Incorporation of Structural Information into Deformable Models

Chris K.Y. Tsang

M. Phil.

Department of Computing

The Hong Kong Polytechnic University

1999

ABSTRACT

Historically, the two major approaches to pattern classification are statistical (or decision theoretic) based and syntactic (or structural) based. In terms of its modelling, syntactic approach is quite satisfactory for rigid objects but not non-rigid ones, without increasing the number of reference patterns for each class. It is even more problematic if the number of classes is very large, like that in Chinese character recognition. Comparatively, statistical approach may be easy enough to handle non-rigid objects. However, it is always a difficult task to choose or designate an effective feature set for such problems. Hence, there is always a need to opt for an alternative approach that does not require sophisticated feature extraction process and is effective in handling non-rigid objects.

Recently, there has been a growing interest in deformable models (DMs). They generally possess shape-varying capability, making them particularly suitable for extracting and recognizing non-rigid objects. When compared with syntactic approach, DM is in fact a kind of flexible graph matching algorithm. Due to its ability to deform, there is basically no need to increase the number of reference models for each pattern class. When compared with statistical approach, it can also be treated as a kind of feature extraction algorithm, in which the resultant value of objective function is the most obvious feature being extracted. In the simplest case, it is possible to designate the objective function value as the only feature, and so there is basically no need to have a sophisticated feature extraction scheme. DMs have been proposed for many different pattern recognition tasks. However, it is observed that most of the existing DMs do not incorporate structural information into the model and can merely deform according to the spatial relationship between primitives. Structural information, which is essential in various pattern recognition tasks, is usually ignored. Even for those attempted to incorporate structural information into the model, most of them are indeed only capable to model open or close contours, without any mechanism to account for highly structural patterns. In this dissertation, we address this issue by proposing a new class of DMs called structural deformable model (SDM) which is capable to model the complex structure of patterns and being able to deform in a well-controlled manner.

The new model takes structural information into accounts by representing an image as a hierarchy of components, namely, image, objects, snakes, segments and snaxels that are structurally connected with each others. It deforms by minimizing the distortion of its inter-object and intra-object structure while matching with the desired image. Concepts like inter-object distance, snaxel evenness, orientation of snaxel edge, and point-to-edge matching are employed in formulating the internal and external energy functionals of the SDM. In addition, a smoothing scheme is introduced to achieve coarse-to-fine matching, making the deformation process behaved in a desirable way. Classification is carried out by treating every deformable matching as a kind of feature extraction process. Two features, namely, resultant objective function value and clustering error are extracted and used by a Bayes classifier to determine the class label of the input image. The effectiveness of the proposed model has been demonstrated through various experiments in Chinese character recognition, which is well-known for its highly structural patterns.

ACKNOWLEDGEMENT

I would like to take this opportunity to express my deepest appreciation to an important person for his careful guidance and supervision, encouragement, patience, valuable time, critical comment, and many nights of sleeping hours to this research and the completion of thesis. He is my supervisor – Dr. Korris Chung.

Besides, I would like to thank Hung, Janis, Boris, yanyan and Benny for their valuable advice and suggestions on the problems that I faced in the research work. Also, I have to thank Wilson for his sharing of valuable opinions on our common research topic. Thanks also give to Daniel, who can be regarded as my math. teacher, for assisting me in solving huge amount of math. problems. Particularly, I would like to thank Ken, who is my closest partner in this two and a half year of research study, for giving me a lot of inspiration in my research.

Specially, I have to thank my father, my mother and my brother for their constant support, immeasurably care and love so that I can finish this research. Moreover, thanks also give to Emerald, Sam, Derek and many lovely brothers in Christ who constantly prayed for me.

