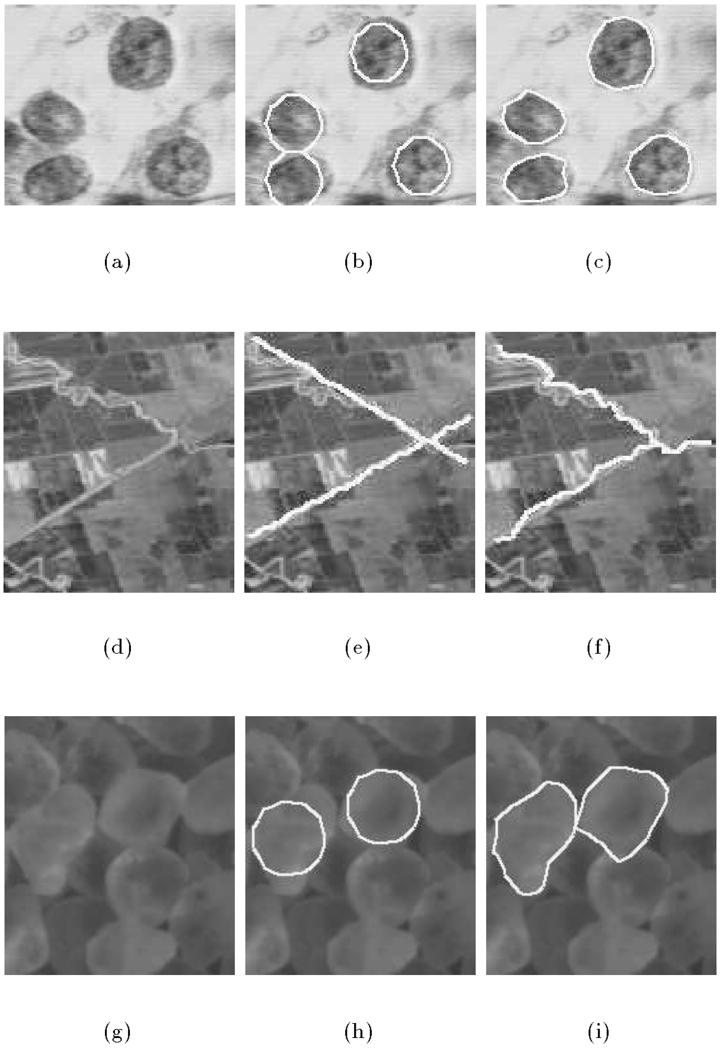
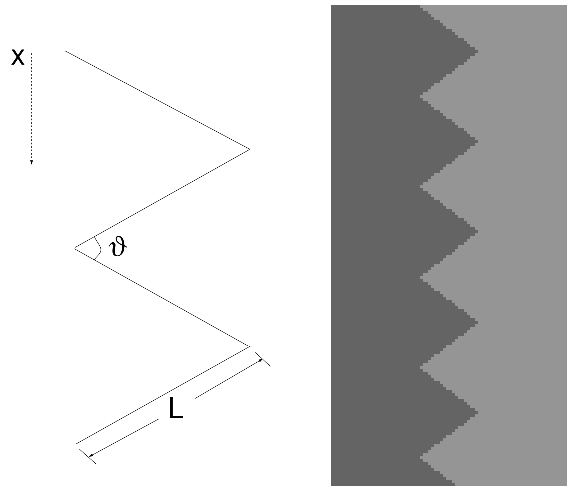


Figure 2.3: Some results obtained by applying the proposed realization on real images. (a) Image of body cells;(b) Initial positions given by Hough transform; (c) Converged results; (d) Aerial photograph; (e) Initial positions; (f) Converged results; (g) Image of soil particles; (h) Initial positions; (i) Converged results.



(a) (b)

Figure 2.4: Test pattern used in experiments. (a) Illustration of sawtooth's parameters; (b) Grey level image.

is defined as follows :

$$g(x) = \begin{cases} x\cot(\frac{\vartheta}{2}); & mT < x \leq (m+\frac{1}{2})T \\ g_0 - x\cot(\frac{\vartheta}{2}); & (m+\frac{1}{2})T < x \leq (m+1)T \end{cases} \tag{2.23}$$

where $g_0 = 2L\cos(\frac{\vartheta}{2})$, $T = 2L\sin(\vartheta/2)$ and $m = 0, 1, 2, \ldots\ldots.$

Let $n_\sigma(x, y)$ be a zero mean gaussian white noise process of variance σ^2. The following method synthesizes a noisy test image $I_\sigma(x, y)$ that contains a boundary described by **B**

$$I_\sigma(x,y) = \begin{cases} 1 + n_\sigma(x,y); & \text{if } y > g(x) \\ n_\sigma(x,y); & \text{otherwise} \end{cases} \tag{2.24}$$

Onto this $I_\sigma(x, y)$ an opened snake is released to detect the boundary. Let the converged snake be given by $\mathbf{v}_\sigma$, where

$$\mathbf{v}_\sigma = \{(x_i^\sigma, y_i^\sigma) : i = 1, 2, \ldots, n\} \tag{2.25}$$

Because the averaged snaxel spacing $D > 1$ in general, $\mathbf{v}_\sigma$ will contain less points than $\mathbf{B}$. Hence it is necessary to reconstruct a boundary from the converged snake to facilitate a meaningful comparison with the ideal boundary. Let the reconstructed boundary be given by $\hat{\mathbf{B}}_\sigma = \{(x, \hat{g}(x, \sigma))\}$. A simple reconstruction rule is to perform linear interpolation between successive snaxels, *i.e.*,

$$\hat{g}(x,\sigma) = \begin{cases} y_i^\sigma & \text{if } x = x_i^\sigma \\ \frac{x - x_{i-1}^\sigma}{x_i^\sigma - x_{i-1}^\sigma}[y_i^\sigma - y_{i-1}^\sigma] + y_{i-1}^\sigma & \text{if } x_{i-1}^\sigma < x < x_i^\sigma \end{cases} \tag{2.26}$$

We may obtain from $\hat{\mathbf{B}}_\sigma$ a corresponding set $\hat{\mathbf{C}}_\sigma = \{(x_c, \hat{g}(x_c, \sigma))\} \in \hat{\mathbf{B}}$ that contains the *estimated corners.*

Finally, we define the reconstruction performance measures, $e^2_{avg}(\sigma)$ and $e^2_{cor}(\sigma)$, as follows :

$$\begin{aligned} e^2_{avg}(\sigma) &= E[|\hat{g}(x,\sigma) - g(x)|^2] && (2.27) \\ e^2_{cor}(\sigma) &= E[|\hat{g}(x_c,\sigma) - g(x_c)|^2] && (2.28) \end{aligned}$$

2.5.4 Effect of Sawtooth Angles, Snaxel Spacing and Noise

These experiments investigate the performance of *local minimax criterion* for various sawtooth angles ϑ, snaxel spacing D, and level of noise σ. From the plots in Figure 7, we make the following observations :

- $e_{avg}(\sigma)$ and $e_{cor}(\sigma)$ increase with the level of noise σ. In addition, with all other parameters fixed, $e_{cor}(\sigma) > e_{avg}(\sigma)$ in general. This is caused by higher values of internal energy E_{int}, and also higher degree of uncertainties in intensity gradient estimation (and hence E_{ext}) near corners. These factors translate into higher mean squared errors $e_{cor}(\sigma)$ at these locations.

- Figure 2.5(a) and 2.5(b) show the plots of $e_{avg}(\sigma)$ and $e_{cor}(\sigma)$ for different sawtooth angles ϑ, with initial snaxel spacing fixed at $D = 5$. The errors increase with the degree of sharpness. This is a direct consequence of smoothness constraints imposed in the form of internal energy.

- Figure 2.5(c) and 2.5(d) shows the plots of $e_{avg}(\sigma)$ and $e_{cor}(\sigma)$ for snakes initialized with different snaxel spacing D, with ϑ fixed at 90^o. These experiments examine the effect of information content to the reconstruction performance. Since the snaxel spacing is inversely proportional to the number of snaxels available to reconstruct the boundary, larger D indicates that less information is available for boundary reconstruction. Generally, to yield small error measures, the snaxels must be distributed evenly on significant points along the boundary. The local minimax criterion only suffers small degradation in performance when D is increased. As will be evidenced later, this degradation is less severe when compared to snakes with fixed regularization.

2.5.5 Comparative Studies on Regularization

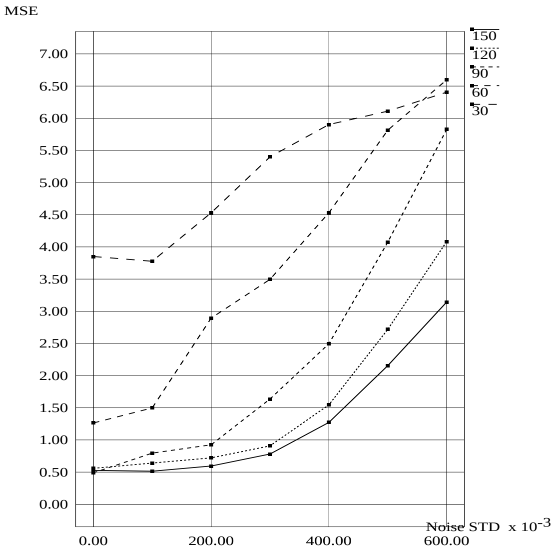
In this section, we compare the performance measures of snakes with fixed regularization, i.e.,

$$\mathbf{V}_\lambda = \arg\min_{\mathrm{V}} \sum_{i=1}^{n} [\lambda E_{int}(\mathbf{v}_i) + (1-\lambda) E_{ext}(\mathbf{v}_i)] \tag{2.29}$$

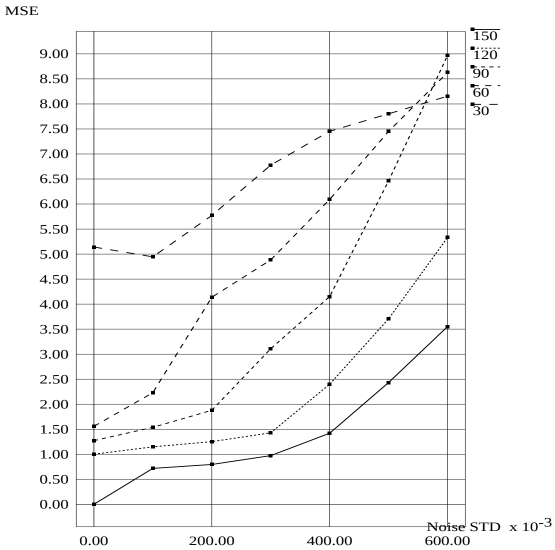
for $\lambda = 0.2$, $0.4, 0.6$ and 0.8, to that of local minimax criterion, i.e.,

$$\mathbf{V}^* = \arg\min_{\mathrm{V}} \sum_{i=1}^{n} \max(E_{int}(\mathbf{v}_i) E_{ext}(\mathbf{v}_i)) \tag{2.30}$$

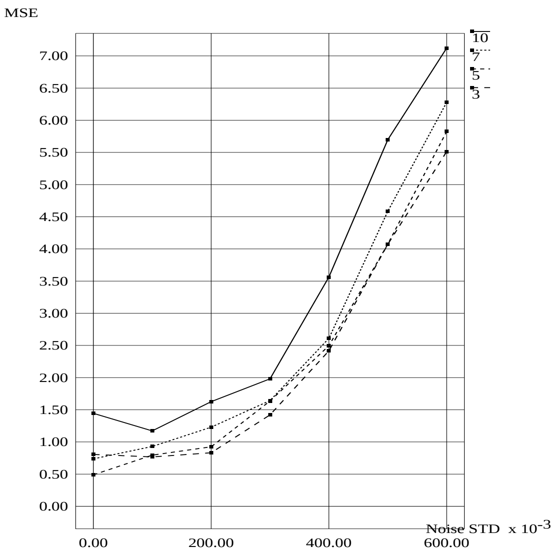
and present the results in Figures 2.6 through 2.9.



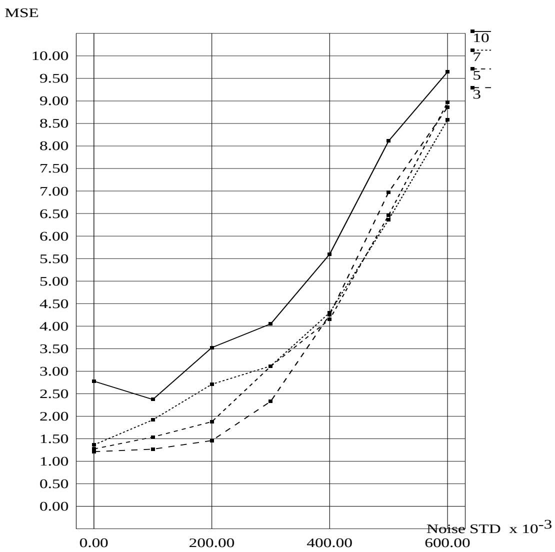
(a) $e_{avg}(\sigma)$ of *minimax* regularization for various sawtooth angle ϑ, $D = 5$.



(b) $e_{cor}(\sigma)$ of *minimax* regularization for various sawtooth angle ϑ, $D = 5$.



(c) $e_{avg}(\sigma)$ of *minimax* regularization for various snaxel spacing D, $\vartheta = 90^o$.



(d) $e_{cor}(\sigma)$ of *minimax* regularization for various snaxel spacing D, $\vartheta = 90^o$.

Figure 2.5: Effect of sawtooth angle ϑ and snaxel spacing D on $e_{avg}(\sigma)$ and $e_{cor}(\sigma)$ of *minimax* regularization. (a) (b) varying ϑ with $D = 5$, (c) (d) varying D with $\vartheta = 90^o$.

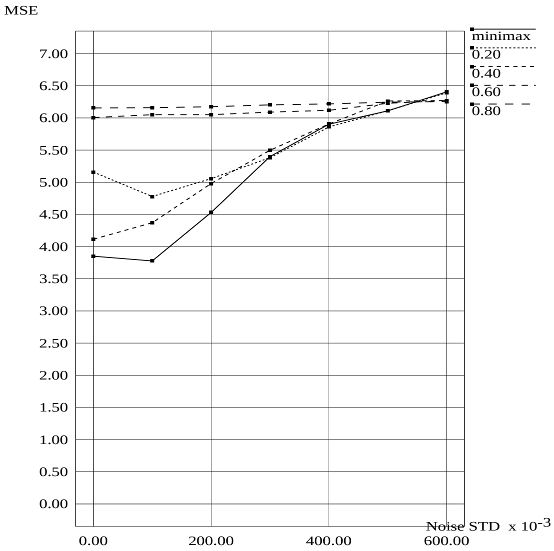
Figure 2.6 and 2.7 show the effect of sawtooth angles ϑ on the performance measures. Strong regularization, or higher values of λ (*e.g.* $0.8, 0.6$), generate higher resistant to bending in general. This results in snakes that yield large errors for sharp sawtooth angles. In fact one can observe from Figure 9 that $e_{cor}(\sigma)$ are always higher for snakes with larger λ. However, these snakes are also more robust and suffer less severe deterioration in performance with noise. As can be seen from Figure 8, snakes with $\lambda = 0.8$ almost always generate smaller $e_{avg}(\sigma)$ compared to other values of λ.

On the contrary, smaller values of λ (*e.g.*, $0.2, 0.4$) allows the snakes to more accurately capture the corners and thus yield smaller errors when noise is low. Their performance, however, degrades rapidly with noise due to lack of regularization. This produces very large $e_{avg}(\sigma)$ for noisy images, as shown in Figure 2.6.

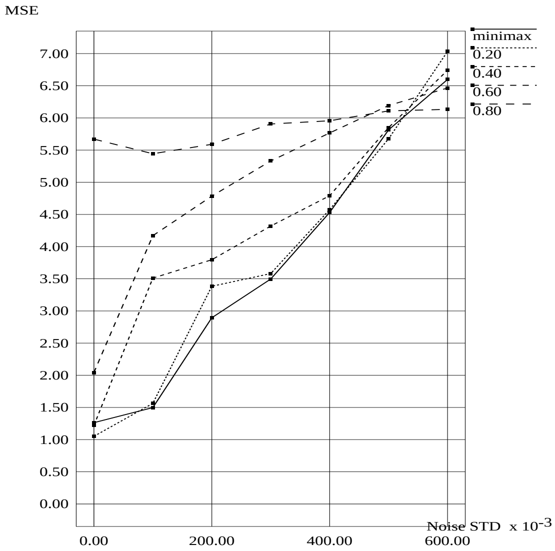
The local minimax criterion exhibits good compromise in these cases. It provides the capability to capture corner discontinuities and only suffers moderate degradation with noise. Compared to snakes with fixed regularization, its performance is the best for sharp sawtooth angles and acceptable for very noisy images.

Figure 2.8 and 2.9 show the effect of initial snaxel spacing D on the performance measures, with ϑ fixed at 90^o. As D is increased, snakes with fixed λ suffer relatively rapid degradation in performance. The local minimax criterion, however, is robust and better capable to account for this change. Consequently, while there may be little difference in performance for small D, the local minimax criterion clearly outperforms all other parameters for large D.

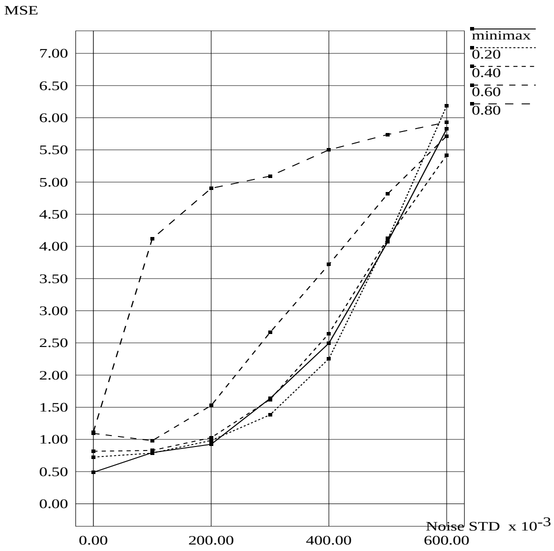
These comparisons confirm that the local minimax criterion yields robust performance under varying conditions of noise, boundary discontinuities and snaxel spacing. Its overall performance measures are superior when compared to snakes with regularization parameters selected *a priori*. Thus we conclude that it is a viable criterion when no prior knowledge on noise and geometrical structure of the boundary of interest is available.



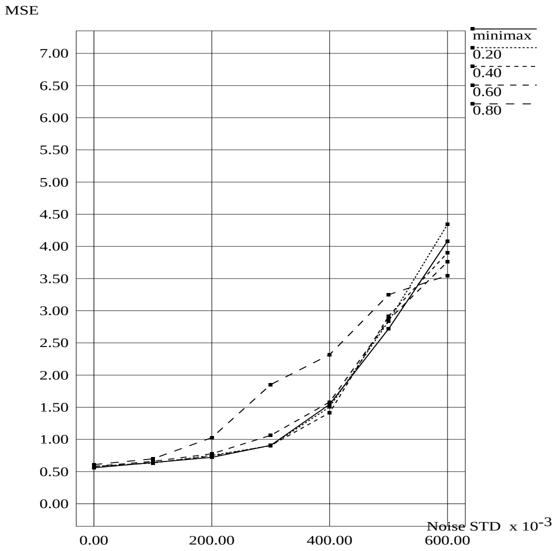
(a) $\vartheta = 30^o$: $e_{avg}(\sigma)$ for various λ



(b) $\vartheta = 60^o$: $e_{avg}(\sigma)$ for various λ

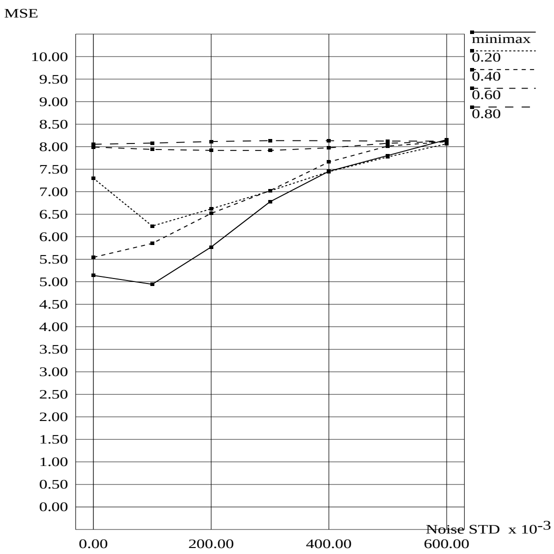


(c) $\vartheta = 90^o$: $e_{avg}(\sigma)$ for various λ

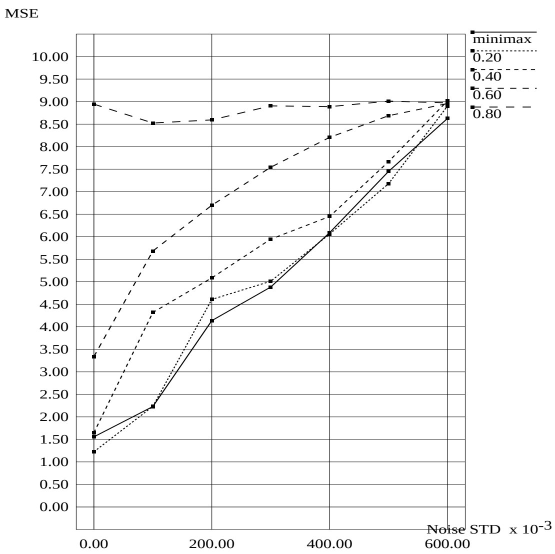


(d) $\vartheta = 120^o$: $e_{avg}(\sigma)$ for various λ

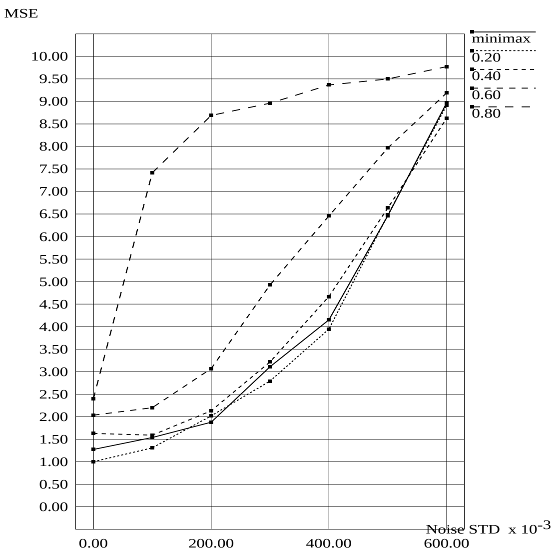
Figure 2.6: Plots of $e_{avg}(\sigma)$ for various λ with different sawtooth angles, $\vartheta = 30^o$, 60^o, 90^o, 120^o. $D = 5$, $L = 20$.



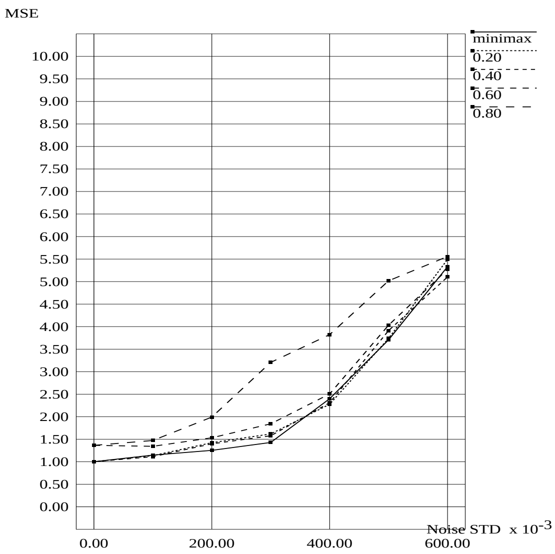
(a) $\vartheta = 30^o$: $e_{cor}(\sigma)$ for various λ



(b) $\vartheta = 60^o$: $e_{cor}(\sigma)$ for various λ

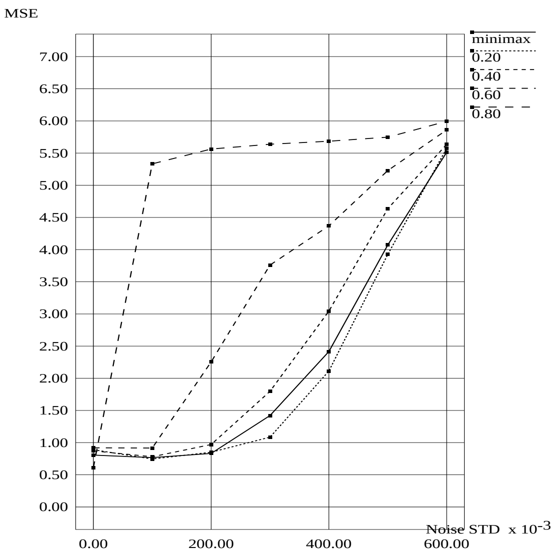


(c) $\vartheta = 90^o$: $e_{cor}(\sigma)$ for various λ

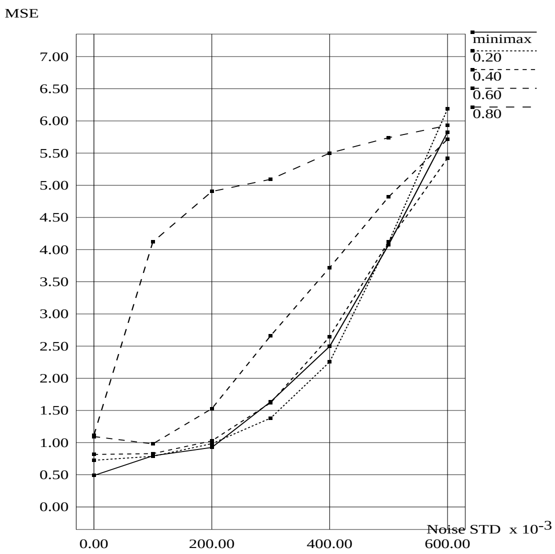


(d) $\vartheta = 120^o$: $e_{cor}(\sigma)$ for various λ

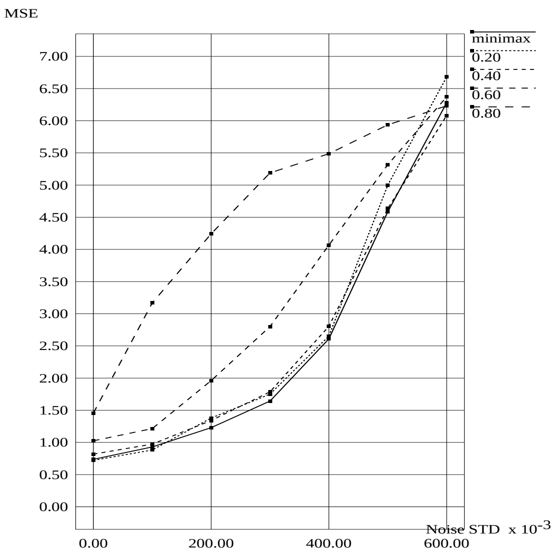
Figure 2.7: Plots of $e_{cor}(\sigma)$ for various λ with different sawtooth angles, $\vartheta = 30^o$, 60^o, 90^o, 120^o. $D = 5$, $L = 20$.



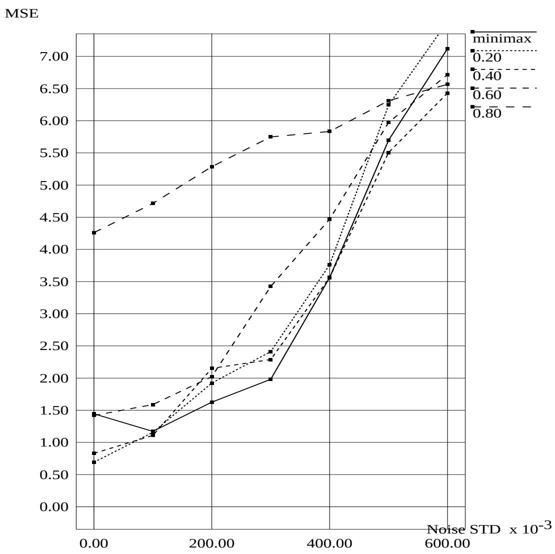
(a) $D = 3$: $e_{avg}(\sigma)$ for various λ



(b) $D = 5$: $e_{avg}(\sigma)$ for various λ

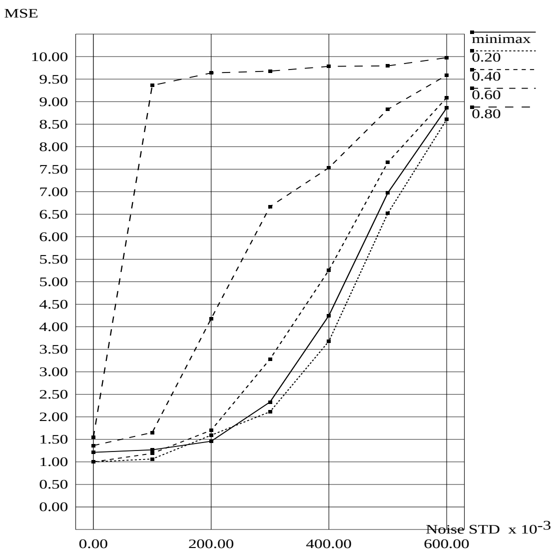


(c) $D = 7$: $e_{avg}(\sigma)$ for various λ

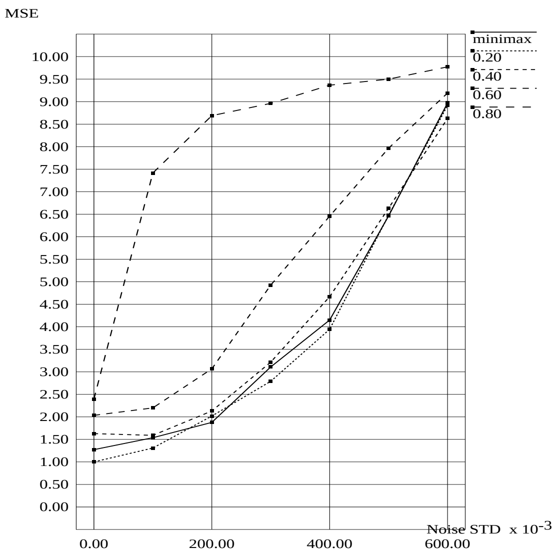


(d) $D = 10$: $e_{avg}(\sigma)$ for various λ

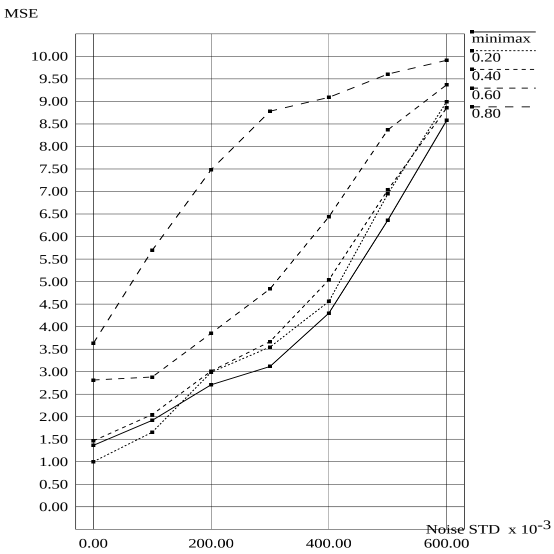
Figure 2.8: Plots of $e_{avg}(\lambda)$ for various λ with different snaxel spacings, $D = 3, 5, 7, 10$. $\vartheta = 90^o$, $L = 20$.



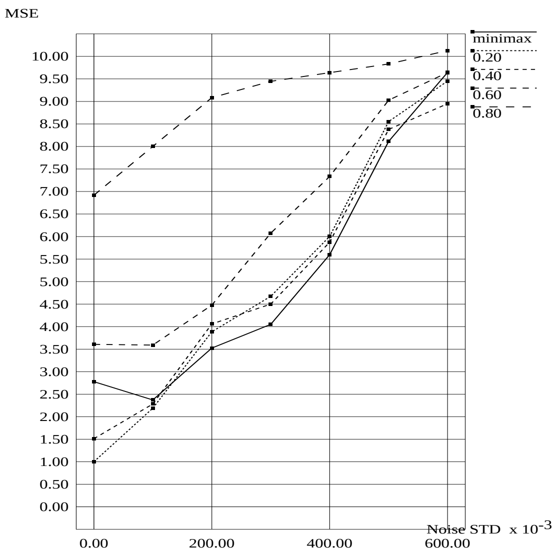
(a) $D = 3 : e_{cor}(\sigma)$ for various λ



(b) $D = 5 : e_{cor}(\sigma)$ for various λ



(c) $D = 7 : e_{cor}(\sigma)$ for various λ



(d) $D = 10 : e_{cor}(\sigma)$ for various λ

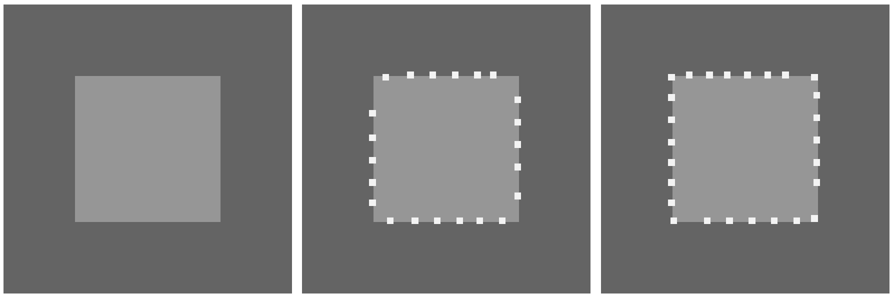
Figure 2.9: Plots of $e_{cor}(\sigma)$ for various λ with different snaxel spacings, $D = 3, 5, 7, 10$. $\vartheta = 90^o$, $L = 20$.

2.5.6 Comparative Studies on E_{ext} Formulation

These experiments examine the effect of incorporating gradient direction on E_{ext}. Recall that

$$E_{ext}(\mathbf{v}_i) = 1 - \mathbf{h}_i^T \mathbf{f}(\mathbf{v}_i) \tag{2.31}$$

we compare the performance measure for $\mathbf{h}_i, \mathbf{f}(\mathbf{v}_i) \in \mathbb{R}$ (gradient magnitude only) to that of $\mathbf{h}_i, \mathbf{f}(\mathbf{v}_i) \in \mathbb{R}^2$ (gradient magnitude and direction). Qualitatively, the primary effect of the ommision of gradient direction is the smoothing of corners, as depicted in Figure 2.10.



(a) (b) (c)

Figure 2.10: Effect of intensity gradient directions on detected boundaries. (a) Image of a square; (b) Converged snaxels without reformulating E_{ext} (c) Converged snaxels after reformulation.

We obtain $e_{cor}(\sigma)$ from snakes with gradient direction omitted and present the plots in Figure 2.11(a) and (b). These $e_{cor}(\sigma)$ exhibit similar charateristics to the effect of sawtooth angles ϑ and initial snaxel spacings D as before. However, its performance with varying noise is quite different: the $e_{cor}(\sigma)$ remain constant for low noise levels and then deterioriate rapidly with noise.

Figure 2.11(c) and (d) shows the difference in errors, $\Delta e_{cor}(\sigma)$, of the two formulations. Positive differences indicate improvements due to reformulation. We can observe substantial improvements for cases involving sharp sawtooth angle and large snaxel spacing. Moreover, at high levels of noise, gradient direction provides the discriminating power needed to avoid noisy interpolation and so yields lower errors. In conclusion, these comparisons confirm that the reformulated E_{ext} has indeed improved the overall performance.

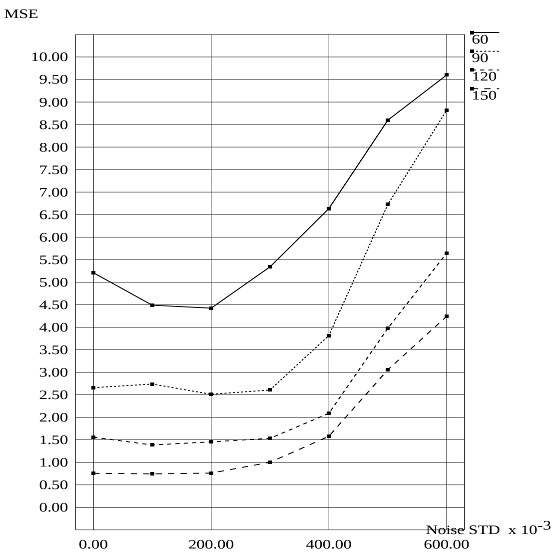
2.5.7 Effect of Scale

All the experiments described above used sawtooth images with parameter $L = 20$. Because in general $D > 1$, the snakes required less number of snaxels then the actual length of the sawtooth for reconstruction. For instance, if $D \approx 5$, then only 8 snaxels were needed for reconstruction for one period of sawtooth. This shows that the snakes only require small amount of information for boundary reconstruction.

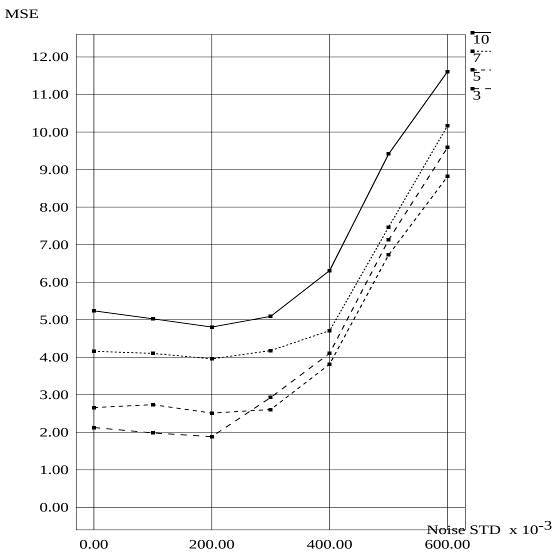
As the reformulated snakes exhibit scale invariance, it is unnecessary to obtain performance measures for other values of L. Figure 2.12 shows some examples of applying the snakes on sawtooth images with $L = 10$, 20, and 30. The snakes converged to positions as shown, with averaged snaxel spacing $D = 3.3$, 5.1, and 7.2. The resulting minimax energy were 0.18, 0.13 and 0.13 respectively. While all converged to equivalent solutions, the larger sawtooths required bigger search windows and hence caused noticeable increased in computational time. We used windows with sizes 9×9, 15×15 and 21×21 in these examples and obtained convergence times of approximately 1.1, 3.6 and 8.0 seconds respectively for one period of sawtooth.

2.6 Conclusions

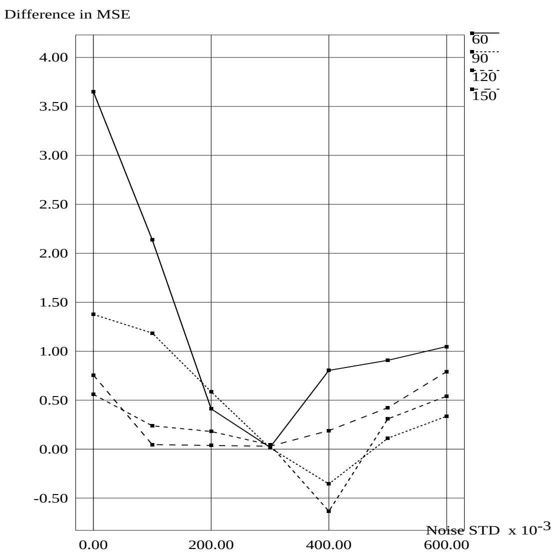
We considered the effect of regularization in the active contour models and showed that one can control the characteristics of the solutions by choosing appropriate values of λ. While large values of λ provide the regularization needed for robustness against noise and gaps, small values of λ allow the snake to capture boundary variations more accurately. Moreover, one can make λ a function of position to account for local variations



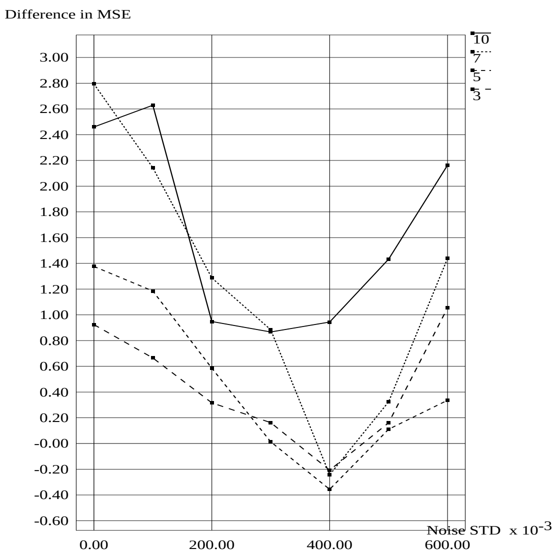
(a) Using only gradient magnitude in E_{ext}: $e_{cor}(\sigma)$ for various ϑ, $D = 5$.



(b) [Using only gradient magnitude in E_{ext}: $e_{cor}(\sigma)$ for various D, $\vartheta = 90^o$.