Last, but not the least, I would like to thank the Institute for Posts and Telecommunications Policy for providing us with the "IPTP CD-ROM2" database that enables our experiments to be carried out smoothly.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Classical Approaches to Pattern Classification

Computer, after its first commercial introduction in 1951, has demonstrated its effectiveness in offloading humans' works. However, it has been quite limited to those routine jobs like automating a set of accounting ledges. Soon, when our expectation becomes higher and higher, we would like it to do more, in particular those difficult tasks. For example, we are expecting it to be able to do weather forecasting and economic prediction, to do face, speech and character recognition, to do fingerprint identification and signature verification, and even to communicate and talk with us naturally. Thus, researchers have been trying every effort in making computers more intelligent. However, it is certainly not easy, particularly for pattern recognition tasks. Being able to differentiate and understand complex patterns is still a unique capability of human beings.

Historically, the two major approaches to pattern classification are statistical (or decision theoretic) based [1-3] and syntactic (or structural) based [1,4-5]. Statistical approach relies on a statistical framework for classification under which a set of representative features of the pattern of interest is extracted and classified into one of a finite number of pattern classes. For each pattern class in the feature space, it is usually specified by a multivariate probability distribution function that may be known *a priori* or estimated from a series of training samples. The classifier on the other hand is designed typically with the criterion of minimizing the Bayesian error probability or a cost function based upon it. However, there is always a concern that the pattern cannot be fully described by the numerical feature values. The relationship between features, which yields the so-called structural information, should also be an important cue to effective classification. In this regard, syntactic approach was advocated, aiming at taking the structural information into accounts. One syntactic approach [1,4-5] is to relate the structure of patterns with the syntax of a formally defined language. By constructing a grammatical model for each pattern class characterized by its structure primitives, a pattern is classified to a pattern class whose grammatical model can generate itself by parsing technique.

In these two classical approaches, an image[1] characterized by quantifiable and measurable features is matched against each pattern class and classified to the one with highest similarity. In terms of modelling technique, syntactic approach is quite satisfactory in modelling rigid objects like machinery parts but not those non-rigid objects such as handwriting patterns, gestures and human faces without increasing the number of reference patterns for each class. It is even more problematic if the number of classes is very large, like the case in Chinese character recognition with over thousands of pattern classes. Comparatively, statistical approach may be easy enough to handle non-rigid objects. However, it is always a difficult task to choose or designate an effective feature set for such problems, particularly when the shape of the objects varies significantly, e.g., different writing style of handwritings. Could it be possible to by-pass these problems fundamentally ? Is there any alternative approach that does not require sophisticated feature extraction process and is effective in handling non-rigid objects ?

1.2 Deformable Models

Recently, there has been a growing interest in deformable model (DM) based approach which is considered being situated somewhere between the previous two conventional approaches. When compared with syntactic approach, DM is in fact a kind of flexible graph matching algorithm. Due to its ability to deform, there is basically no need to increase the number of reference models for each pattern class in classification problems. When compared with statistical approach, it can also be treated as a kind of feature extraction algorithm, in which the most obvious feature being extracted is the resultant value of objective function. In the simplest case, it is possible to designate the objective function value as the only feature, and so there is basically no need to have a sophisticated feature extraction scheme.

DMs have been proposed for many different pattern recognition tasks as they generally possess shape-varying capability, making them particularly suitable for extracting and recognizing objects with large shape variations. DMs have been shown to be able to capture the natural variability within a class of shapes and can be used in image search to find examples of the structures that they represent [6]. They have been successfully applied to edge and subjective contour detection [7], motion tracking [8], object matching [9] and more recently handwriting recognition [10-15].

DM is mainly characterized by some energy terms/functionals that govern the way it deforms and moves onto the image. On one hand, it is required to preserve its original shape while on the other hand, it is required to match with the desired object or objects in an image. This can be achieved by two opposing forces, namely, internal and external (or image) force. Matching is performed by minimizing the total energy to attain equilibrium of the two forces.

The development of DMs has been carried out for about two decades. Widrow [16] is considered the first exploring DMs by constructing a so-called "Rubber-Mask" to classify human chromosome and highly irregular waveforms of different kinds. Later in 1981, Burr [17] tried to apply elastic model technique to line drawings by defining a set of pushing and pulling forces as a kind of displacement vectors which align the line drawing model with the target one in a number of stretches.

In 1988, Kass *et al.* [18] tried to formulate DM as "Snakes" which is in fact an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges, while being able to preserve a certain kind of local continuity and smoothness. All forces are specified

[1] *Unless otherwise stated, we will focus on image patterns in the rest of the dissertation*

by energy functionals, and thus the resulted deformation task has been formulated as an energy minimization problem.

Such energy minimizing technique has now been widely used by various DMs to tackle different object identification, extraction, and classification problems. However, among existing DMs, it is observed that most of them model their shapes mainly as a set of the relative displacement between model primitives. Structural information, which is essential in various pattern recognition tasks, is usually ignored. Consequently, deformation can only be made based on the spatial relationship between primitives rather than on the their structure.

Jain *et al.*'s rubber sheet model [19] is a typical example of DMs of which the model does not explicitly carry any structural information. It has been shown to be effective in handwritten digit recognition [10]. To control how the rubber sheet deforms, a displacement function, which consists of trigonometric functions of different frequencies, varying from global and smooth to local and coarse, is defined for each location of the rubber sheet. Although such DM can be used to model almost any 2D image pattern, nearly all structural information inside the pattern has been ignored. Simulations have been conducted on this type of DMs and it is observed that the deformation being made on the spatial relationship only is not quite natural. The DM proposed by Wakahara [11] has adopted a different way to do deformation, i.e., based on local affine transformation (LAT) technique. Matching is carried out by determining the LAT parameters that yield the best match in each local area of two images. Again, no structural information has been explicitly incorporated in its modelling.

Some researchers have proposed spline-like DMs, like the Generative Model [12], for handwritten digit recognition. Each digit is modelled by a deformable B-spline with "ink generators" spaced along the path of the spline. The similarity between patterns is interpreted as how likely the input pattern is generated by those "ink generators". Although some form of structural information has been incorporated into the model, it can merely model open or close contours but not highly structural patterns. Similar to spline-like DMs, the snake-like DMs, e.g., [13], have also been proposed to recognize handwritten digits. Structural information has been taken into considerations but it is only limited to local properties such as local continuity and aspect ratio. No higher level structural information has been incorporated.

On the other hand, Nishida and Mori [14] tried to model global structure by developing a novel DM characterized by its quasi-topological features (i.e., convexity and concavity), directional features, and singular points (i.e., branch points and crossings). A number of structural transformation rules or operators are also defined for removing or merging primitives inside the structural patterns, while keeping its so-called global structure preserved. Although global structural information has been incorporated into the model, too much freedom is allowed and hence the model may easily deform into many shapes not necessarily similar to the original structure. Hence, it may not be under proper control when dealing with highly structured patterns like Chinese characters.

1.3 Objectives of the Research

In view of the superiority of DMs in handling non-rigid objects and the limitations of most existing DMs for their deficiency in handling structural information, the main objective of this work is to develop a new class of DMs called structural deformable model (SDM) which

- explicitly take the structural information into accounts,
- incorporate it into its modelling process, and hence
- can handle highly structural patterns like Chinese characters and signatures.

As mentioned in previous section, the structural information has been ignored by most existing DMs in their modelling process and consequently deformation takes place in the spatial domain only. Even for those who attempted to incorporate structural information into the model, most of them are indeed only capable to model open or close contours, without any mechanism to account for highly structural patterns. There are still some others who tried to model the global structure of patterns, however, it is found that their models may easily deform into undesirable shapes for their overmuch deformation flexibility. Hence, in this research, we aim at devising a model that is capable to model complex structure and can deform in a well-controlled manner.