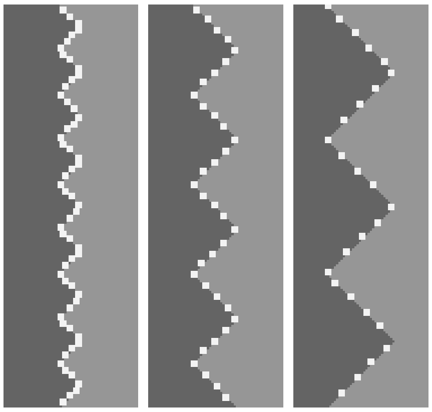


(c) $\Delta e_{cor}(\sigma)$ for case (a)



(d) $\Delta e_{cor}(\sigma)$ for case (b)

Figure 2.11: Effect of gradient direction on $e_{cor}(\sigma)$. (a) (b) $e_{cor}(\sigma)$ without gradient direction, under varying ϑ and D; (c) (d) Difference in MSE, $\Delta e_{cor}(\sigma)$, for two formulations of E_{ext}, positive differences indicate improvements.



(a) (b) (c)

Figure 2.12: Results of applying snake on scaled sawtooth images. (a) $L = 10$, $D = 3.3$, (b) $L = 20$, $D = 5.1$, (c) $L = 30$, $D = 7.2$.

of noise and geometrical structure along the boundary of interest. Nonetheless, without prior knowledge on noise and the geometrical structure of the boundary of interest, it is very difficult to determine these parameters: a global regularization parameter chosen *a priori* often results in over-smoothing in parts of the boundary and excessive interpolation of noisy boundary in the rest.

We defined a *local minimax criterion* whereby values of λ can be automatically and implicitly determined without user input or heuristics. We presented mathematical derivation of the criterion, and explained its intuitively pleasing charateristics of optimizing the worst case behaviour. The end results call for an energy formulation that completely bypasses the explicit selection of the nuisance parameters λ_i.

Subsequently, we formulated a set of energy functionals that impose constraints on equal snaxel spacing, smoothness, edge strength and consistency. The reformulated E_{int} is invariant in scale and rotation, yielding a snake that contains the Hough transform as a special case. This allows the use of the Hough transform to determine the initial snaxel positions efficiently and reliably from a global perspective. In addition, the reformulated

E_{ext} incorporates information on gradient direction to evaluate the consistency of a snake's local orientation to its underlying intensity profile. This improves the reliability and effectiveness of the snake in boundary detection.

Finally, we tested and examined the performance of the proposed minimax regularization, energy formulation and method of initialization on a series of real and synthetic images. Using real images, we demonstrated that the proposed realization successfully and efficiently accomplished the given tasks of initializing the snaxels and converging to the desired boundaries. Furthermore, through extensive experimentations on synthetic images, we obtained performance measures that were noticebly superior to existing methods.

Chapter 3

Deformable Contours: Modeling and Extraction

3.1 Introduction

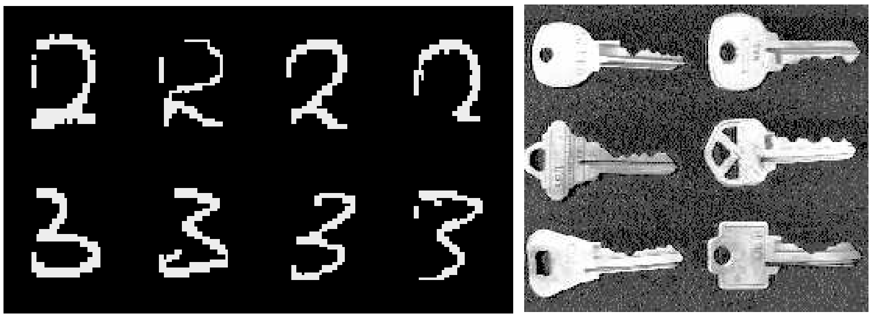
The problems of contour modeling and extraction have conventionally been viewed as two separate issues in computer vision. On one hand, the works in contour modeling [13, 33, 35, 45, 48] typically ignore the problem of extraction. Thus the starting point of modeling usually consists of segmented (binary) images, or an ordered set of points forming the contour. On the other hand, the two essential tasks in extraction, edge detection [8, 11, 30] and edge linking [3], are usually attempted without reference to a contour model.

Such strict dichotomy of visual tasks is charateristic of Marr's paradigm [31], widely adopted because it decomposes the problem of image analysis into independent and managable components. Unfortunately, this paradigm also results in unidirectional and irreversible flow of error from one component to the next. Moreover, due to image noise and the image projection process, edge detection and linking are ill-posed [37] problems, i.e., they are underconstrained and so do not have unique solutions.

We submit that it is fundamentally questionable to treat contour modeling and extraction as two separate issues. To model a contour, one must first extract it from an image; but without prior knowledge of the contour of interest, extraction is an ill-posed problem. In existing literature dealing with contour modeling, specific knowledge is sometimes used to yield satisfactory extraction. See [33] for an example. However, tailored to specific applications, such techniques are seldom transferrable.

The Hough transform [4, 12] is a well-known method that combines contour modeling and extraction for *rigid* objects. Choosing a model *a priori*, one accumulates local evidence of its presence by incrementing cells in the appropriate parameter space. Because this method works by accumulating large number of votes, it is relatively insensitive to small vote fluctuation caused by noise and occlusion. In fact, if the superimposed noise is Gaussian and white, then the Hough transform is simply an efficient implementation of the *matched template*, which is optimum in terms of detection error.

Rigid templates, however, cannot account for *deformations* which frequently arise from diversity and irregularity of shape. See Figure 3.1 for examples. Since the degree of deformation is unknown in advance, a rigid template chosen *a priori* cannot produce satisfactory results for all cases. In fact, we will show that its performance degrades with deformation.



(a) Handwritten characters (b) Keys

Figure 3.1: Examples of object deformation and shape irregularities

Deformable templates, in contrast, employ weak models which deform in conformation to salient image features. Mathematically, this translates into an objective function that consists of two different terms: one measures deviation from an ideal contour, and the other evaluates the match of the deformed contour with the underlying image. Contour extraction is then formulated as an optimization problem to maximize or minimize the overall objective function.

Examples of deformable template include the active contour model (snake) which uses smoothness constraints to restrict its solutions to controlled continuity splines [21]; parameterized templates for facial feature extraction [47]; elliptic Fourier decomposition for objects with shape irregularities [41]; implicit polynomials for curve and surface modeling [42]; movable control points for hand-printed character recognition [19]; and affine-invariant contour tracking [5].

With the exception of [5], these methods typically consider only global [41, 42, 47] or local [19, 21] deformations. While global templates involve large structural interactions and contain fewer parameters to be optimized, these global parameters cannot exercise local control along the contour and their physical meaning is sometimes obscure. In contrast, local models such as snakes contain more parameters and exert local control, but they are ill-suited for incorporation of global contour models.

This chapter considers the problem of modeling and extracting arbitrary deformable contours from noisy images. Our method is general and capable of representing any arbitrary shape, accounts for global changes due to rigid motions, and retains ability for local control. The contour model is based on a stable and regenerative shape matrix which is invariant and unique under rigid motions. Combined with the local characteristics of the Markov random field to model local deformations, this yields a prior distribution that exerts influence over a global model while allowing for deformations.

Our approach differs from [5] which achieves affine invariance by creating a subspace containing all allowable transformations of a memorized template. Besides possessing the added advantage of being unique, affine invariance of shape matrix is *implicit* and does not require exhaustive comparison into a created subspace. Furthermore, our approach utilizes the Markov random field for local deformations, thus providing a more realistic and effective model for highly variable but locally predictable objects.

Subsequently, we present an image model and obtain the probability distribution given that a contour is present. Under a Bayesian framework, the problem of extracting contours with unknown deformation from noisy images turns into maximum *a posteriori* estimation. This is equivalent to minimizing the energy of a generalized active contour model (g-snake), which is a generalization of the snake of [21]. We discuss pertinent issues in g-snakes, including shape training, energy minimization, line search strategies, minimax regularization and initialization by generalized Hough transform. Finally, we demonstrate with experiments the effectiveness and limitations of g-snakes in various applications and compare its matching performance to rigid template.

3.2 Modeling Deformable Contours

We define a contour as the vector containing an ordered set of points, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$. Each $\mathbf{v}_i$ is defined on the finite grid: $\mathbf{v} \in \mathbb{E} = \{(x, y) : x, y = 1, 2, \ldots, M\}$, thus $\mathbf{V} \in \mathbb{E}^n$.

Denote $\mathbf{U} \in \mathbb{E}^n$ where each $\mathbf{u}_i = \mathbf{v}_i - \mathbf{g}$ represents the displacement from an arbitrary reference point $\mathbf{g}$.

Each $\mathbf{u}_i$ can be expressed as a linear combination of two linearly independent vectors. For modeling of highly variable but locally predictable contours, the two neighboring points are suitable as its basis:

$$\mathbf{u}_i = \alpha_i \mathbf{u}_{i_\alpha} + \beta_i \mathbf{u}_{i_\beta} \tag{3.1}$$

where the basis indices are given by:

$$i_\alpha = \begin{cases} i-1; & i > 1 \\ 3; & i = 1 \end{cases} \qquad i_\beta = \begin{cases} i+1; & i < n \\ n-2; & i = n \end{cases} \tag{3.2}$$

We say that $\mathbf{U}$ is a *nontrivial* contour if $|x_{i_\alpha} y_{i_\beta} - x_{i_\beta} y_{i_\alpha}| > 0$ for $1 \leq i \leq n$. An example of a trivial contour is a straight line. More generally, the nontrivial condition ensures that $\mathbf{u}_{i_\alpha}$ and $\mathbf{u}_{i_\beta}$ are linearly independent for all i.

Collecting and rearranging similar equations for all i, we obtain the *shape equation*

$$\mathbf{A}\mathbf{U}^T = \mathbf{0} \tag{3.3}$$

where $\mathbf{A}$ is an $n \times n$ *shape matrix* that contains the necessary information to describe a shape:

$$\mathbf{A} = \begin{bmatrix} 1 & -\beta_1 & -\alpha_1 & 0 & \dots & 0 \\ -\alpha_2 & 1 & -\beta_2 & 0 & 0 & \dots \\ 0 & -\alpha_3 & 1 & -\beta_3 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -\alpha_{n-1} & 1 & -\beta_{n-1} \\ 0 & 0 & \dots & -\beta_n & -\alpha_n & 1 \end{bmatrix}$$

$\mathbf{A}$ is *regenerative*: the complete chain $\mathbf{U}$ can be generated from $\mathbf{A}$ by appropriate inversion if any two points on $\mathbf{U}$ are given. This is possible because $\text{rank}(\mathbf{A}) = n - 2$, and so an $(n-2) \times (n-2)$ invertible submatrix always exists within $\mathbf{A}$. To show this, we write, without loss of generality:

$$\mathbf{A}_r\mathbf{U}_r + \mathbf{b} = \mathbf{0} \tag{3.4}$$

where $\mathbf{A}_r = \{\mathbf{A}\}_{(2:n-1)\times(1:n-2)}$ is an invertible submatrix, $\mathbf{U}_r = \{\mathbf{U}\}_{1:n-2}$ contains the unknown portion of $\mathbf{U}$, and

$$\mathbf{b} = \begin{bmatrix} \dots \\ \mathbf{0} \\ \dots \\ -\beta_{n-2}\mathbf{u}_{n-1} \\ \mathbf{u}_{n-1} - \beta_{n-1}\mathbf{u}_n \end{bmatrix} \tag{3.5}$$

Thus

$$\mathbf{U}_r = -\mathbf{A}_r^{-1}\mathbf{b} \tag{3.6}$$

On the other hand, if a nontrivial $\mathbf{U}$ is given, we can compute the shape coefficients as follows:

$$\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} x_{i_\alpha} & x_{i_\beta} \\ y_{i_\alpha} & y_{i_\beta} \end{bmatrix}^{-1} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{3.7}$$

The following theorem states that $\mathbf{A}$ is *stable*, i.e., small changes in the contour imply small perturbations of the shape matrix and vice versa. Stability is extremely crucial when $\mathbf{A}$ must be estimated from sample contours which may be locally deformed.

Theorem 1 *The shape matrix of any nontrivial contour* $\mathbf{U}$ *is stable.*

Proof The proof follows directly from the definition of nontriviality. Since $\mathbf{U}$ is nontrivial, the inverse in (3.7) exists. Hence the shape coefficients satisfy linear equations where all elements are finite. It is then easy to show that small perturbations in x's and y's causes small perturbations in α's and β's. The converse is true by applying the same arguments to (3.6).

To adequately represent objects with shape irregularities, one must account for both global and local deformations, which are discussed next.

3.2.1 Global Deformations

Global deformations correspond to effects of rigid motion such as scaling, rotation, stretching and dilation, as shown in Figure 3.2.1(a). These operations can be represented by affine transformations on $\mathbf{V}$. The following theorem states that $\mathbf{A}$ is invariant and unique under such transformations:

Theorem 2 *Two nontrivial contours satisfy the same shape equation if and only if they are related by a linear transformation.*

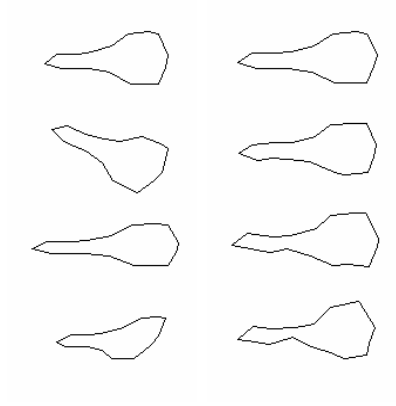
Proof Let $\mathbf{T}$ denote a 2×2 transformation matrix. First, note that if $\mathbf{A}\bar{\mathbf{U}}^T = \mathbf{0}$ and $\mathbf{U} = \mathbf{T}\bar{\mathbf{U}}$, then $\mathbf{A}\mathbf{U}^T = \mathbf{A}\bar{\mathbf{U}}^T\mathbf{T}^T = \mathbf{0}$.

To prove necessity, recall that $\text{rank}(\mathbf{A}) = n - 2$. Equivalently, the dimension of the null space of $\mathbf{A}$ is 2. Write $\bar{\mathbf{U}}^T = [\bar{\mathbf{x}}, \bar{\mathbf{y}}]$, where $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ are nx1 vectors. Because $\bar{\mathbf{U}}$ is nontrivial, $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are linearly independent and thus span the null space of $\mathbf{A}$.

Now, since $\mathbf{A}\mathbf{U}^T = \mathbf{A}[\mathbf{x}, \mathbf{y}] = \mathbf{0}$, both $\mathbf{x}$ and $\mathbf{y}$ are also in the null space of $\mathbf{A}$. Hence they can be expressed as a linear combination of $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, i.e.,

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} t_{11}\bar{\mathbf{x}} + t_{12}\bar{\mathbf{y}} \\ t_{21}\bar{\mathbf{x}} + t_{22}\bar{\mathbf{y}} \end{bmatrix} \Rightarrow \mathbf{U} = \mathbf{T}\bar{\mathbf{U}} \tag{3.8}$$

Finally, note that contour displacement, $\mathbf{g} = \bar{\mathbf{g}} + \mathbf{d}$, has no effect on the shape equation. This completes the proof.



(a) global (b) local

Figure 3.2: Examples of global and local deformations. (a) Global deformations described by rotation, stretching and dilation of the first contour; (b) local deformations of successively higher degree.

The invariance property implies that the shape coefficients of a contour are unaffected by affine transformations, while the uniqueness property makes recognition possible. For example, rectangles and parallograms have the same $\mathbf{A}_p$, and so are circles and ellipsoids with $\mathbf{A}_e$. However, $\mathbf{A}_p \neq \mathbf{A}_e$, i.e., they are unique, because families of parallograms and ellipsoids cannot be related by an affine transformation.

3.2.2 Local Deformations

Suppose we are interested in a family of contours $\mathbf{U} \in \Omega \subseteq \mathbb{E}^n$ which are similar but exhibit small shape irregularities, as shown in Figure 3.2.1(b). To represent these random fluctuations in small localities, we define an internal energy which is induced by the shape matrix $\mathbf{A}$:

$$E_{int}(\mathbf{U}) = \frac{(\mathbf{A}\mathbf{U}^T)^T \mathbf{R}^{-1} (\mathbf{A}\mathbf{U}^T)}{l(\mathbf{U})} \tag{3.9}$$

where $l(\mathbf{U}) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{u}_{i+1} - \mathbf{u}_i\|^2$ is a normalizing constant, and $\mathbf{R} = \text{diag}\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2\}$ contains the deformation variances σ_i^2 that allows assignment of location dependent weightings on deformations.

Now, we may assign probabilities to $\mathbf{U}$ as follows :

$$p(\mathbf{U}) = \frac{1}{Z} \exp(-E_{int}(\mathbf{U})) \tag{3.10}$$

where $Z = \sum_{\mathbf{U} \in \Omega} \exp(-E_{int}(\mathbf{U}))$ is a normalizing constant. A probability measure of the form in (3.10) is called a *Gibbs measure* [22]. By equivalence, it also defines a *Markov random field*, i.e.,

$$p(\mathbf{u}_i | \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n) = p(\mathbf{u}_i | \mathbf{u}_{i_\alpha}, \mathbf{u}_{i_\beta}) \tag{3.11}$$

where the conditional probability of $\mathbf{u}_i$ given the entire chain $\mathbf{U}$ is completely specified by the conditional probability of $\mathbf{u}_i$ given its two basis points.

The following theorem on E_{int} implies that $p(\mathbf{U})$ is scale and rotation invariant:

Theorem 3 *The internal energy of a contour is scale and rotation invariant.*

Proof Let

$$\mathbf{T}(z) = \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} \text{and } \mathbf{T}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

represents scale change and rotation respectively. Now

$$\mathbf{u}^T\mathbf{T}^T(z)\mathbf{T}(z)\mathbf{u} = z^2\mathbf{u}^T\mathbf{u} \tag{3.12}$$

$$\mathbf{u}^T\mathbf{T}^T(\theta)\mathbf{T}(\theta)\mathbf{u} = \mathbf{u}^T\mathbf{u} \tag{3.13}$$

Applying (3.12) and (3.13) to numerator and denominator of E_{int} yields the appropriate results. This completes the proof.

In summary, we have combined a stable, invariant and unique contour model with the local characteristic of Markov random field to yield prior distributions for any arbitrary contour. We proceed next to present an image model which allows us to cast contour extraction from noisy images into posterior estimation.

3.3 Contour Extraction

3.3.1 Rigid Template

We define image as the vector function $\mathbf{F} : \mathbb{E} \rightarrow \mathbb{D}$. Depending on the data type, either $\mathbb{D} = \mathbb{R}$ (intensity image or edge magnitude) or $\mathbb{D} = \mathbb{R}^2$ (2×1 intensity gradient vector).

A *template* of a contour is the image $\mathbf{F} = \mathrm{B}_{\mathrm{U,g}}$:

$$\mathrm{B}_{\mathrm{U,g}}(\mathbf{r}) = \sum_{i=1}^{n} \mathbf{h}_i \delta(\mathbf{r} - \mathbf{u}_i - \mathbf{g}) \tag{3.14}$$

where $\mathbf{r} = (x, y) \in \mathbb{E}$, δ is the delta function, $\mathbf{h}_i \in \mathbb{D}$ and $|\mathbf{h}_i| = 1$. In other words, a template $B_{U,g}(\mathbf{r})$ is a special image with values equal to $\mathbf{h}_i$ if $\mathbf{r} = \mathbf{u}_i + \mathbf{g}$ but zero otherwise. For $\mathbb{D} = \mathbb{R}^2$, $\mathbf{h}_i$ can be adaptively determined by rotating the tangent direction of the contour by 90°, i.e., let $\mathbf{t}_i$ represents the *tangent* vector at $\mathbf{u}_i$:

$$\mathbf{t}_i = \frac{\mathbf{u}_i - \mathbf{u}_{i-1}}{\|\mathbf{u}_i - \mathbf{u}_{i-1}\|} + \frac{\mathbf{u}_{i+1} - \mathbf{u}_i}{\|\mathbf{u}_{i+1} - \mathbf{u}_i\|} \tag{3.15}$$

then $\mathbf{h}_i$ is a unit vector which is normal to $\mathbf{t}_i$.

A *noisy* image $\mathbf{F} = \mathbf{f}$ containing a contour can be modeled as follows:

$$\mathbf{f}(\mathbf{r})|\mathbf{U}, \mathbf{g} = B_{U,g}(\mathbf{r}) + \eta(\mathbf{r}) \tag{3.16}$$

where $\eta(\mathbf{r}) : N(0, \sigma_\eta^2 \mathbf{I})$ is Gaussian, $\mathbf{I} = 1$ if $\mathbb{D} = \mathbb{R}$; otherwise $\mathbf{I}$ is a 2×2 identity matrix. Consequently, $p(\mathbf{f}|\mathbf{U}, \mathbf{g})$ is Gaussian distributed and can be simplified to yield:

$$p(\mathbf{f}|\mathbf{U}, \mathbf{g}) = \frac{1}{C} \exp\left\{ \frac{-\mathbf{f}^T\mathbf{f} - n + 2\sum_{i=1}^{n} \mathbf{h}_i^T \mathbf{f}(\mathbf{u}_i + \mathbf{g})}{2\sigma_\eta^2} \right\} \tag{3.17}$$

where $C = (2\pi\sigma_\eta^2)^{M^2/2}$.

The displacement of a rigid contour can be estimated from the noisy image using the principle of maximum likelihood:

$$\bar{\mathbf{g}} = \arg\max_{\mathbf{g}} \sum_{i=1}^{n} \mathbf{h}_i^T \mathbf{f}(\mathbf{u}_i + \mathbf{g}) \tag{3.18}$$

In other words, the estimated reference point is the location that best correlates with the image. This forms the basis of *matched template* operation.

In the same token, contours that may be deterministically deformed have the following ML estimates:

$$\{\bar{\mathbf{g}}, \bar{\mathbf{T}}\} = \arg\max_{\mathbf{g}} \max_{\mathbf{T}} \sum_{i=1}^{n} \mathbf{h}_i^T \mathbf{f}(\mathbf{T}\mathbf{u}_i + \mathbf{g}) \tag{3.19}$$

where the estimated contour $\bar{\mathbf{U}} = \bar{\mathbf{T}}\mathbf{U}$. The correlation can be efficiently implemented using generalized Hough transforms [4]. Note that the computation cost required in (3.19) depends greatly on the range and resolution of $\mathbf{T}$. For example, if one expects the contours to be scaled and rotated, then the range of $\mathbf{T}$ should span the two operations. In many instances the range can be restricted through learning or prior knowledge. Otherwise $\mathbf{T}$ will cover all the transformations, thus increasing the cost of searching.

3.3.2 Deformable Template

We assert that any rigid template $\bar{\mathbf{U}}$ will perform poorly if the contours are deformable:

Assertion 1 *The expected correlation of a matched template decreases with deformation.*

Proof Let $\mathbf{E}[\cdots]$ denotes mathematical expectation. Observe that

$$\begin{aligned}
\mathbf{E}[\mathrm{B}^T_{\bar{\mathrm{U}},\mathrm{g}}\mathbf{f}] &= \mathbf{E}[\mathbf{E}[\mathrm{B}^T_{\bar{\mathrm{U}},\mathrm{g}}\mathbf{f}|\mathbf{U},\mathbf{g}]] \\
&= \mathbf{E}[\mathrm{B}^T_{\bar{\mathrm{U}},\mathrm{g}}\mathrm{B}_{\mathrm{U},\mathrm{g}}] \\
&= \mathbf{E}[\sum_{i=1}^{n}\sum_{j=1}^{n}\delta(\bar{\mathbf{u}}_i - \mathbf{u}_j)]
\end{aligned}$$

For σ_i sufficiently small, $p(\bar{\mathbf{u}}_i = \mathbf{u}_j) \approx 0$ if $i \neq j$. Thus

$$\mathbf{E}[\mathrm{B}^T_{\bar{\mathrm{U}},\mathrm{g}}\mathbf{f}] \approx \mathbf{E}[\sum_{i=1}^{n}\delta(\bar{\mathbf{u}}_i - \mathbf{u}_i)] = p(\bar{\mathbf{U}})$$

Now, since $E_{int}(\bar{\mathbf{U}}) = 0$, from (3.10), we have

$$p(\bar{\mathbf{U}}) = 1/Z = \sum_{\mathrm{U}\in\Omega}\exp(E_{int}(\mathbf{U}))$$

which decreases if σ_i increases. This completes the proof.

Following the assertion, any rigid template chosen *a priori* will yield poor performance as the degree of deformation increases. Consequently, we need to generalize the idea of template matching to include *deformable* templates.

The problem at hand is equivalent to extracting a contour with *unknown* deformation from a noisy image. Using Bayesian framework, this turns into maximum *a posteriori* (MAP) estimation. The estimates $\mathbf{U}_{map}$ and $\mathbf{g}_{map}$ are given by:

$$\begin{aligned} \{\mathbf{U}_{map}, \mathbf{g}_{map}\} &= \arg\max_{\mathrm{U,g}} p(\mathbf{U}, \mathbf{g}|\mathbf{f}) \\ &= \arg\max_{\mathrm{U,g}} p(\mathbf{U}) p(\mathbf{f}|\mathbf{U}, \mathbf{g}) \end{aligned} \tag{3.20}$$

noting that $p(\mathbf{U}, \mathbf{g}) = p(\mathbf{U})$. While this prior distribution creates a bias to a global model, it is revised into posterior distribution after observing the image. In the next section, we will show how one may formulate and solve these estimation problems using the generalized active contour models.

3.4 Generalized Active Contour Models

3.4.1 Formulation

Denote the constituent *internal* and *external* energy as follows:

$$E_{int}(\mathbf{u}_i) = \frac{\|\mathbf{u}_i - \alpha_i \mathbf{u}_{i_\alpha} - \beta_i \mathbf{u}_{i_\beta}\|^2}{l(\mathbf{U})} \tag{3.21}$$

$$E_{ext}(\mathbf{u}_i, \mathbf{g}) = 1 - \mathbf{h}_i^T \mathbf{f}(\mathbf{u}_i + \mathbf{g}) \tag{3.22}$$

We will substitute these equations for $p(\mathbf{U})$ in (3.10) and $p(\mathbf{f}|\mathbf{U}, \mathbf{g})$ in (3.17). Solving for MAP estimation and ignoring constants, we have

$$\begin{aligned} \{\mathbf{U}_{map}, \mathbf{g}_{map}\} &= \arg\min_{\mathrm{U,g}} \sum_{i=1}^{n} \left\{ \frac{E_{int}(\mathbf{u}_i)}{\sigma_i^2} + \frac{E_{ext}(\mathbf{u}_i, \mathbf{g})}{\sigma_\eta^2} \right\} \\ &= \arg\min_{\mathrm{U,g}} \sum_{i=1}^{n} \left\{ \frac{\lambda_i}{1 - \lambda_i} E_{int}(\mathbf{u}_i) + E_{ext}(\mathbf{u}_i, \mathbf{g}) \right\} \end{aligned} \tag{3.23}$$

where

$$\lambda_i = \frac{\sigma_\eta^2}{\sigma_\eta^2 + \sigma_i^2} \in [0, 1] \tag{3.24}$$

are the local *regularization* parameters which control the amount of local template deformation.

The formulation in (3.23) is analogous to the *active contour models* (snakes) [21]. However, the original E_{int} only constrains the solution to the class of controlled continuity splines. Our formulation generalizes E_{int}, allowing for incorporation of prior models to create bias towards a particular type of contour. We call the resulting model *generalized active contour models* (g-snakes), or model-based snakes.

From (3.23), the task of finding $\mathbf{U}_{map}$ and $\mathbf{g}_{map}$ turns into an energy minimization problem, which is discussed next.

3.4.2 Minimization Algorithm

While $E_{int}(\mathbf{U})$ is a convex function, $E_{ext}(\mathbf{U})$ is typically non-convex as it is derived from intensity or gradient of noisy images. Consequently, unless the initial $\bar{\mathbf{U}}$ is placed very close to the global minimum, a gradient-based [21] or point-wise [46] algorithm will perform poorly as many local minima are present. Therefore an exhaustive search algorithm should be used.

Suppose that for each point $\bar{\mathbf{u}}_i$, we perform exhaustive search in its neighborhood of size m. The computation complexity is $O(m^n)$ which is prohibitively high for most applications. Fortunately, we can exploit the local characteristics of the Markov random field to yield an algorithm with complexity $O(nm^3)$.

The basic idea is to decompose the minimization process into n independent stages, where each stage considers only 3 neighboring points. This idea is first proposed in [2] under the framework of dynamic programming. We have generalized the idea to include any distribution with *local characteristics*.

To accomplish our goal, we define a state variable e_i, where

$$e_i(\mathbf{u}_{i_\alpha}, \mathbf{u}_i, \mathbf{u}_{i_\beta}) = \frac{E_{int}(\mathbf{u}_i|\mathbf{u}_{i_\alpha}, \mathbf{u}_{i_\beta})}{\sigma_i^2} + \frac{E_{ext}(\mathbf{u}_i, \mathbf{g})}{\sigma_\eta^2} \tag{3.25}$$

and let $\mathbf{U}^*$ be the solution to

$$\mathbf{U}^* = \arg\min_{\mathrm{U}} \sum_{i=1}^{n} e_i(\mathbf{u}_{i_\alpha}, \mathbf{u}_i, \mathbf{u}_{i_\beta}) \tag{3.26}$$

We begin by computing $s_1(\mathbf{u}_2, \mathbf{u}_3)$, where

$$s_1(\mathbf{u}_2, \mathbf{u}_3) = \min_{\mathbf{u}_1} \{e_1(\mathbf{u}_3, \mathbf{u}_1, \mathbf{u}_2) + e_2(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)\} \tag{3.27}$$

For the next stage, we use s_1 and e_3 to obtain

$$\begin{aligned} s_2(\mathbf{u}_3, \mathbf{u}_4) &= \min_{\mathbf{u}_2} \{s_1(\mathbf{u}_2, \mathbf{u}_3) + e_3(\mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)\} \\ &= \min_{\mathbf{u}_2, \mathbf{u}_1} \sum_{i=1}^{3} e_i(\mathbf{u}_{i_\alpha}, \mathbf{u}_i, \mathbf{u}_{i_\beta}) \end{aligned} \tag{3.28}$$

Proceeding in the like fashion, we obtain the partial solutions s_i for $i = 3$ through $i = n - 3$ by successively using e_{i+1} and s_{i-1}:

$$s_i(\mathbf{u}_{i+1}, \mathbf{u}_{i+2}) = \min_{\mathbf{u}_i} \{s_{i-1}(\mathbf{u}_i, \mathbf{u}_{i+1}) + e_{i+1}(\mathbf{u}_i, \mathbf{u}_{i+1}, \mathbf{u}_{i+2})\} \tag{3.29}$$

We then complete the computation in stage $n - 2$ to obtain

$$s_{n-2}(\mathbf{u}_{n-1}, \mathbf{u}_n) = \min_{\mathbf{u}_{n-2}} \{s_{n-3}(\mathbf{u}_{n-2}, \mathbf{u}_{n-1}) + e_{n-1}(\mathbf{u}_{n-2}, \mathbf{u}_{n-1}, \mathbf{u}_n) + e_n(\mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n-2})\} \tag{3.30}$$

Now, we can verify that

$$\min_{\mathbf{u}_{n-1}, \mathbf{u}_n} s_{n-2}(\mathbf{u}_{n-1}, \mathbf{u}_n) = \min_{\mathrm{U}} \sum_{i=1}^{n} e_i(\mathbf{u}_{i_\alpha}, \mathbf{u}_i, \mathbf{u}_{i_\beta}) \tag{3.31}$$

Consequently, we can use backward substitution to yield $\mathbf{U}^*$. Beginning from $i = n-1$ and $i = n$, we have

$$\{\mathbf{u}_{n-1}^*, \mathbf{u}_n^*\} = \arg \min_{\mathbf{u}_{n-1},\mathbf{u}_n} s_{n-2}(\mathbf{u}_{n-1}, \mathbf{u}_n) \tag{3.32}$$

We continue to backward substitute for $i = n-2$ through $i = 1$ to obtain

$$\mathbf{u}_i^* = \arg \min_{\mathbf{u}_i} s_{i-1}(\mathbf{u}_i, \mathbf{u}_{i+1}^*) \tag{3.33}$$

Note from (3.29) that we only require m^3 operations to compute s_i. This yields an algorithm that performs exhaustive search in the m-neighborhood with complexity $O(nm^3)$. We can generalize this idea to include any distribution with local characteristics: if the distribution can be decomposed into n-stages, each involving p neighboring points, then we can derive similar algorithm with complexity $O(nm^p)$.

To obtain $\mathbf{U}_{map}$, we iteratively apply the algorithm to yield $\mathbf{U}_1^*, \mathbf{U}_2^*, \ldots$, where each $\mathbf{U}_j^*$ is the solution to (3.26) by searching in the neighborhood of $\mathbf{U}_{j-1}^*$. The algorithm terminates when the energy cannot be further reduced, i.e., $\mathbf{U}_j^* = \mathbf{U}_{j-1}^*$. As the energy decreases monotonically, the algorithm is guaranteed to converge to a strong local minimum.

3.4.3 Basic and Stratified Line Search

In order for the minimization algorithm to converge to the global minima, the search regions must be sufficiently large to include at least part of the solution. In some cases this renders the algorithm infeasible as the complexity $O(nm^3)$ increases rather rapidly with m.

We present here search strategies which encompass large search regions without drastically increasing m. The basic idea is to concentrate the initial search in regions that will more likely yield the solution $\mathbf{U}^*$, instead of spreading them out evenly.

Refer to Figure 3.3. In the initial stages, we desire to rapidly inflate or deflate parts of $\mathbf{U}$ to locate the neighborhoods of the global minima. The objective can be achieved

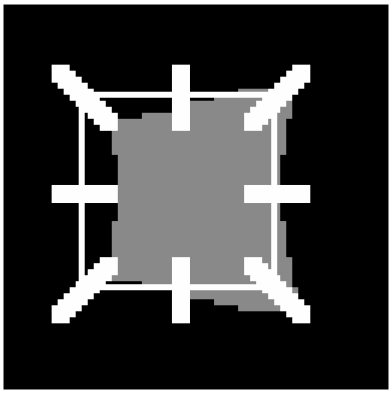


Figure 3.3: Regions of interest in line search. The gray blob is the image data, the white square is the initial contour, and the 8 normal vectors are the initial search directions.

by searching in the normal directions of $\mathbf{u}_i$. Note that this differs from [10] that we do not introduce additional inflation or deflation force; different parts of $\mathbf{U}$ can be inflated or deflated simultaneously.

The *basic* line search restricts its search in region $\Theta = \cup_{i=1}^{n} \Theta_i$ where Θ_i contains all the points on the normal vector $\mathbf{h}_i$:

$$\Theta_i = \left\{ \mathbf{u}_i = \bar{\mathbf{u}}_i + k\mathbf{h}_i ; k = 0, \pm 1, \ldots, \pm \frac{m-1}{2} \right\} \tag{3.34}$$

assuming that m is odd without loss of generality.

The *stratified* line search extends this idea to encompass even larger search regions by breaking Θ_i into disjoint segments of length l, $\Theta_i = \cup_{j=1}^{m/l} \Theta_{ij}$ where

$$\Theta_{ij} = \left\{ \mathbf{u}_i = \bar{\mathbf{u}}_i + (lj + k)\mathbf{h}_i ; k = 0, \pm 1, \ldots, \pm \frac{l-1}{2} \right\} \tag{3.35}$$

assuming that l is odd. The energies of each segment are then approximated as follows:

$$\hat{E}_{int}(\Theta_{ij}) = E_{int}(\bar{\mathbf{u}}_i + lj\mathbf{h}_i) \tag{3.36}$$

$$\hat{E}_{ext}(\Theta_{ij}) = \min_{\mathbf{u} \in \Theta ij} E_{ext}(\mathbf{u}, \mathbf{g}) \tag{3.37}$$

In other words, the internal energy of a segment is that of its center point, while its external energy is the minimum among its members. Assuming similar computation cost for internal and external energies, this yields drastically reduced complexity of $O(nm^3/l^2)$.

The stratified line search is typically performed in the initial stages of minimization to quickly locate regions which contain the global minima. This can then be followed by basic line search and completed by searching in 3×3 regions.

3.4.4 Regularization

The regularization parameters λ_i in (3.23) are derived from the variances σ_η and σ_i. In some applications these parameters may be learned from training samples. In cases where learning is impossible or unreliable, the minimax principle [14] can be used.

The *local minimax criterion* [23] determines the optimal regularization by minimizing the worst case energy. We subject σ_η and σ_i to the constraint

$$\frac{1}{\sigma_i^2} + \frac{1}{\sigma_\eta^2} = 1 \tag{3.38}$$

and seek a solution for

$$\begin{aligned} \{\mathbf{U}_{lmc}, \mathbf{g}_{lmc}\} &= \arg\min_{\mathrm{U,g}} \sum_{i=1}^{n} \max_{\sigma_\eta^2, \sigma_i^2} \left\{ \frac{E_{int}(\mathbf{u}_i)}{\sigma_i^2} + \frac{E_{ext}(\mathbf{u}_i, \mathbf{g})}{\sigma_\eta^2} \right\} \\ &= \arg\min_{\mathrm{U,g}} \sum_{i=1}^{n} \max \left\{ E_{int}(\mathbf{u}_i), E_{ext}(\mathbf{u}_i, \mathbf{g}) \right\} \end{aligned} \tag{3.39}$$

The local minimax criterion is *pareto* optimum, i.e., there does not exist another solution $\mathbf{U}^*$ that will simultaneously lower both the internal and external energy. Moreover, the strategy solves the problem without requiring prior knowledge or explicit computation of the regularization parameters, and thus can be easily incorporated into the proposed minimization algorithm.

In the absence of reliable prior information, this criterion minimizes the penalty of erroneous regularization selection. Extensive experimentation in Chapter 2 has verified that it yields robust performance in various applications.

3.4.5 Initialization

The objective of initialization is to place the g-snake in the proximity of the minima in (3.23) or (3.39) so as to facilitate speedy convergence in minimization. In other words, we want to pick initial $\{\bar{\mathbf{U}}, \bar{\mathbf{g}}\}$ that will, on average, converge to $\mathbf{U}_{map}$ or $\mathbf{U}_{lmc}$ more rapidly then all other choices.

Such criterion is difficult to specify in precise mathematical terms. However, we can make a reasonable assumption that a g-snake of lower energy is located closer to the minima compared to one of higher energy. Then the objective becomes that of finding a initial contour that will, on average, begin with lower energy than all other choices. Let

$$e(\mathbf{U}, \Lambda) = \sum_{i=1}^{n} \frac{\lambda_i}{1-\lambda_i} E_{int}(\mathbf{u}_i) + E_{ext}(\mathbf{u}_i) \tag{3.40}$$

represents the total energy. We assert that

Assertion 2 *A contour* $\bar{\mathbf{U}}$ *that satisfies* $E_{int}(\bar{\mathbf{U}}) = 0$ *will, on average, begin with the lowest total energy. Moreover, this contour can be initialized through generalized Hough transform.*

Proof Since $E_{int}(\bar{\mathbf{U}}) = 0$ and $E_{int}(\mathbf{U}) \geq 0$, it immediately follows that $\mathbf{E}[E_{int}(\bar{\mathbf{U}})] \leq \mathbf{E}[E_{int}(\mathbf{U})]$. Also,

$$\begin{aligned} \mathbf{E}[E_{ext}(\mathbf{U})] &= 1 - \mathbf{E}[\mathrm{B}^T_{\mathrm{U,g}}\mathbf{f}] \\ &= 1 - p(\mathbf{U}) \\ &= 1 - \frac{1}{Z}\exp(-E_{int}(\mathbf{U})) \end{aligned}$$

$E_{int}(\bar{\mathbf{U}}) = 0$ implies that $p(\bar{\mathbf{U}}) \geq p(\mathbf{U})$ and hence $E[E_{ext}(\bar{\mathbf{U}})] \leq E[E_{ext}(\mathbf{U})]$. Combining the results for internal and external energy, we conclude that

$$\mathbf{E}[e(\bar{\mathbf{U}}, \Lambda)] \leq \mathbf{E}[e(\mathbf{U}, \Lambda)]$$

In other words, $\bar{\mathbf{U}}$ has the lowest total energy on average. One should therefore initialize the g-snake using a reference contour $\bar{\mathbf{U}}$ generated from the shape matrix, i.e., one can initialize the g-snake using generalized Hough transform.

The next section demonstrates by experiments how one may estimate the shape matrix from training samples, and subsequently apply the trained g-snakes in various applications. We also compare the matching performance of g-snakes and rigid templates.