Despite the intention to develop a class of SDMs for object classification as well as object extraction (potentially being useful in content-based image retrieval [20]), we have chosen to concentrate on Chinese character recognition, which is well-known for its highly structural patterns. In this regard, the SDM is expected to be a post-processor of statistical or syntactic recognizers, i.e., it helps to pick a much more accurate class label from a list of *n* choices produced.

1.4 Organization of the Thesis

The thesis is made up of six chapters. In Chapter 2, general concepts about DMs will be introduced. A probabilistic interpretation of deformable matching is also included which serves as the basic of the Bayesian framework for SDM described in Chapter 4. At the end of the chapter, several kinds of classification of DMs are presented and they are the conceptual tools being used to derive the proposed model in subsequent chapters. Chapter 3 is devoted to describe the proposed model, from preprocessing, model structure hierarchy down to the derivation of various energy functionals. Chapter 4 describes the integration of the SDM into a Bayesian framework as well as the incorporation of global-to-local deformation ability into the SDM in order to enable it to deform in a well-controlled manner. The classification scheme adopted by the SDM is also presented there. The effectiveness of the proposed model demonstrated through various experiments is reported in Chapter 5 where parameter sensitivity analysis, feature extraction analysis and functionality analysis are also included. The last chapter concludes the thesis by summarizing the contributions and limitations of the proposed model together with some suggestions for further research.

Chapter 2

DEFORMABLE MODELS

2.1 General Concepts

Deformable model (DM) is generally referred to a kind of models which possess shape-varying capability [18]. Once a pattern is captured or modelled by a DM, it is changed from a rigid body to a non-rigid one which in turns can deform itself freely. This technique is quite useful in various matching problems. Instead of matching by simple affine transformations or some regular transformations, the reference pattern modelled by a DM can be matched by irregular transformations. One may call this kind of irregular transformation as deformable transformation or simply deformation. However, as DM may keep distorting itself in an attempt to match with the target pattern, there is a problem of defining the termination criteria and the similarity measure. In fact, two criteria are required for desirable deformation, namely, shape preservation and data match.

2.1.1 Shape Preservation Criterion

The preservation of the model shape during deformation can be achieved by using an internal energy which measures how good/bad the model maintains its original shape. The smaller the internal energy, the better the model has preserved its shape. It is this internal energy which stops the model from deviating too much from its representative class.

Consider a particular type of DMs called active contour model or snake proposed by Kass *et al.* [18]. It is in fact an elastic spline that is slithering under forces created by some kinds of energy and is represented by a set of ordered points $\{v_i = (x_i, y_i) \mid i \in c\}$ where v_i denotes the position vector of each interpolated node called snaxel along the snake contour c. Figure 2.1 shows an example of a snake consisting of eight snaxels. An internal energy functional E_{int} which is designed to preserve the shape of the snake [18], is formulated as

$$E_{int} = \int_c \left(\alpha |v_i'|^2 + \beta |v_i''|^2 \right) di \tag{2.1}$$

where the first and second spatial derivatives with respect to snaxels correspond to the evenness and smoothness of the snake respectively. In this case, when the shape of the snake, i.e., the evenness of inter-snaxel distance or contour smoothness, deviates too much from the original one, internal energy will be comparatively high and the snake will subsequently experience a friction/force which stops it from further distorting its shape.

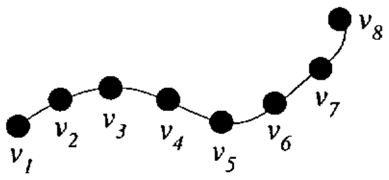


Figure 2.1 A snake composing of eight snaxels

2.1.2 Data Match Criterion

In DMs, matching between patterns can be achieved via an external energy which measures the data mismatch. The higher the external energy, the larger the data discrepancy between the model and the target pattern. Such a discrepancy provides a guidance or force for the model to deform towards the target pattern.