3.5 Applications and Experiments

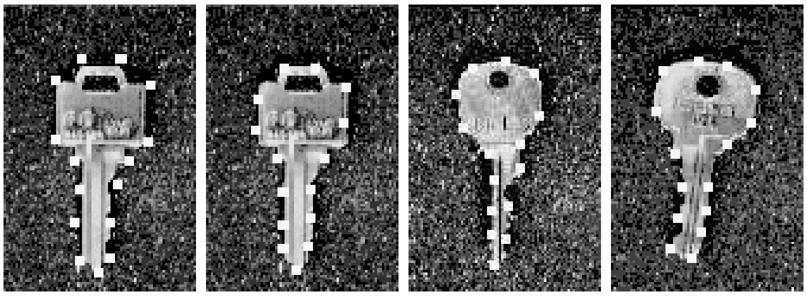
3.5.1 Training

Figure 3.4 shows the sequence of steps taken to train the shape matrix for key localization. In these examples the 2×1 intensity gradient vectors are computed by fitting planes in 2×2 windows using the method of least squares.

Figure 3.4(a) shows the feature points that form the chain $\mathbf{U}$. The set generally include locations of high curvature which can be selected either manually or automatically through dominant point detection [39]. An initial estimate of the shape matrix are computed from $\mathbf{U}$. Using this shape matrix and minimax regularization, the total energy of the g-snake is minimized to yield $\hat{\mathbf{U}}_1$ as in Figure 3.4(b). The shape matrix is then updated to $\hat{\mathbf{A}}_1$.

We repeat the procedure for many key samples to obtain $\hat{\mathbf{A}}_1$, $\hat{\mathbf{A}}_2, \ldots,$ $\hat{\mathbf{A}}_m$. Using the principle of maximum likelihood, the estimated shape matrix is given as follows:

$$\hat{\mathbf{A}} = \frac{1}{m} \sum_{j=1}^{m} \hat{\mathbf{A}}_j \tag{3.41}$$



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

Figure 3.4: G-snake training. (a) Selected feature points; (b)(c)(d) $\hat{\mathbf{U}}_j$ used to generate $\hat{\mathbf{A}}_j$ for $j = 1, 2$ and 3. These $\hat{\mathbf{A}}_j$ are averaged for $\hat{\mathbf{A}}$ estimation.

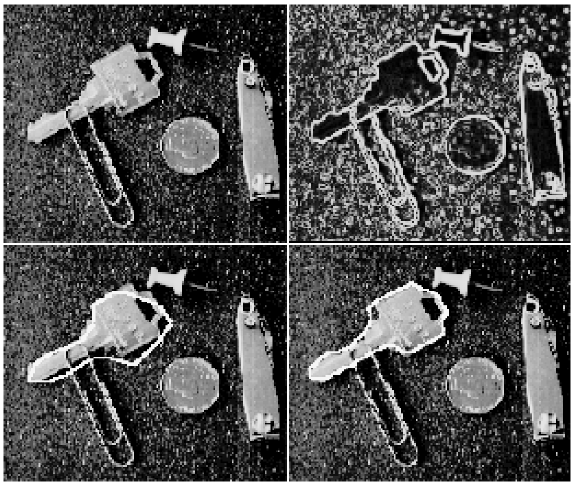
Similar method is employed to estimate the two reference points used to generate $\bar{\mathbf{U}}$ from $\hat{\mathbf{A}}$.

We may also estimate the local regularization parameters λ_i by learning the deformation and noise variance from the training samples. As insufficient training samples were available for this experiment, minimax regularization was used.

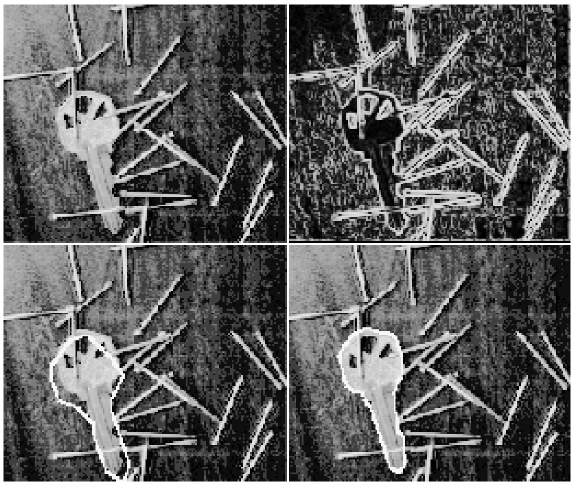
3.5.2 Initialization and Extraction

We used the generalized Hough transform to initialize the g-snakes in this experiment, considering only rotation for $\mathbf{T}$ in equation (3.19). From the initialized contours, we minimize the energy on the g-snake and show the results in Figure 3.5. Beginning from the top left corner, each example shows the original image, edge magnitude map, final and initial boundaries in the clockwise order.

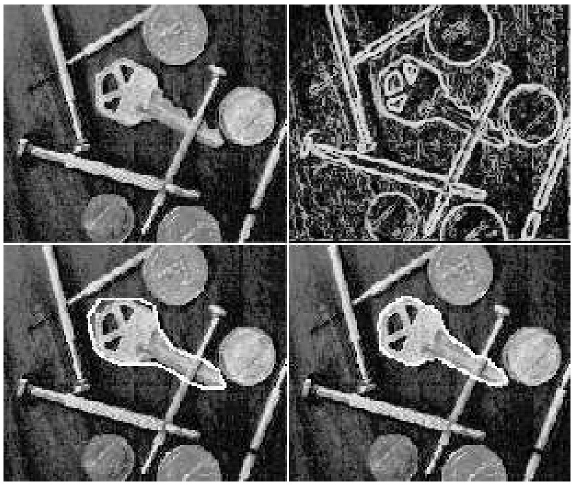
As evidenced in the edge maps, these scenes contain considerable clutter and occlusion. As a consequence, traditional methods based on edge linking are likely to produce poor results. It can be seen that the generalized Hough transform successfully approximate the locations of the keys, but cannot produce precise boundaries. The g-snakes refined these initial contours to yield accurate reproduction of the boundaries.



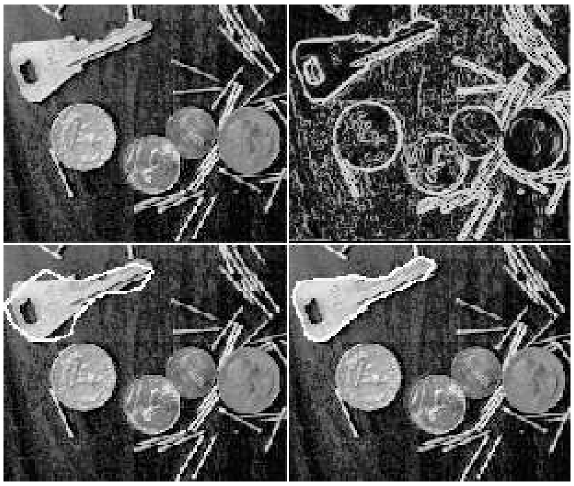
(a)



(b)



(c)



(d)

Figure 3.5: Extraction of keys from noisy, occluded and cluttered images. In clockwise direction, each example shows original image, edge magnitude, final and initial boundary.

3.5.3 Affine Invariant Contour Model

Figure 3.6 shows the invariance of contour model to affine transformations. Similar training steps are used to estimate the shape matrix for rectangles. The results show that generalized Hough transform successfully estimated the necessary displacement, scale change, rotation and dilation to yield good initialization, see Figure 3.6(a), (b) and (c) (top row). The g-snake accounts for local deformation to produce contours that match the underlying images, see Figure 3.6(a), (b) and (c) (bottom).

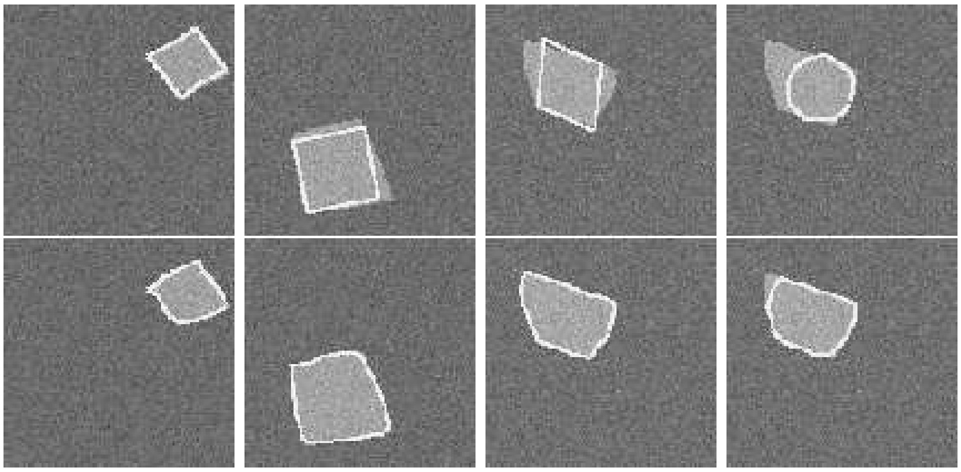
3.5.4 Comparison to Snake

Figuer 3.6(d) shows the results of applying the *snake* of [21]. As its internal energy restricts the solution to controlled continuity spline, the initialized state is a circle. Failing to utilize the appropriate contour model (i.e., rectangle), it yields suboptimal solution with smoothed corners. The g-snake, in contrast, utilize the contour model to yield good solution.

3.5.5 Handwritten Characters

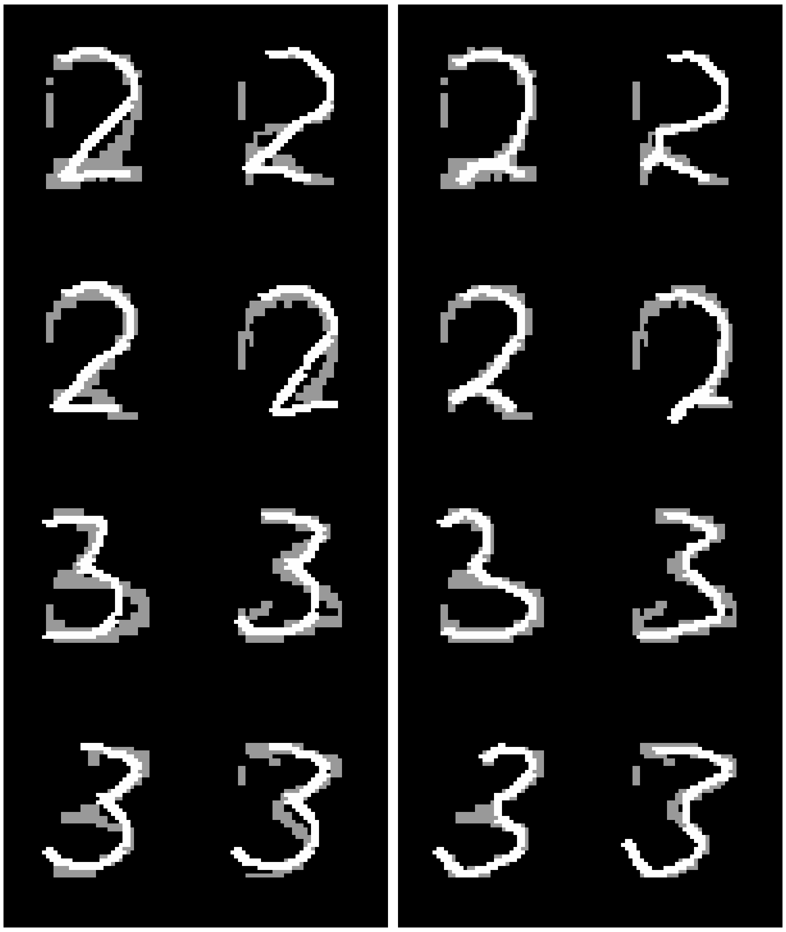
Figure 3.7 shows applications of g-snake in extracting the skeleton of handwritten characters, using intensity image as the data type. After undergoing necessary deformations, the g-snake produces skeletons that represent the underlying characters.

We performed the simulation on a SUN-IPX workstation, using 60×50 images and 18 snake points. Combined with stratified line search with $m = 15$ and $l = 3$, the energy minimization algorithm consumed 1.13 seconds on average. On the other hand, the generalize Hough transform required 0.5 second on average when only 5 quantized states of rotation were considered. While the average time consumed by the generalized Hough transform increased drastically to 21.5 seconds with 243 quantized states of rigid transformations (include scale change, rotation and stretching), averaged run time of energy minimization remained stable. Similarly, time required by the generalized Hough



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

Figure 3.6: (a)(b)(c): Invariance of contour model to affine transformations. The g-snakes refined the initial guess to account for local deformations. (d) Results of applying original *snake*. Failing to incorporate the appropriate contour model (i.e., rectangle), it yields suboptimal solution.



(a) initial (b) final

Figure 3.7: Extraction of handwritten characters.

transform also depended greatly on image size, but run time of energy minimization depended only on n, the number of snake points, and m, the search region size.

The simulation demonstrates the fact that regardless of the range of affine transformations and image size, g-snake is capable to refine the initial guess in a constant amount of time. Consequently, g-snake is particularly effective and powerful in applications such as handwritten character extraction. As an alternative, the range of transformations can be significantly reduced by normalization, resulting in rapid and reliable initialization.

3.5.6 Severe Clutter and Deformation

Figure 3.8 shows an example of the effect of severe cluttering and deformations on g-snakes. Under such conditions, it has been shown [15, 16] that the generalized Hough transform has high probability of generating false peaks, especially when one considers all possible transformations. This results in a poorly initialized g-snake which also yields poor final results.

For successful extraction in these scenes, one should either remove the clutters through pre-processing or restrict the range of allowable transformations. Otherwise, alternative strategies for initialization must be employed.

3.5.7 Matching Performance

This section compares the matching performance of g-snake to that of rigid template under the effect of deformation and image noise. Matching performance is defined as the degree of correlation between a template and an image:

$$c(\mathbf{U}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{h}_i^T \mathbf{f}(\mathbf{u}_i + \mathbf{g}) \tag{3.42}$$

where $c(\mathbf{U})$ takes value between 0 for no correlation and 1 for perfect correlation. Using rigid template (generalized Hough transform) and g-snake with minimax regularization,

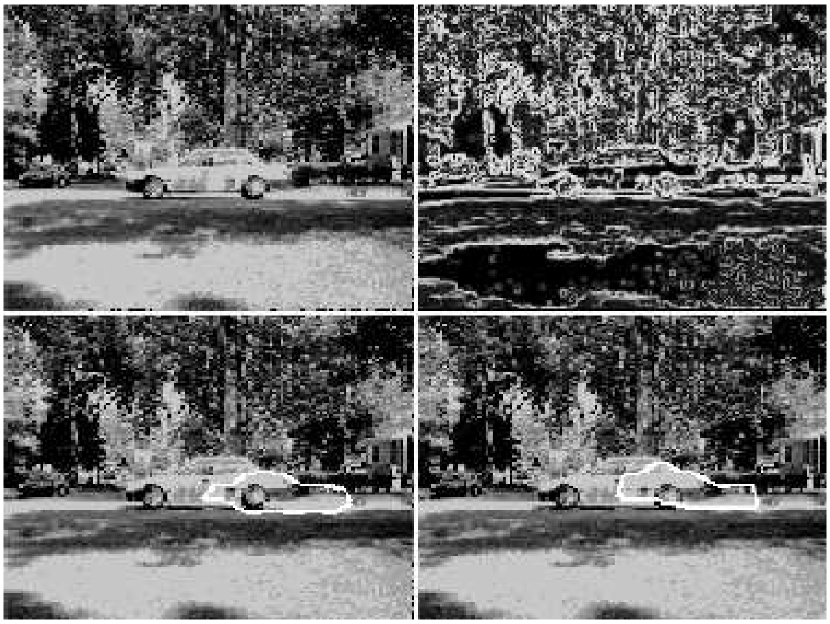


Figure 3.8: Effects of severe clutter and deformation. Poor initialization yields poor solution.

we obtain the respective average correlation $\mathbf{E}[c(\mathbf{U})]$ and plot the results against deformation.

The experiment uses synthesized images containing deformed rectangles as shown in Figure 3.9. We deform the boundary on each of the 4 sides independently using a randomly generated cosine function. Let $b(s)$ represents an undeformed boundary indexed by $s \in [0, 1]$. The deformed boundary $d(s)$ is given by:

$$d(s) = b(s) + \nu \cos(\pi s + \phi) \tag{3.43}$$

where $\nu : N(0, \sigma_\nu^2)$ is a normally distributed boundary deformation process and ϕ is a random phase shift uniformly distributed in $[0, 2\pi)$. Large values of σ_ν induce large values of deformation variance σ_i, although the exact relationship has not been determined. A plot of $\mathbf{E}[c(\mathbf{U})]$ versus σ_ν will therefore illustrate the effect of deformation on matching performance.

The intensity images are generated by setting the pixel value to 1 if it is enclosed by the boundaries, and 0 otherwise. Image noise is then introduced using zero-mean Gaussian white noise of variance σ_n.

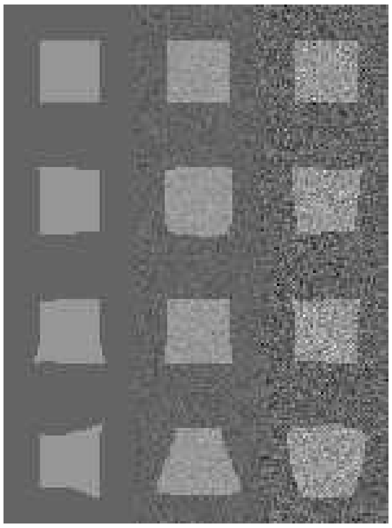
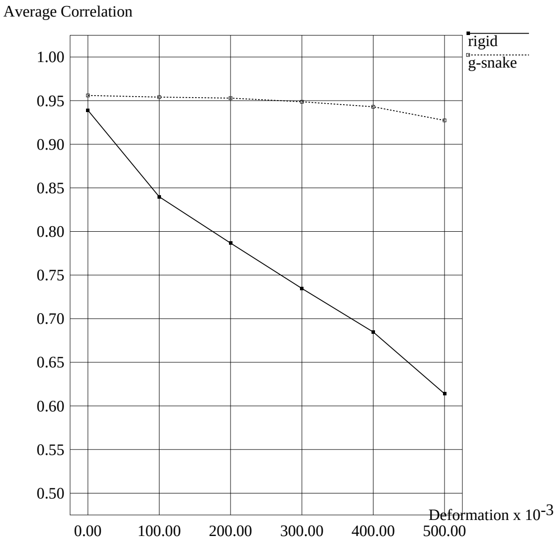


Figure 3.9: Synthetic images used to compare matching performance. Deformation variance σ_ν increases from top to bottom while image noise variance σ_n increases from left to right.

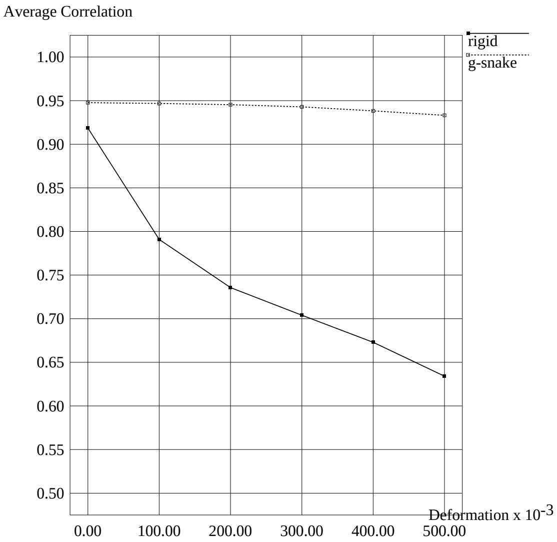
Figure 3.10(a) to (d) show the plots of $\mathbf{E}[c(\mathbf{U})]$ versus deformation with σ_n fixed at 0.0, 0.1, 0.2 and 0.3. These experiments confirm the assertion that matching performance of rigid template degrades with deformation. In contrast, the g-snake adapts well with deformation to yield high degree of correlation. Furthermore, by comparing the plots obtained with various σ_n, we observe that g-snake exhibits higher robustness to image noise.

3.6 Conclusions

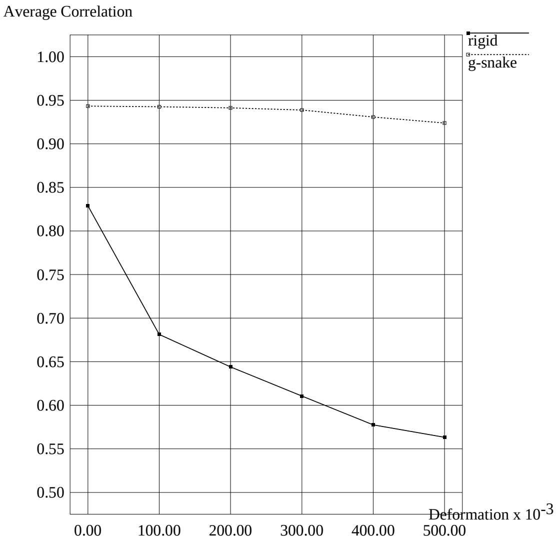
We considered the problems of modeling and extracting arbitrary deformable contours from noisy images. Based on a *regenerative* shape matrix, our model encompasses both global and local deformations. In addition, it satisfies the following conditions:



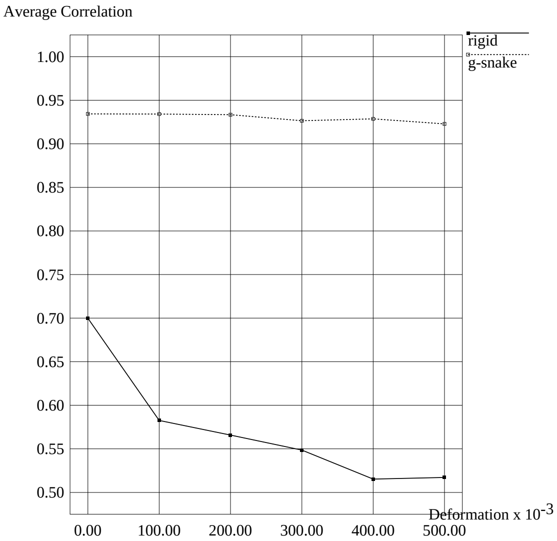
(a) $\sigma_n = 0.0$



(b) $\sigma_n = 0.10$



(c) $\sigma_n = 0.20$



(d) $\sigma_n = 0.30$

Figure 3.10: Plots of average correlation $\mathbf{E}[c(\mathbf{U})]$ versus standard deviation of boundary deformation process, σ_ν. Standard deviation of image noise σ_n was fixed in each of the plots.

- *Stability.* Small changes in contour implies small perturbations in the shape matrix and vice versa. Thus one can reliably estimate the shape matrix from sample contours even though these contours may be locally deformed.

- *Invariance.* The shape matrix is implicitly invariant to affine transformations. This is crucial in extracting contours undergoing rigid transformation unknown in advance.

- *Uniqueness.* Two contours that cannot be related by an affine transformation have different shape matrix representation. This makes recognition possible.

We exploited the local characteristics of the Markov random field to model highly variable but locally predictable contours. This yields invariant prior distribution that exert influence over a contour model, while maintaining the ability to deform locally.

Subsequently, we present an image model and obtain the probability distribution given that a contour is present. Using the Bayesian framework, the problem of extracting a contour with unknown deformation from noisy images turns into MAP estimation. This is equivalent to energy minimization in the *generalized active contour models* (g-snakes). We addressed several primary issues in g-snakes:

- *Energy Formulation.* The overall objective function consists of two terms: an internal energy function incorporating a global contour model, and an external energy function adjusting the resulting contour based on the image data. Thus the g-snakes enable incorporation of specific prior knowledge of an arbitrary shape of interest while allowing for deformations. Furthermore, the shape matrix can be estimated from training samples.

- *Energy Minimization.* We present a multistage minimization algorithm which is guaranteed to converge to the global minimum. The complexity of the algorithm is $O(nm^3)$, where n is the number of points on g-snake, and m is the size of the search region. Combined with efficient line search strategies, the algorithm will rapidly converge to the final solutions from the initialized states.

- *Regularization.* When sufficient training samples are available, the deformation and noise variance can be estimated and translated into local regularization parameters. Otherwise, the minimax principle provides a way to adaptively determine the optimal regularization from the underlying image data, thus no input parameter is needed.

- *Initialization.* In most applications, we may reliably and efficiently initialize the g-snakes using generalized Hough transform. Nevertheless, the Hough transform is prone to generating false peaks under severe cluttering and deformation. Under these conditions, one should either remove the clutters through pre-processing or restrict the range of allowable transformations. Otherwise, alternative strategies for initialization must be employed.

Finally, we demonstrated with experiments how one may apply the proposed g-snake in various applications. The g-snakes are particularly powerful and effective when appropriate prior models can be incorporated and the range of allowable global transformation is known. Moreover, quantitative measures obtained from the experiments confirm that g-snakes yield superior matching performance compared to rigid templates.

Chapter 4

Deformable Contours: Detection and Classification

4.1 Introduction

An important problem in computer vision and pattern recognition is to detect the presence of a particular contour, or to classify among several possible contours from noisy images. A popular approach is to assume that the images contain rigid contours in additive Gaussian white noise. Under these assumptions, it can be shown [40] that *matched template* is optimum in terms of probability of error. Moreover, since it essentially involves correlating a particular template with the image, the matched template can be efficiently implemented using the classical or generalized Hough transform [4, 12].

Figure 4.1: Handwritten characters with deformations.

In practice, however, the assumption of rigidity is often violated. The contours of interest often exhibit *deformation* which arises from diversity and irregularity of shape. A good example is handwritten character recognition, where considerable deformation is produced by variation in writing style, as evidenced in Figure 4.1. Because matched template fails to model this effect, we have shown in Chapter 3 that its averaged correlation degrades rapidly with deformation. Moreover, it has high probability of generating false peaks for complex scenes [15, 16].

Although there exist numerous methods [18, 45] that detect and classify deformable contours, these typically assume that the contour has been previously extracted. However, without a contour model, contour extraction has been shown to be an ill-posed problem [37]. Consequently, the practicality of these methods is limited for applications involving noisy images.

Existing methods dealing with detecting and classifying deformable contours directly from noisy image are mostly based on intuitive ideas or heuristics. In handwritten character recognition, [19] uses a weighted combination of energy functions which measure deviation from an ideal model and closeness of match. However, there is no formal mechanism to determine the weighting factor which significantly affects the performance. In shape matching, [20] uses the Hausdorff distance which is tolerant of small position errors. However, the method is likely to break down when outliers or considerable deformations are present. Moreover, since no image noise model was considered, the algorithm must heuristically determine the fraction of edge points to be compared with the model. In curve recognition, [42] uses the implicit polynomials and formally invokes the Bayes decision rules. Nonetheless, to achieve adequate representation power, polynomials of rather high degree are typically required. In the presence of noise, these coefficients quickly become unstable and their joint probabilities are difficult to compute.

This chapter presents a formal strategy to detect and classify deformable contours directly from noisy images. Our formulation is based on the *generalized active contour models* (g-snake) [25], which incorporates both a deformation model and an image noise model. Unlike Fourier descriptors [41] or implicit polynomials [42], the shape representation in g-snake is *stable* i.e., small deviations in shape will yield small perturbations in shape representation and vice versa. Moreover, this shape representation is unique, implicitly invariant to affine transformations, and possesses metric properties. These characteristics make g-snakes suitable for detection and classification tasks.

Using the Nearmen-Pearson lemma, we derive the optimal detection and classification tests. Although these tests require the marginalization of the distribution, the summation is peaked in most practical applications and thus only small regions need to be considered. Finally, we perform extensive experimentations using synthesized images and report significant improvement over matched template in handwritten numeral recognition.

We begin in next section by giving a summary of the contour and image models presented in Chapter 3.

4.2 Contour and Image Models

4.2.1 Representing Shape

A contour is a vector containing an ordered set of points, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$. Each $\mathbf{v}_i$ is defined on the finite grid: $\mathbf{v} \in \mathbb{E} = \{(x, y) : x, y = 1, 2, \ldots, M\}$, thus $\mathbf{V} \in \mathbb{E}^n$.

Denote $\mathbf{U} \in \mathbb{E}^n$ where each $\mathbf{u}_i = \mathbf{v}_i - \mathbf{g}$ represents the displacement from an arbitrary reference point $\mathbf{g}$. We can encode the shape of an arbitrary contour in the following *shape equation*:

$$\mathbf{A}\mathbf{U}^T = \mathbf{0} \tag{4.1}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & -\beta_1 & -\alpha_1 & 0 & \ldots & 0 \\ -\alpha_2 & 1 & -\beta_2 & 0 & 0 & \ldots \\ 0 & -\alpha_3 & 1 & -\beta_3 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & -\alpha_{n-1} & 1 & -\beta_{n-1} \\ 0 & 0 & \ldots & -\beta_n & -\alpha_n & 1 \end{bmatrix}$$

is an $n \times n$ regenerative and stable *shape matrix*.

It can be shown that the shape equation is invariant and unique to affine transformations on $\mathbf{V}$. In other words, for a given $\mathbf{A}$, if $\bar{\mathbf{U}}$ satisfies $\mathbf{A}\bar{\mathbf{U}}^T = \mathbf{0}$, then

$$\mathbf{A}\mathbf{U}^T = \mathbf{0} \Leftrightarrow \mathbf{U} = \mathbf{T}\bar{\mathbf{U}} \tag{4.2}$$

where $\mathbf{T}$ is a 2x2 invertible transformation matrix which may describe *rigid motions* such as scaling, rotation, stretching and dilation. Moreover, displacement on the center of gravity, $\bar{\mathbf{g}}+\mathbf{d}$, has no effect on the shape equation. In our discussions we shall assume that $\mathbf{T}$ and $\mathbf{d}$ are either known, or can be estimated reliably using the generalized Hough transforms. In other words, we shall assume knowledge of the *reference contour* $\bar{\mathbf{U}}$ and $\bar{\mathbf{g}}$.

4.2.2 Rigid and Deformable Contours

We say that a family of contours $\mathbf{U} \in \Omega \subseteq \mathbb{E}^n$ is *rigid* if

$$p(\mathbf{U}) = \begin{cases} 1; & \mathbf{U} = \bar{\mathbf{U}} \\ 0; & \text{otherwise} \end{cases} \tag{4.3}$$

In other words, $\Omega = \{\bar{\mathbf{U}}\}$.

Conversely, we say that a family of contours $\mathbf{U} \in \Omega$ is *deformable* if

$$p(\mathbf{U}) > 0, \quad \text{for some } \mathbf{U} \neq \bar{\mathbf{U}} \tag{4.4}$$

The next section presents the deformation model which is used to obtain the distribution $p(\mathbf{U})$ in (4.4).

4.2.3 Modeling Deformation

Suppose the family of contours $\mathbf{U} \in \Omega$ are similar but exhibit small shape irregularities. To represent these random fluctuations in small localities, we define an internal energy which is induced by the shape matrix $\mathbf{A}$:

$$E_{int}(\mathbf{U}) = \frac{(\mathbf{A}\mathbf{U}^T)^T \mathbf{R}^{-1} (\mathbf{A}\mathbf{U}^T)}{l(\mathbf{U})} \tag{4.5}$$

where $l(\mathbf{U}) = \sum_{i=1}^{n} \|\mathbf{u}_{i+1} - \mathbf{u}_i\|^2$ is a normalizing constant, and $\mathbf{R} = \mathrm{diag}\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2\}$ contains the deformation variances σ_i^2 that allows assignment of location weightings on deformations. We can decompose $E_{int}(\mathbf{U})$ into its n constituents:

$$E_{int}(\mathbf{U}) = \sum_{i=1}^{n} \frac{E_{int}(\mathbf{u}_i)}{\sigma_i^2} \tag{4.6}$$

where $E_{int}(\mathbf{u}_i) = \frac{1}{l(\mathbf{U})}\|\mathbf{u}_i - \alpha_i \mathbf{u}_{i_\alpha} - \beta_i \mathbf{u}_{i_\beta}\|^2$ and the basis indices are given by:

$$i_\alpha = \begin{cases} i-1; & i > 1 \\ 3; & i = 1 \end{cases} \qquad i_\beta = \begin{cases} i+1; & i < n \\ n-2; & i = n \end{cases} \tag{4.7}$$

We assign probability to $\mathbf{U}$ using the induced energy function:

$$p(\mathbf{U}) = \frac{1}{Z} \exp(-E_{int}(\mathbf{U})) \tag{4.8}$$

where $Z = \sum_{\mathbf{U} \in \Omega} \exp(-E_{int}(\mathbf{U}))$ is a normalizing constant. A probability measure of the form (4.8) is called a Gibbs measure. By equivalence, it also defines a Markov random field [22], i.e.,

$$p(\mathbf{u}_i | \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n) = p(\mathbf{u}_i | \mathbf{u}_{i_\alpha}, \mathbf{u}_{i_\beta}) \tag{4.9}$$

where the conditional probability of $\mathbf{u}_i$ given the entire chain $\mathbf{U}$ is completely specified by the conditional probability of $\mathbf{u}_i$ given its two basis points.

In summary, we have combined a global contour model with Markov random field to yield prior distribution for any arbitrary contour. Next, we examine various characteristics of this formulation to determine its suitability for the tasks of detection and classification.

4.2.4 Characteristics of Contour Model

The contour model presented above possess several important and desirable characteristics for the tasks of detection and classification.

Firstly, the shape matrix is *invariant* under affine transformation, which is crucial in detecting or classifying objects undergoing rigid motions in the 3-D space. Furthermore, affine invariance of the shape matrix is *implicit*, unlike Fourier descriptors [41] or memorized templates [19, 20, 5] which require exhaustive comparison into a created subspace.

Secondly, the shape matrix is *unique*. Hence two contours that cannot be related by an affine transformation cannot have similar shape matrix. Recognition is impossible without this characteristics.

Finally, the internal energy E_{int} possesses the *metric properties* which corresponds to the intuitive notion of shape resemblence. Let $\bar{\mathbf{U}}$ denotes the reference contour where $\mathbf{A}\bar{\mathbf{U}}^T = \mathbf{0}$, then

- $E_{int}(\mathbf{U}) \geq 0$, $E_{int}(\mathbf{U}) = 0$ if and only if $\mathbf{U} = \mathbf{T}\bar{\mathbf{U}}$ for some $\mathbf{T}$. Thus a contour is only similar to itself or its affine transformations.

- $|E_{int}(\mathbf{U}_p) - E_{int}(\mathbf{U}_q)| = |E_{int}(\mathbf{U}_q) - E_{int}(\mathbf{U}_p)|$. The order of comparison of two contours does not matter.

- $|E_{int}(\mathbf{U}_p) - E_{int}(\mathbf{U}_q)| \leq |E_{int}(\mathbf{U}_p) - E_{int}(\mathbf{U}_r)| + |E_{int}(\mathbf{U}_r) - E_{int}(\mathbf{U}_q)|$. The triangular inequality ensures that two contours that are highly dissimilar cannot both be similar to some third contour.

A contour, either rigid or deformable, typically corresponds to strong edge points or high intensity lines in an image. However, these edge or intensity points may also arise due to noise. In the next section, we describe an image model consisting of a contour template in additive Gaussian white noise.

4.2.5 Image Model

We define an *image* as the vector function $\mathbf{F} : \mathbb{E} \rightarrow \mathbb{D}$. Depending on the data type, either $\mathbb{D} = \mathbb{R}$ (intensity image or edge magnitude) or $\mathbb{D} = \mathbb{R}^2$ (2×1 intensity gradient vector).

A *template* of a contour is the image $\mathbf{F} = \mathrm{B}_{\mathrm{U,g}}$:

$$\mathrm{B}_{\mathrm{U,g}}(\mathbf{r}) = \sum_{i=1}^{n} \mathbf{h}_i \delta(\mathbf{r} - \mathbf{u}_i - \mathbf{g}) \tag{4.10}$$

where $\mathbf{r} = (x, y) \in \mathbb{E}$, δ is the delta function, $\mathbf{h}_i \in \mathbb{D}$ and $|\mathbf{h}_i| = 1$. In other words, a template $\mathrm{B}_{\mathrm{U,g}}(\mathbf{r})$ is a special image with values equal to $\mathbf{h}_i$ if $\mathbf{r} = \mathbf{u}_i + \mathbf{g}$ but zero otherwise. For $\mathbb{D} = \mathbb{R}^2$, $\mathbf{h}_i$ is the unit vector which is normal to the contour.

A *noisy* image $\mathbf{F} = \mathbf{f}$ containing a contour can be modeled as follows:

$$\mathbf{f}(\mathbf{r})|\mathbf{U}, \mathbf{g} = \mathrm{B}_{\mathrm{U,g}}(\mathbf{r}) + \eta(\mathbf{r}) \tag{4.11}$$

where $\eta(\mathbf{r}) : N(0, \sigma_\eta^2 \mathbf{I})$ is Gaussian, $\mathbf{I} = 1$ if $\mathbb{D} = \mathbb{R}$; otherwise $\mathbf{I}$ is a 2×2 identity matrix. Consequently, $p(\mathbf{f}|\mathbf{U}, \mathbf{g})$ is Gaussian distributed and can be simplified to yield:

$$p(\mathbf{f}|\mathbf{U}) = \frac{1}{C} \exp \left\{ \frac{-\mathbf{f}^T\mathbf{f} + n - 2\sum_{i=1}^{n} E_{ext}(\mathbf{u}_i)}{2\sigma_\eta^2} \right\} \tag{4.12}$$

where $C = (2\pi\sigma_\eta^2)^{M^2/2}$, and the external energy $E_{ext}(\mathbf{u}_i) = 1 - \mathbf{h}_i^T \mathbf{f}(\mathbf{u}_i)$.

Next, we will use the contour and image models described above to derive the optimum detection and classification tests. We will then show how these tests can be implemented in practical applications.

4.3 Detection and Classification

Suppose there are two candidate models ω_0 and ω_1, which generate images $\mathbf{F} = \mathbf{f}$ that are distributed according to the density function $p_0(\mathbf{f})$ and $p_1(\mathbf{f})$ respectively. We identify these models with hypothesis H_0 and H_1 as follows:

$$\begin{aligned} H_0 : \ \mathbf{f} : \ p_0(\mathbf{f}) \\ H_1 : \ \mathbf{f} : \ p_1(\mathbf{f}) \end{aligned} \tag{4.13}$$

We shall assume that the only difference between ω_1 and ω_0 is the contour distribution, and that $\mathbf{f}^T\mathbf{f}$ is constant. Consequently we can compute the probabilities $p_i(\mathbf{f})$ by marginalizing the distribution i.e.,

$$p_i(\mathbf{f}) = \sum_{\mathrm{U}\in\Omega_i} p_i(\mathbf{U})p(\mathbf{f}|\mathbf{U}) \tag{4.14}$$

If ω_i is rigid, i.e., $\Omega_i = \{\bar{\mathbf{U}}_i\}$, then (4.14) reduces to a simple form:

$$p_i(\mathbf{f}) = p(\mathbf{f}|\bar{\mathbf{U}}_i) \tag{4.15}$$

4.3.1 Detection

Given an observed image $\mathbf{f}$, the problem of *detection* can be cast into that of testing the null hypothesis H_0 against H_1. We say that ω_0 is a *null model* if $p_0(\mathbf{f})$ is *uniform* for all images $\mathbf{f}$. In other words, the null model does not generate contours of any coherent structure.

Naturally, we want to minimize both the probability of false alarm, P_{fa}, and the probability of miss, P_{miss}. However, in most cases we cannot simultaneously minimize P_{fa} and P_{miss}. Thus we seek a test which will minimize P_{miss}, or maximize the probablity of detection, $P_D = 1 - P_{miss}$, for a fixed false alarm probability P_{fa}. Such a test is called the *most powerful* test.

Define the likelihood ratio $L_{ij}(\mathbf{f}) = p_i(\mathbf{f})/p_j(\mathbf{f})$. The Neyman-Pearson lemma states that a test taking the form

$$\text{select} \begin{cases} H_1: & \text{if } L_{10}(\mathbf{f}) > k \\ H_0: & \text{otherwise} \end{cases} \tag{4.16}$$

for some $k \geq 0$, is the most powerful test for a given $P_{fa} > 0$.

Since $p_0(\mathbf{f})$ is uniform for null models, the detection test reduces to

$$\text{select} \begin{cases} H_1: & \text{if } p_1(\mathbf{f}) > k' \\ H_0: & \text{otherwise} \end{cases} \tag{4.17}$$

for some $k' > 0$.

A simple detection test arises if ω_1 is a rigid model. Using (4.15) and (4.12), we select H_1 if

$$\sum_{i=1}^{n} \mathbf{h}_i^T \mathbf{f}(\bar{u}_i) > k'' \tag{4.18}$$

for some $k'' > 0$. In other words, we detect contour $\bar{\mathbf{U}}$ if it yields sufficient correlation with the observed image. This is equivalent to a matched template operation which can be efficiently implemented using the generalized Hough transform.

If ω_1 is a deformable model, it is necessary to marginalize the distribution. Thus we select H_1 if

$$\sum_{\mathbf{U} \in \Omega_1} p_1(\mathbf{U}) p(\mathbf{f}|\mathbf{U}) > k' \tag{4.19}$$

for some $k' > 0$.