For example in contour extraction applications, a snake is expected to move towards pixel positions with shape edges. The corresponding external energy functional can be formulated as

$$E_{ext} = \int_c \left(-\left| \nabla I(v_i) \right| \right) di \tag{2.2}$$

where $\nabla I(v_i)$ is the edge magnitude or intensity gradient of the pixel coincident with snaxel v_i. In this case, when the snake is adjacent to an edge, external energy will be comparatively high (compared with the case when the snake is situated on an edge) which in turn induces an attraction force to the snake. Under the influence of this attraction force, the snake will move towards the edge and external energy will be decreased subsequently. When the external energy becomes small, the snake knows that it is close to the edge and will stop moving eventually.

2.1.3 Regularization

Intuitively, we enable the model to deform; but if the model shape starts to distort, we should stop the process. Thus, the internal and external energies should be considered collectively and the resultant DM will have shape-varying capability together with shape preservation ability. To combine these two conflicting criteria, i.e., data match and shape preservation, the most common way is to define a combined criterion function as a weighted sum of internal and external energy functionals, which is given by

$$E_{tot} = w_{int} \cdot E_{int} + w_{ext} \cdot E_{ext} \tag{2.3}$$

This technique is called regularization [15] and trade-off can be made by properly setting the regularization parameters, w_{int} and w_{ext}.

2.2 Matching

Loosely speaking, when a DM has reached a state that it is close or align with (if possible) the target pattern and not deviated too much from its original shape, the deformation is considered to be finished. In that sense, the final state corresponds to an equilibrium of the two energy functionals. Seeking the equilibrium state can be achieved by minimizing the combined criterion function E_{tot}. There exists a number of minimization schemes ever proposed, see, e.g., Dynamic Programming [21], Greedy Algorithm [22] and Hopfield Network [23]. Thus, the elastic matching process is eventually formulated as an energy minimization problem.

From a probabilistic point of view, internal energy E_{int} and external energy E_{ext} of a DM can be interpreted in a different way. The former can be interpreted as the uncertainty that it is deformed from some particular model while the latter can be interpreted as the uncertainty that the data come from some particular model. With the help of Gibbs distribution, these two uncertainties can be converted to probability distributions

$$p(M)=\frac{1}{Z_{int}(\sigma_{int})}\exp\left(-\frac{E_{int}}{\sigma_{int}}\right) \tag{2.4}$$

and

$$p(I\,|\,M)=\frac{1}{Z_{ext}(\sigma_{ext})}\exp\left(-\frac{E_{ext}}{\sigma_{ext}}\right) \tag{2.5}$$

respectively, where σ_{int} and σ_{ext} control the spread of the probability distributions, Z_{int} and Z_{ext} are used to normalize the two distributions. Under the probabilistic interpretation, $p(M)$ and $p(I\,|\,M)$ are called prior distribution and likelihood respectively. By using Bayes rule, they can be combined to obtain the a posteriori probability density of the deformed model given the input image, i.e.,

$$p(M\,|\,I)=\frac{p(I\,|\,M)\cdot p(M)}{p(I)} \tag{2.6}$$

The problem of deformable matching is usually formulated as a problem of maximizing the a posteriori probability $p(M \mid I)$, which in turn is equivalent to minimizing the Bayes objective function given by

$$\Psi = \frac{1}{\sigma_{int}} \cdot E_{int} + \frac{1}{\sigma_{ext}} \cdot E_{ext} \tag{2.7}$$

In fact, this objective function is exactly the same as E_{tot} in eq.(2.3) with regularization parameters as $\frac{1}{\sigma_{int}}$ and $\frac{1}{\sigma_{ext}}$. As a result, one may consider the importance of the probabilistic interpretation of deformable matching more than just a nice theoretical framework.

2.3 Classification of DMs

DMs can be categorized in many different ways. For example, in terms of modelling techniques, DMs can be categorized into parametric and non-parametric one. In this section, we further introduce several classification schemes of DMs which serve as important conceptual tools to develop the proposed model in subsequent chapters.