4.3.2 Classification

In classification, one normally does not distinguish between P_{fa} and P_{miss}, but rather seeks to minimize the total probability of error, $P_e = P_{fa} + P_{miss}$. Using the NP lemma, a test of the form in (4.16) with $k = 1$ satisfies the requirement. Alternatively, Bayes rules can be employed to yield

$$k = \frac{p_0 C_{01}}{p_1 C_{10}} \tag{4.20}$$

where p_i are the prior probabilities of observing model ω_i, and C_{ij} is the *cost* associated with deciding H_j given that H_i is in force. Without loss of generality, we shall use $k = 1$ in subsequent discussions.

If both models ω_1 and ω_0 are rigid, then we select H_1 if it yields higher correlation with the image:

$$\sum_{i=1}^{n} \mathbf{h}_i^T \mathbf{f}(\bar{u}_{1,i}) > \sum_{i=1}^{n} \mathbf{h}_i^T \mathbf{f}(\bar{u}_{0,i}) \tag{4.21}$$

In general, however, a classification test may involve one or more deformable models and thus require marginalization. In these cases, we select H_1 if

$$\sum_{\mathrm{U} \in \Omega_1} p_1(\mathbf{U}) p(\mathbf{f}|\mathbf{U}) > \sum_{\mathrm{U} \in \Omega_0} p_0(\mathbf{U}) p(\mathbf{f}|\mathbf{U}) \tag{4.22}$$

The classification test can be easily generalized to the case of multiple classes $\omega_1, \omega_2, \ldots, \omega_q$. In this case, we select ω_γ if

$$\gamma = \arg \max_{1 \leq i \leq q} \sum_{\mathrm{U} \in \Omega_i} p_i(\mathbf{U}) p(\mathbf{f}|\mathbf{U}) \tag{4.23}$$

It can be seen from (4.23) that depending on the size of Ω_i, the tests involving deformable models may involve rather extensive computations. In the worst case, we must enumerate all possible M^{2n} instances of $\mathbf{U} \in \mathbb{E}^n$ on the finite grid. Clearly this computation load is infeasible even for relatively small images.

The next section addresses some practical issues in implementing these detection and classification tests. Firstly, we exploit the local characteristics of the Markov random field to design a summation algorithm with complexity $O(nm^3)$, where n is the number of points and m is the region size. As the summation is peaked around the posterior estimate in most applications, we can significantly reduce the computation time by applying the algorithm in small rectangular regions. Finally, we show how one can use the Monte-Carlo method and importance sampling to obtain the normalizing constant Z.

4.4 Issues in Practical Implementation

4.4.1 Summation Algorithm

Suppose that in Ω there are m_i possible locations for each $\mathbf{u}_i$. To exhaustively compute

$$\sum_{\mathbf{U}\in\Omega} p(\mathbf{U})p(\mathbf{f}|\mathbf{U}) = \sum_{\mathbf{u}_1}\sum_{\mathbf{u}_2}\cdots\sum_{\mathbf{u}_n} p(\mathbf{U})p(\mathbf{f}|\mathbf{U}) \tag{4.24}$$

we require $\prod_{i=1}^{n} m_i$ enumerations. This computational load is factorial and easily leads to combinatoric explosion.

Fortunately, we can exploit the local characteristics of the Markov random field to yield an algorithm that computes in polynomial time. To accomplish this, we define a state variable e_i, where

$$e_i(\mathbf{u}_{i_\alpha}, \mathbf{u}_i, \mathbf{u}_{i_\beta}) = \exp\left\{-\frac{E_{int}(\mathbf{u}_i|\mathbf{u}_{i_\alpha}, \mathbf{u}_{i_\beta})}{\sigma_i^2} - \frac{E_{ext}(\mathbf{u}_i)}{\sigma_x^2}\right\} \tag{4.25}$$

and observe that

$$p(\mathbf{U})p(\mathbf{f}|\mathbf{U}) = \frac{\exp(-\mathbf{f}^T\mathbf{f} + n)}{ZC}\prod_{i=1}^{n} e_i(\mathbf{u}_{i_\alpha}, \mathbf{u}_i, \mathbf{u}_{i_\beta}) \tag{4.26}$$

This allows us to decompose the problem into an n-stage process and only consider three snake points at each stage.

We begin by computing the partial sum $\mathcal{S}_1(\mathbf{u}_2, \mathbf{u}_3)$:

$$\mathcal{S}_1(\mathbf{u}_2, \mathbf{u}_3) = \sum_{\mathbf{u}_1} e_1(\mathbf{u}_3, \mathbf{u}_1, \mathbf{u}_2)e_2(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) \tag{4.27}$$

We then obtain the next partial sum $\mathcal{S}_2(\mathbf{u}_3, \mathbf{u}_4)$ using e_3 and $\mathcal{S}_1$:

$$\begin{aligned}\mathcal{S}_2(\mathbf{u}_3, \mathbf{u}_4) &= \sum_{\mathbf{u}_2} \mathcal{S}_1(\mathbf{u}_2, \mathbf{u}_3) e_3(\mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) \\ &= \sum_{\mathbf{u}_2} \sum_{\mathbf{u}_1} e_1(\mathbf{u}_3, \mathbf{u}_1, \mathbf{u}_2) e_2(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) e_3(\mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) \qquad (4.28)\end{aligned}$$

Proceeding in the like fashion, we accumulate the partial sums $\mathcal{S}_i$ for $i = 3$ through $n-3$ by successively using e_{i+1} and $\mathcal{S}_{i-1}$:

$$\mathcal{S}_i(\mathbf{u}_{i+1}, \mathbf{u}_{i+2}) = \sum_{\mathbf{u}_i} \mathcal{S}_{i-1}(\mathbf{u}_i, \mathbf{u}_{i+1}) e_i(\mathbf{u}_i, \mathbf{u}_{i+1}, \mathbf{u}_{i+2}) \qquad (4.29)$$

Finally, at stage $n-2$, we compute

$$\mathcal{S}_{n-2}(\mathbf{u}_{n-1}, \mathbf{u}_n) = \sum_{\mathbf{u}_{n-2}} \mathcal{S}_{n-3}(\mathbf{u}_{n-2}, \mathbf{u}_{n-1}) e_{n-1}(\mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n-2}) e_{n-2}(\mathbf{u}_{n-2}, \mathbf{u}_{n-1}, \mathbf{u}_n) \qquad (4.30)$$

Now, we can verify that

$$\begin{aligned}\mathcal{S}_\Omega &= \sum_{\mathrm{U}\in\Omega} \prod_{i=1}^{n} e_i(\mathbf{u}_{i_\alpha}, \mathbf{u}_i, \mathbf{u}_{i_\beta}) \\ &= \sum_{\mathbf{u}_n} \sum_{\mathbf{u}_{n-1}} \mathcal{S}_{n-2}(\mathbf{u}_{n-1}, \mathbf{u}_n) \qquad (4.31)\end{aligned}$$

Hence, from (4.26), we have

$$\sum_{\mathrm{U}\in\Omega} p(\mathbf{U}) p(\mathbf{f}|\mathbf{U}) = \frac{\exp(-\mathbf{f}^T\mathbf{f} + n)}{ZC} \mathcal{S}_\Omega \qquad (4.32)$$

and this completes the summation.

Note from (4.29) that we only require $m_i \cdot m_{i+1} \cdot m_{i+2}$ operations to compute the partial sums $\mathcal{S}_i$. Suppose $m_i = m$ for all i, then we have successfully reduced the order of computation to $O(nm^3)$.

4.4.2 Peaked Summation

In most practical applications, the summation in (4.14) is peaked around the posterior estimate $\mathbf{U}_{map}$, where

$$\mathbf{U}_{map} = \max_{\mathbf{U}\in\Omega} p(\mathbf{U})p(\mathbf{f}|\mathbf{U}) \tag{4.33}$$

is the solution to energy minimization in g-snake. In other words, most of the samples that contributes to the summation is concentrated in a small region around $\mathbf{U}_{map}$, as shown in Figure 4.2. Consequently, we can significantly reduce the computation time by applying the summation algorithm in small rectangular regions (e.g. 3×3 windows) centered around $\mathbf{U}_{map}$. Hence the solution of g-snake can be rapidly post-processed to yield the solution for detection or classification.

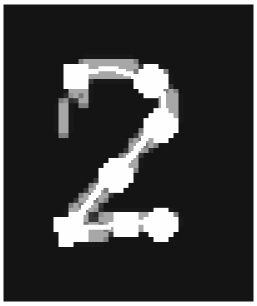


Figure 4.2: Regions that contribute significantly to summation. For clarity, some points (and regions) along the contour have been omitted.

4.4.3 Normalizing Constant

From (4.14) and (4.32), the likelihood function $L_{ij}(\mathbf{f})$ can be written as follows:

$$L_{ij}(\mathbf{f}) = \frac{p_i(\mathbf{f})}{p_j(\mathbf{f})} = \frac{\mathcal{S}_{\Omega_i} Z_j}{\mathcal{S}_{\Omega_j} Z_i} \tag{4.34}$$

assuming that image noise is uniform over all the classes. Thus the normalizing constant Z_i must be calculated *once* during the training stage. Now, since

$$Z_i = \sum_{\mathrm{U} \in \Omega_i} \exp(-E_{int}(\mathbf{U})) \tag{4.35}$$

exhaustive computation of Z_i is a time-consuming process if Ω_i is large. Nevertheless, we can approximate the constants using Monte-Carlo integration method and importance sampling.

The Monte-Carlo integration method treats the integrand as a random variable and uses an interative sampling and averaging scheme to yield a parameter estimate of the mean of the random variable. To apply the method to obtain Z, we take a set of q independent samples $\mathbf{U}_1, \mathbf{U}_2, \ldots \mathbf{U}_q$ drawn uniformly from Ω. The q^{th} estimate of Z is

$$Z^q = \mathrm{size}(\Omega) \cdot \frac{1}{q} \sum_{i=1}^{q} \exp(-E_{int}(\mathbf{U}_i)) \tag{4.36}$$

where size(Ω) is the total number of samples in Ω. By the strong law of large numbers, $|Z^q - Z| \to 0$ as $q \to \infty$. Moreover, the variance of the estimate $\mathbf{E}[(Z^q - Z)^2]$ is reduced at a rate of $1/q$.

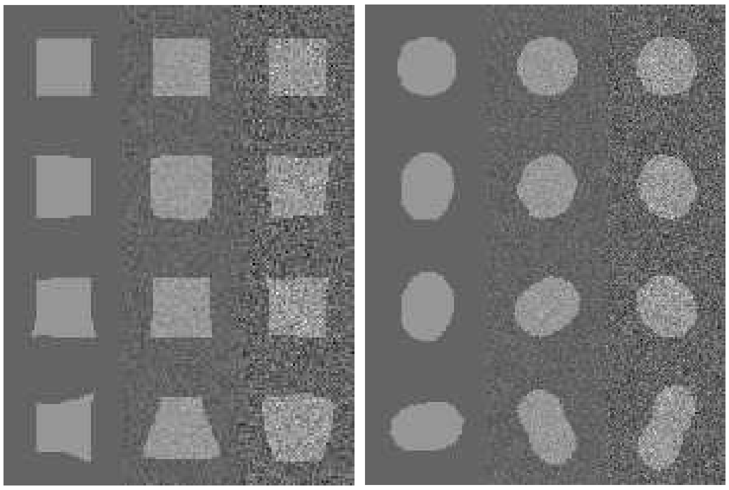
For deformation variances σ_i sufficiently small, the summation is peaked around the reference contour $\bar{\mathbf{U}}$. Thus we may invoke the concept of *importance sampling* to concentrate the samples in the peaked regions, which yields significantly reduced computation time.

The next section presents experimental results which confirm the validity of our formulation.

4.5 Experiments

4.5.1 Quantitative Comparisons

This section compares the detection and classification performance of the proposed deformable template to rigid (matched) template under the effect of deformation and image noise.



(a) deformed rectangles (b) deformed ellipsoids

Figure 4.3: Synthesized images for quantitative comparison

The experiment uses synthesized images containing deformed rectangles and ellipsoids as shown in Figure 4.3. For rectangles, we deform the boundary on each of the 4 sides independently using a randomly generated cosine function. Let $b(s)$ represents an undeformed boundary indexed by $s \in [0, 1]$. The deformed boundary $d(s)$ is given by:

$$d(s) = b(s) + \nu \cos(\pi s + \phi) \tag{4.37}$$

where $\nu : N(0, \sigma_\nu^2)$ is a normally distributed boundary deformation process and ϕ is a random phase shift uniformly distributed in $[0, 2\pi)$. For ellipsoids, we simply use

$d(s)$ as the boundary. In other words, we superimpose a smoothly varying cosine wave on the boundary of the ellipsoids. Note that large values of σ_ν induce large values of deformation variance σ_i, although the exact relationship has not been determined.

The intensity images in Figure 4.3 are generated by setting the pixel value to 1 if it is enclosed by the boundaries, and 0 otherwise. Image noise is then introduced using i.i.d. Gaussian noise with variance σ_n.

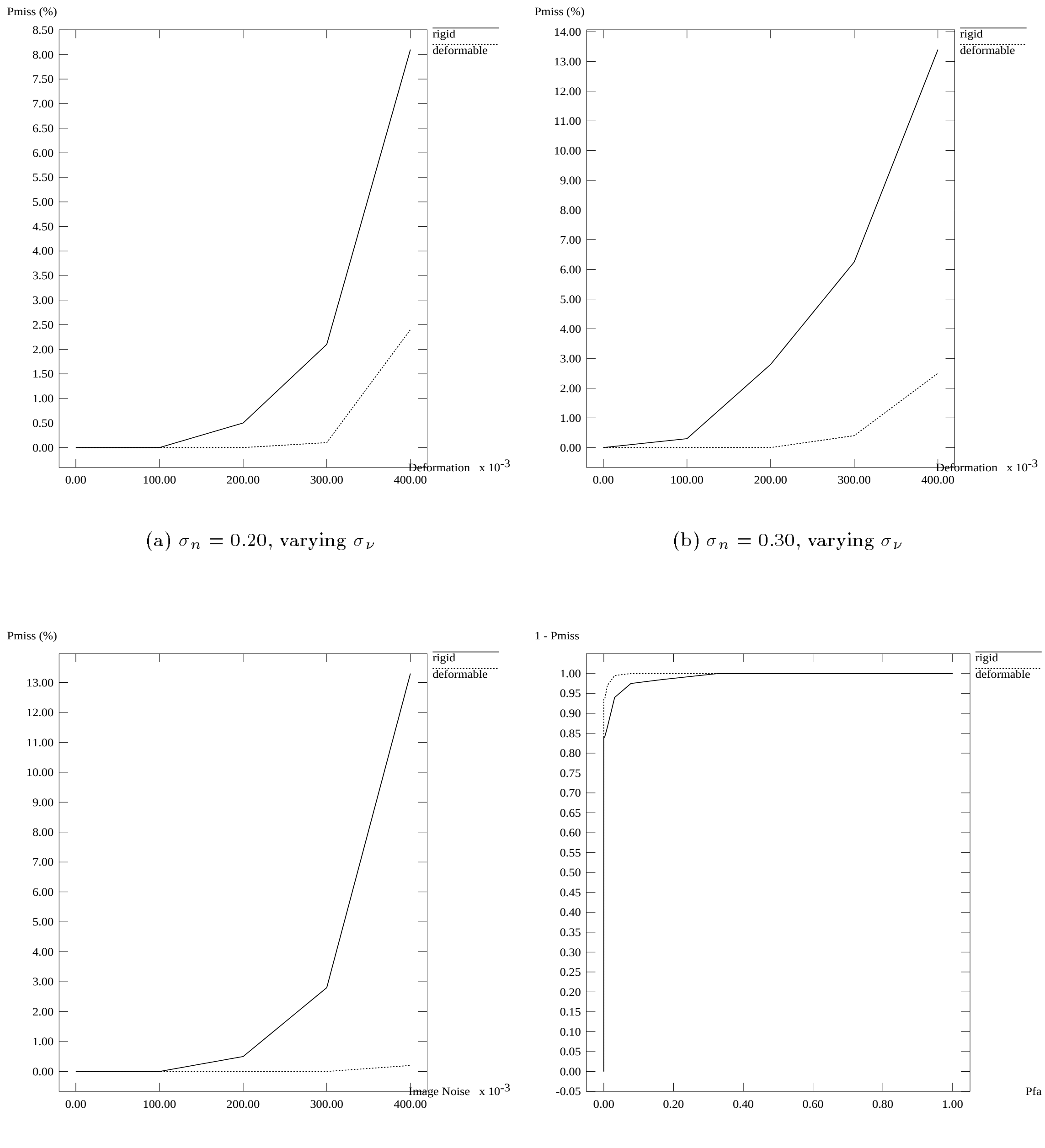
The rigid templates (generalized Hough transform) consist of 16 equally spaced points obtained from perfect rectangles and circles ($\sigma_\nu = \sigma_n = 0$). For each pair of σ_ν and σ_n, we generate training samples and use them to estimate the regularization parameters for deformable templates.

To marginalize the distribution in (4.19) and (4.22), we first obtain the posterior estimate using energy minimization and search strategies presented in [25]. As the summation is peaked, we can approximate the results by performing summation in 3×3 regions centered around the posterior estimate. Using the summation algorithm presented in the previous section, the computation complexity is $O(16 \times 9^3)$.

Detection

For detection tests, two sets of image are used for each specified pair of parameters: one contains deformed rectangles, while the other contains only noise. We then perform detection using equation (4.18) for rigid template and (4.19) for deformable template and compared their performance. Figure 4.4 shows the detection results under various conditions of deformation and image noise.

Figure 4.4(a) and (b) illustrates the effect of deformation on the templates. By enforcing the condition that $P_{fa} = 0.01$ we obtained the probability of miss, P_{miss}, versus the standard deviation of boundary deformation process, σ_ν, with standard deviation of image noise fixed at $\sigma_n = 0.20$ and 0.30. These plots confirm the assertion that at a fixed level of image noise, rigid templates generate large detection error when considerable deformation is present. The deformable templates, in contrast, provide better



(a) $\sigma_n = 0.20$, varying σ_ν

(b) $\sigma_n = 0.30$, varying σ_ν

(c) $\sigma_\nu = 0.20$, varying σ_n

(d) Operating Curve, $\sigma_\nu = 0.30$, $\sigma_n = 0.40$

Figure 4.4: Detection results. (a)(b) Probability of miss, P_{miss} (with $P_{fa} = 0.01$) versus standard deviation of boundary deformation process, σ_ν, with standard deviation of image noise fixed at $\sigma_n = 0.20$ and 0.30; (c) P_{miss} ($P_{fa} = 0.01$) versus σ_n with $\sigma_\nu = 0.20$; (d) Operating Curve, 1 - P_{miss} versus P_{fa}, with $\sigma_\nu = 0.30$ and $\sigma_n = 0.40$.

adaptation to deformation and thus yield smaller detection error.

Figure 4.4(c) examines the effect of image noise on the templates by varying σ_n and fixing $\sigma_\nu = 0.20$. One can observe that the performance of the rigid template deteriorates rapidly when large image noise is present. As the probability of false alarm increases dramatically for higher image noise, fixing $P_{fa} = 0.01$ at a low level causes P_{miss} to increase dramatically. In contrast, the marginalization employed in deformable templates tends to "smooth out" the noise and thus produces smaller detection error.

Figure 4.4(d) shows the operating curve (1 - P_{miss} versus P_{fa}) with $\sigma_\nu = 0.30$ and $\sigma_n = 0.40$. The results demonstrate that the deformable templates produce superior operating characteristics for detection when compared to rigid templates.

Classification

For classification tests, we generated noisy images containing deformed rectangles and ellipsoids for each specified pair of parameters. We then classified the images using equation (4.21) for rigid template and (4.22) for deformable template and present the results in Figure 4.5.