2.3.1 Image Guided Search (IGS) and Model Guided Search (MGS)

DMs can be employed for searching the desired objects elastically in an image for its shape varying capability. This kind of search can be categorized into a local one or an extensive one. For a local search, the DMs will merely look for the nearby image primitives actively by itself. Not all image primitives will provide image force for it to deform. We call this feature as model guided search (MGS). For the extensive search, on the contrary, the DMs will search over the whole image. All image primitives will provide image forces for its deformation. In this way, DMs seem to possess a global picture of the image and searching is guided by all image primitives. We call this feature as image guided search (IGS). The notion similar to the categorization of DMs into MGS and IGS like the forward matching and backward matching proposed by Cheung *et al.* [24-25].

Almost all contour extraction problems tackled by DMs are solved using MGS. Using MGS will result in finding a sub-image as it searches for the image locally and does not account for the whole image. For problems like object location where it does not require to account for the whole image, searching sub-image is already enough to achieve the goal. DMs like Snake [18] and the rubber sheet model [19] are using MGS. An important drawback of models using MGS is that they will easily get trapped in local minimum either under the influence of noise or undesirable image primitives.

On the other hand, for matching problems, we are required to determine if two patterns are in total match rather than locating a sub-image. From image explanation point of view, the problem can be tackled by examining if every part of the image can be explained by the model. In this way, IGS should be used instead. The advantage of using IGS is that initializing the model close to the image is not necessary. Instead of searching actively by the model, huge amount of information (i.e., the entire image) is provided for movement guidance. DMs like Generative Model [12] are using IGS. However, in order to achieve a total match between patterns, the following two requirements are considered necessary:

1. Every part of the image should have some model primitives nearby, and
2. every part of the model should have some image primitives nearby.

In fact, adopting IGS satisfies the former while adopting MGS satisfies the latter. As a result, the integration of both schemes is proposed in the SDM.

2.3.2 Pixel Matching and Edge Matching

In previous section, we mentioned model primitives and image primitives which are employed by DMs for image matching. The most commonly used primitives are pixels and edge segments, and the resultant matching schemes are called pixel matching and edge matching respectively.

By adopting pixel matching [11,12,19,26], all pixels will participate in the matching process, and thus no information of the pattern image are missed. However, it is quite time consuming to carry out pixel matching and hence some tackle the problem by using multi-resolution approach in which matching is carried out in a coarse-to-fine manner [19].

On the other hand, due to the fact that pixel matching is computationally expensive, some researchers chose to adopt edge matching [17,27]. In the literature, among those adopting edge matching approach, some of them tried to use one-to-one edge mapping. For example, Wakahara and Odaka [27] employed one-to-one edge matching technique for the problem of on-line Kanji character recognition. Since it is on-line recognition, one-to-one stroke correspondence could be easily extracted from the available temporal information, even before actual deformation taking place. However for off-line cases tackled by, say, relaxation approach of elastic matching [28], it is not so easy to find accurate stroke correspondence without the requirement of a sophisticated preprocessing like redundant strokes deletion, connected strokes separation, and possible substrokes mergence. There exist some other researchers considering other types of edge matching, like many-to-many edge mapping employed by Burr [17]. No matter what edge matching technique one has adopted, processing time will be substantially reduced compared with that of pixel matching.

2.3.3 Active Shape Preservation (ASP) and Passive Shape Preservation (PSP)

In Section 2.1, we have mentioned two deformation criteria, namely, shape preservation and data match. In general, all DMs consider both criteria, but not all of them will define internal and external energy functionals to realize them. Instead, one may employ some schemes to replace one or both of the energy functionals. For example, Burr [17] realized two deformation criteria by defining a scheme in which displacement vectors correspond to image force and neighbourhood of influence corresponds to internal force. In this scheme, at each iteration of deformation, all displacement vectors will be updated and then a smoothing function is applied to produce a smoothed version of the displacement vectors. Deformation can be performed accordingly. On the other hand, Wakahara's model [11] only made use of an external energy functional which is minimized by computing two parameters used by the Local Affine Transformation (LAT). No internal energy functional is used because the shape preservation criterion has already been considered in LAT by using the neighbourhood of influence.