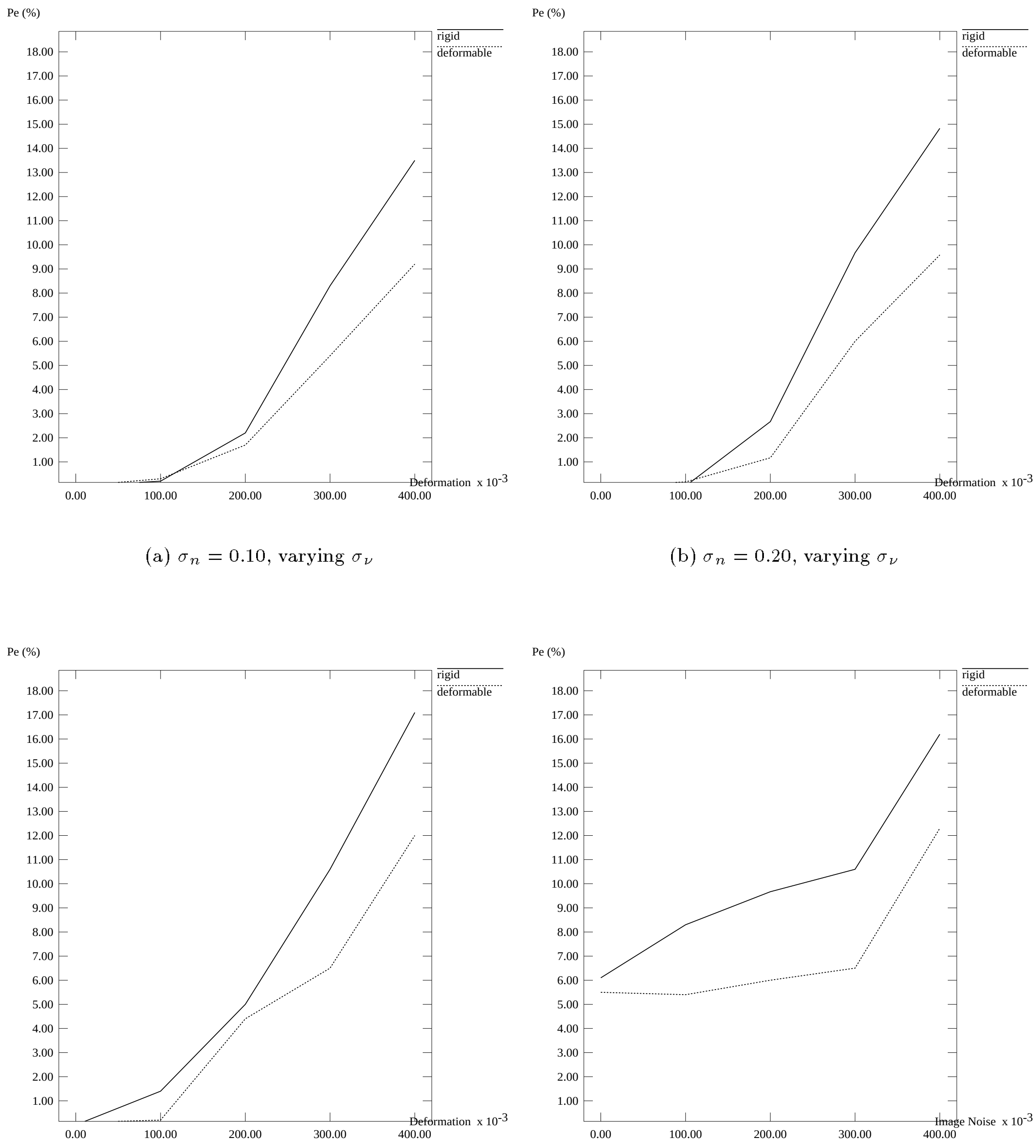
Figure 4.5(a), (b) and (c) demonstrate the effect of deformation on classification performance. As in detection, when considerable deformation is present, deformable templates provide the necessary adaptation to yield smaller classification error.

Figure 4.5(d) examines the effect of image noise on the classification performance. One can observe that the deformable templates produce smaller classification error for all levels of image noise.

In summary, these extensive and rigorous experimentations have verified the superiority of deformable templates over rigid templates in the presence of deformation.

4.5.2 Numeral Recognition

This section compares the classification performance of deformable template versus rigid template, using printed and handwritten numerals (2 & 3) as shown in Figure 4.6.



(a) $\sigma_n = 0.10$, varying σ_ν

(b) $\sigma_n = 0.20$, varying σ_ν

(c) $\sigma_n = 0.30$, varying σ_ν

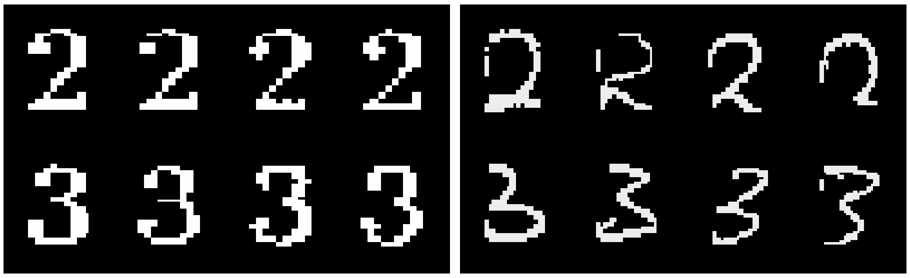
(d) $\sigma_\nu = 0.30$, varying σ_n

Figure 4.5: Classification results. (a)(b)(c) Probability of error, P_e versus standard deviation of boundary deformation process, σ_ν, with standard deviation of image noise fixed at $\sigma_n = 0.10, 0.20$ and 0.30; (d) P_e versus σ_n with $\sigma_\nu = 0.30$.

	rigid	deformable
printed	0.00	0.00
handwritten	0.22	0.07

Table 4.1: Probability of error, P_e, for printed and handwritten numerals (2 & 3) classification using rigid and deformable templates.

Table 4.1 shows the classification results using 190 character samples each.



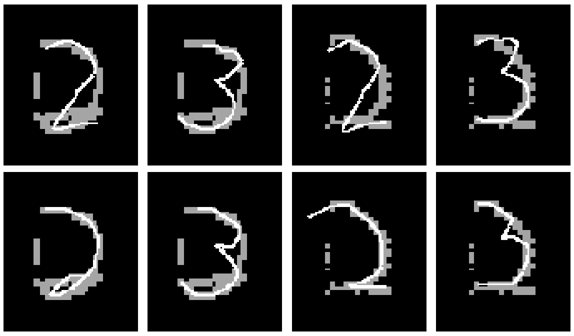
(a) printed numerals (b) handwritten numerals

Figure 4.6: Printed and handwritten numerals

As there are minimal deformation in printed numerals, rigid template is optimum and thus yields very good classification results. In this case it is unnecessary to employ deformable templates because very little improvement can be achieved.

Handwritten numerals, however, exhibit considerable deformation due to variation in writing style. Consequently, the rigid templates performed poorly and misclassified 22% of the samples presented. In contrast, deformable templates were successful in reducing the misclassification rate to 7%.

Figure 4.7 demonstrates the effectiveness of deformable templates in using both the goodness of match and the degree of deformation to achieve correct classification. The top row of Figure 4.7 shows the best match provided by rigid templates, while the bottom row shows the posterior estimates of deformable templates. In both cases the rigid template corresponds to numeral "3" yielded higher correlation when compared to that of "2", and thus misclassified the character as "3". In contrast, the deformable templates in Figure 4.7(a) underwent the necessary deformation to yield higher correlation for "2", and thus correctly classified the character. In Figure 4.7(b), although the deformable templates yielded higher correlation for "3", the template corresponds to "3" underwent more severe deformation in order to achieve the match. Consequently, the deformable templates correctly classified the character as "2" using information derived from prior distributions.



(a) (b)

Figure 4.7: Illustrations of rigid and deformable templates. The top row shows the best match provided by rigid templates, while the bottom row shows the posterior estimates of deformable templates. In both cases the rigid templates misclassified the character as "3". (a) The deformable templates yielded higher correlation for "2" and correctly classified the character; (b) The deformable template yielded higher correlation for "3", but correctly classified the character as "2" using information derived from prior distributions.

We performed the simulation on a SUN-IPX workstation, using 60×50 images and 18 snake points. Considering 25 quantized states of rigid transformation in the generalized Hough transform, the rigid template consumed 1.2 seconds per input image on average. The deformable templates required an additional 5.4 seconds on average to perform energy minimization and marginalization for each input image. The run time can be significantly reduced if appropriate hardware and parallelization are employed.

4.6 Conclusions

We have presented a formal strategy to detect and classify deformable contours directly from noisy images. The proposed *deformable template* is based on the generalized active contour model (g-snake). The g-snake is suitable for these tasks because its shape representation is stable, unique, affine invariant and possesses metric properties which correspond to the intuitive notion of shape resemblence.

We derived the optimal detection and classification tests using the Nearman-Pearson lemma, and showed that these require marginalization of the distribution. However, in most practical applications, the summation is peaked around the posterior estimate and thus only small regions need to be included in marginalization.

Finally, we performed extensive experimentations to verify the validity of the proposed formulation and reported significant improvement over rigid template in handwritten numeral recognition.

Chapter 5

Conclusions and Future Works

5.1 Conclusions

We have presented an integrated approach in modeling, extracting, detecting and classifying deformable contours directly from noisy images.

We began by conducting a case study on regularization, formulation and initialization of the active contour models. We considered the effect of regularization and showed that one can control the characteristics of the solutions by choosing appropriate values of regularization parameter. We defined a local minimax criterion whereby these values can be automatically and implicitly determined without any user input or heuristics. This criterion possess an intuitively pleasing characteristics of optimizing the worst case behavior, and can be incorporated into energy minimization with no additional computational cost. Subsequently, we formulated a set of energy functionals that impose constraints on smoothness, edge strength and consistency. These energy functionals yield snakes that contain Hough transform as a special case, thus enabling us to initialize the snake efficiently and reliably using Hough transform.

Subsequently, we considered the problem of modeling and extracting arbitrary deformable contours from noisy images. Based on a regenerative shape matrix, our model encompasses both global and local deformations. Combined with the Markov random field to model local deformations, this yields invariant prior distribution that exect influence over an arbitrary global model while allowing for deformation. Using the Bayesian framework, the problem of extracting an object with unknown deformation turns into MAP estimation, which is in turn equivalent to energy minimization in a generalized active contour model.

Finally, we integrated these lower level visual tasks with the high level pattern recognition processes of detection and classification. The g-snake is suitable for these tasks because its shape representation is stable, unique, affine invariant and possesses metric properties. Using the Nearman-Pearson lemma, we derive the optimal detection and classification tests. As the summation is peaked in most practical applications, only small regions need to be considered in marginalizing the distribution. Using this approach, we have reported significant improvement over matched template in handwritten numeral recognition.

5.2 Future Works

In this section, we present two general research directions which can be pursued in the future: *time sequence analysis of deformable contours* and *deformable surface models*.

5.2.1 Time Sequence Analysis

A natural extension of the current work is to consider *time sequence analysis* of deformable contours. Let $\mathbf{U}(t)$ and $\mathbf{g}(t)$ denote the contour and reference point at a discrete time index t. We can model the *evolution* of a contour as follows:

$$\mathbf{U}(t+1) = \mathbf{T}(t)\mathbf{U}(t) + \Psi(t) \quad (5.1)$$

$$\mathbf{g}(t+1) = \mathbf{g}(t) + \mathbf{d}(t) + \kappa(t) \quad (5.2)$$

where $\mathbf{T}(t)$ and $\mathbf{d}(t)$ represent the affine transformation at time t, and $\Psi(t)$ and $\kappa(t)$ represent the local uncertainty. We can tackle various areas of time sequence analysis using this framework:

- *Deformation analysis.* At each time index t. the transformation matrix $\mathbf{T}(t)$ can be estimated from $\mathbf{U}(t+1)$ and $\mathbf{U}(t)$. We can then factorize $\mathbf{T}(t)$ into scale change, rotation or other linear transformations to describe the deformation process of the object of interest. Possible applications exist in digital echocardiography, the study of the dynamic charateristics of the heart.

- *Motion analysis.* Similarly, the $\mathbf{d}(t)$ sequence can be estimated to provide description on velocity and acceleration of the object of interests.
- *Object Tracking.* Assuming knowledge of $\mathbf{d}(t)$, object tracking is reduced into energy minimization after displacing the contour. Note that $\mathbf{T}(t)$ needs not be known, for the shape matrix is implicitly invariant any linear transformation. In some cases, we can adaptively determine $\mathbf{d}(t)$ assuming smooth motion.
- *Object Interaction.* Imagine a scenario where two objects crossing each other while moving in opposite direction. We can eliminate the effect of interaction and track the objects by exploiting the uniqueness of the shape matrix and smoothness of $\mathbf{d}(t)$.

In summary, buliding on the foundation of current work, the model in (5.2) and (5.2) provides a rich set of applications in time sequence analysis which can be investigated in the future.

5.2.2 Deformable Surface Models

The idea of deformable contour models can be extended to encompass deformable surfaces encountered in various computer vision applications. Consider the problem of smoothing an intensity or depth surface without blurring the discontinuities. Clearly this requires a smooth surface which can be bended and twisted by strong evidence of surface discontinuities. In theory, one can achieve this using a *controlled continuity mesh*, in which potential solutions are regularized through some surface smoothness constriants.

A special case of the proposed surface models involves surfaces of uniform depth enclosed by a deformable contour. This gives rise to the idea of deformable *plates* which can be used in image segmentations. This formulation possesses the distinct advantage of being able to adapt to non-convex areas.

References

[1] G. P. Ashkar & J. W. Modestino, "The Contour Extraction Problem with Biomedical Applications," *Computer Graphics Image Processing*, vol. 7, 1978, pp. 331-355.

[2] A. A. Amini, T.E. Weymouth & R. C. Jain, "Using Dynamic Programming for Solving Variational Problems in Vision," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-12, no. 9, 1990, pp. 855-867.

[3] G. P. Ashkar & J. W. Modestino, "The Contour Extraction Problem with Biomedical Applications," *Computer Graphics Image Processing*, vol. 7, 1978, pp. 331-355.

[4] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, 1981, pp. 111-122.

[5] A. Blake, R. Curwen & A. Zisserman, "Affine-invariant Contour Tracking with Automatic Control of Spatiotemporal Scale", *Proc Int. Conf. on Comp. Vis.*, 1993, pp. 66-75.

[6] M. Berger & R. Mohr, "Towards Autonomy in Active Contour Models," *IEEE 10th International Conference on Pattern Recognition*, 1990, pp.847-851.

[7] P. J. Burt & E. H. Adelson,"The Laplacian Pyramid as a Compact Image Code" *IEEE Trans. on Commun.*, vol. **COM**-31, no. 4, 1983, pp. 532-540.

[8] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol **PAMI**-8, 1986, pp. 679-698.

[9] L. D. Cohen, "On Active Contour Models and Balloons," *CVGIP-Image Understanding*, vol. 53, no. 2, 1991, pp. 211-218.

[10] L. D. Cohen & I. Cohen, "Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-15, 1993, pp. 1131-1147.

[11] F. M. Dickey & K. S. Shanmugam, "Optimum Edge Detection Filter," *Appl. Optics*, vol. 16, no. 1, 1977, pp. 145-148.

[12] R. O. Duda & P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communs. Ass. Comput. Mach.*, vol. 15, 1972, pp. 11-15.

[13] h¿ Freeman, "Computer Processing of Line Drawing Images," *Computer Survey* 6, 1974, pp. 57-98.

[14] M. A. Gennert & A. L. Yuille,"Determining the Optimal Weights in Multiple Objective Function Optimization," *Proceedings of Second International Conference on Computer Vision*, 1988, pp. 87-89.

[15] W. L. Grimson & D. P. Huttenlocher, "On the sensitivity of the Hough Transform for Object Recognition," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-12, 1990, pp. 255-274.

[16] W. L. Grimson, D. P. Huttenlocher & T. D. Alter, "Recognizing 3D Objects from 2D Images: An Error Analysis," *Proc CVPR* 1992, pp. 316-321.

[17] R. M. Haralick, "Digital Step Edges from Zero Crossing of Second Directional Derivatives," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-6, no. 1, 1984, pp. 56-68.

[18] Y. He & A. Kundu, "2-D Shape Classification using Hidden Markov Model," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-13, 1991, pp. 1172-1184.

[19] G. E. Hinton, C. K. I. Williams & M. D. Revow, "Adaptive Elastic Models for Hand-Printed Character Recognition," *Neural Information Processing Systems*, vol 4, ed. J. E. Moody *et. al.*, 1992, pp. 512-519.

[20] D. P. Huttenlocher, G. A. Klanderman & W. J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-15, 1993, pp. 850-863.

[21] M. Kass, A. Witkin & D. Terzopoulos, "Snakes: Active Contour Models," *Proceedings of First International Conference on Computer Vision*, 1987, pp. 259-269.

[22] R. Kindermann & J. L. Snell, *Markov Random Fields and their Applications*, American Mathematical Society, 1980.

[23] K. F. Lai & R. T. Chin, "On Regularization, Formulation and Initialization of the Active Contour Models (Snakes)," *Asian Conference on Computer Vision*, 1993, pp. 542-545.

[24] K. F. Lai & R. T. Chin, "On Regularization, Formulation and Initialization of the Active Contour Models (Snakes)," submitted to *IEEE Trans. Pat. Anal. Mach. Intell.*, 1993.

[25] K. F. Lai & R. T. Chin, "Deformable Contours: Modeling and Extraction," *Proc Comp. Vis. Patt. Recog.*, 1994.

[26] K. F. Lai & R. T. Chin, "Deformable Contours: Modeling and Extraction," to appear in *IEEE Trans. Pat. Anal. Mach. Intell.*, 1994.

[27] K. F. Lai & R. T. Chin, "On Classifying Deformable Contours Using the Generalized Active Contour Model," submitted to *Int. Conf. on Automation, Robotics & Comp. Vis.*, 1994.

[28] K. F. Lai & R. T. Chin, "Deformable Contours: Detection and Classification," in preparation.

[29] F. Leymarie & M. D. Levine, "Simulating the Grassfire Transform Using an Active Contour Model," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-14, no. 1, 1992, pp. 56-75.

[30] D. Marr & E. Hildreth, "Theory of Edge Detection," *Proceedings of Royal Society of London*, **B**207, 1980, pp. 187-217.

[31] D. Marr & H. K. Nishihara, "Visual Information Processing: Artificial Intelligence and the Sensorium of Sight," *Technology Review*, **V**81(1), 1978, pp. 2-23.

[32] A. Martelli, "Edge Detection Using Heuristic Search Methods," *Computer Graphics Image Processing*, vol. 1, 1972, pp. 169-182.

[33] F. Mokhtarian & A. K. Mackworth, "A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-14, No. 8, 1992, pp. 789-805.

[34] U. Montanari, "On the Optimal Detection of Curves in Noisy Pictures," *Commun. ACM*, vol. 14, 1971, pp. 335-345.

[35] D. W. Paglierioni & A. K. Jain, "A Control Point Theory for Boundary Representation and Matching," *Proc ICASSP*, vol. 4, 1985, pp. 1851-1854.

[36] M. J. Paulik & N. Mohankrishnan, "Shape Recognition Using a Nonstationary Autoregressive Hidden Markov Model," *Proc ICASSP*, 1991, pp. 2377-2380.

[37] T. Poggio & V. Torre, "Ill-posed Problems and Regularization Analysis in Early Vision, " *Proceedings of AARPA Image Understanding Workshop*, 1984, pp. 257-263.

[38] H. V. Poor, *An Introduction to Signal Detection and Estimation*, Springer-Verlag, 1988.

[39] A. Rattarangsi & R. T. Chin, "Scale-based Detection of Corners of Planar Curves," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-14, 1992, pp. 430-449.

[40] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation and Time Series Analysis*, Addison-Wesley, 1991.

[41] L. H. Staib & J. Duncan, "Boundary Finding with Parametrically Deformable Models," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-14, 1992, pp. 1061-1075.

[42] J. Subrahmonia, J. D. Karen & D. B. Cooper, "Bayesian Method for the use of Implicit Polynomials and Algebraic Invariants in Practical Computer Vision," *Curves and Surfaces in Computer Vision & Graphics 3*, 1992, pp. 104-117.

[43] H. L. Tan, S. B. Gelfand & E. J. Delp, "A Cost Minimization Approach to Edge Detection Using Simulated Annealing," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-14, no. 1, 1992, pp. 3-18.

[44] D. Terzopoulos, "Regularization of Inverse Visual Problems Involving Discontinuities, " *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-8, no. 4, 1986, pp. 413-424, 1989.

[45] S. Umeyama, "Parameterized Point Pattern Matching and Its Application to Recognition of Object Families," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. **PAMI**-15, 1993, pp. 136-144.

[46] D. J. Williams & M. Shah, "A Fast Algorithm for Active Contours and Curvature Estimation," *Computer Vision, Graphics, Image Processing*, vol. 55, 1992, pp. 14-26.

[47] A. L. Yuille, D. S. Cohen & P. W. Hallinan, "Feature Extraction from Faces using Deformable Templates," *Proc. CVPR* 1989, pp. 104-109.

[48] C. T. Zahn & R. S. Roskies, "Fourier Descriptors for Plane Closed Curves," *IEEE Trans. Computers*, vol. **C**-21, 1972, pp. 269-281.

[49] S. Zucker, C. David, A. Dobbins & L. Iverson, "The Organization of Curve Detection: Coarse Tangent Fields and Fine Spline Coverings, " *Proceedings of Second International Conference on Computer Vision*, 1988, pp. 568-577.