Although whether using energy functionals or not is the way of realizing two deformation criteria, it reflects how DMs preserve their shape (Note that the model shape is defined differently in different models). For those who made use of two energy functionals to realize the two deformation criteria, minimization can not be carried out solely according to the internal energy but also to the external energy. By minimizing both energies via regularization, the total energy can be made monotonically decreasing but it does not hold for the individual energy term. As a result, the preservation of the model shape can not be ensured due to the need to minimize external energy. This phenomenon is due to the fact that the shape is not preserved actively, i.e., instead of restricting its movement to preserve its shape at each iteration, they defined an internal energy together with an external energy, hoping that its shape can be preserved by minimizing both of them. We call this kind of shape preservation mode as Passive Shape Preservation (PSP).

For those who made use of some schemes to preserve the model shape, like those in Burr's model [17] and Wakahara's model [11], its movement is restricted at each iteration to preserve the overall model shape. As the shape is actively preserved, we named such kind of shape preservation mode as Active Shape Preservation (ASP). The main difference between PSP and ASP is that the latter has a higher chance of preserving the model shape.

Chapter 3

STRUCTURAL DEFORMABLE MODEL (SDM)

3.1 System Overview

The proposed system using SDM can be divided into five main modules; namely, data acquisition, preprocessing, modelling & initialization, deformation and finally object classification and extraction. Its flowchart is depicted in Figure 3.1. In this dissertation, we will concentrate on applying SDM to object classification problems, in particular handwriting recognition, and the discussion hereafter will only be referred to this type of applications unless otherwise stated.

In data acquisition module, image patterns (in binary format) are either captured via a flat-bed scanner or obtained from an available database. For object classification applications, an unknown image as well as a number of class representative images (templates) will be fed into the next module. We may either feed the whole set of templates for problems with small number of classes, e.g., handwritten digits classification, or feed merely those most probable template candidates which can be selected by users or other recognition systems, e.g., a statistical recognition system with top *n* choices. In the latter case, SDM will behave like a postprocessor of another system.

Upon receiving the required images, the preprocessing module starts with a noise removal process which filters out those salt-and-pepper noises. The cleaned images will undergo a thinning process which extracts the line skeletons of the objects (characters). Then, feature points such as 3-fork and 4-fork points will be extracted from the thinned images, which will be employed by the modelling process in next module.

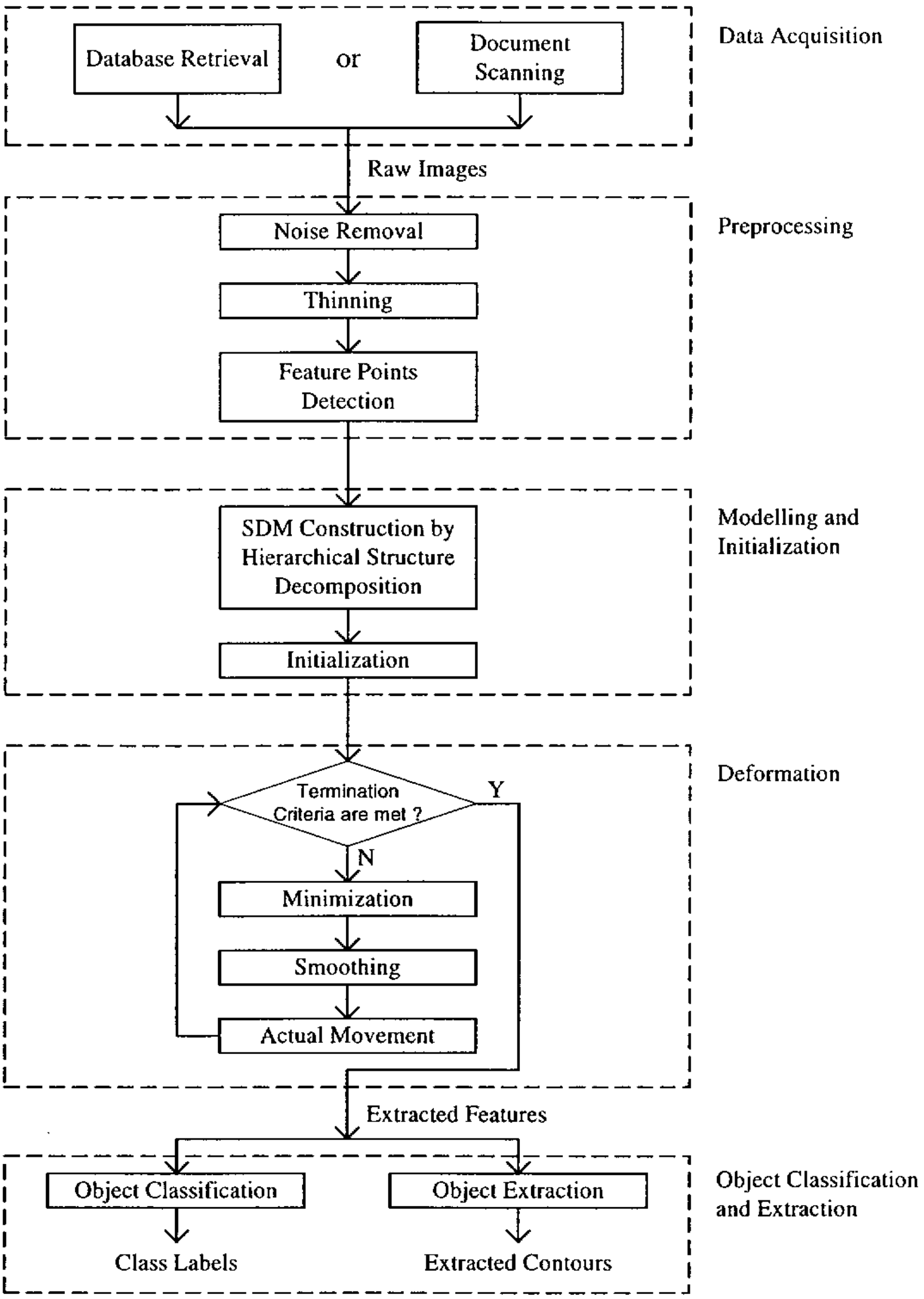


Figure 3.1 A flowchart of the SDM recognition system

In modelling and initialization module, templates are modelled as a hierarchy of structure elements which will be described in details in Section 3.3. Based on the structure hierarchy, deformation can be made accordingly. The unknown image on the other hand is simply modelled as a set of sampled pixels without any structure associated, which will be fixed during the deformation process. After the modelling step, initialization takes place for the unknown image which will firstly be size-normalized and center-positioned. Then, each template will be initialized to the image as close as possible before deformation starts.

The deformable matching of each template with the unknown image is carried out in the deformation module. It consists of three processes which are performed iteratively, namely, minimization, smoothing and actual movement. In order to enable the model to deform in a well-controlled manner, a global-to-local deformation ability is incorporated. It is realized by applying a smoothing process in each iteration to convert the set of displacement vectors resulted from the minimization to a set of smoothed displacement vectors before the actual movement taking place. Such smoothing process will be described in detail in Section 4.2.

In fact, the deformable matching itself can be interpreted as a kind of feature extraction process which will be elaborated in Section 4.3. So after deformation, a set of features representing the dissimilarity between the image and each template is extracted. They will be employed by a dedicated classification algorithm to determine the class label. Details of this module will be presented in next chapter.

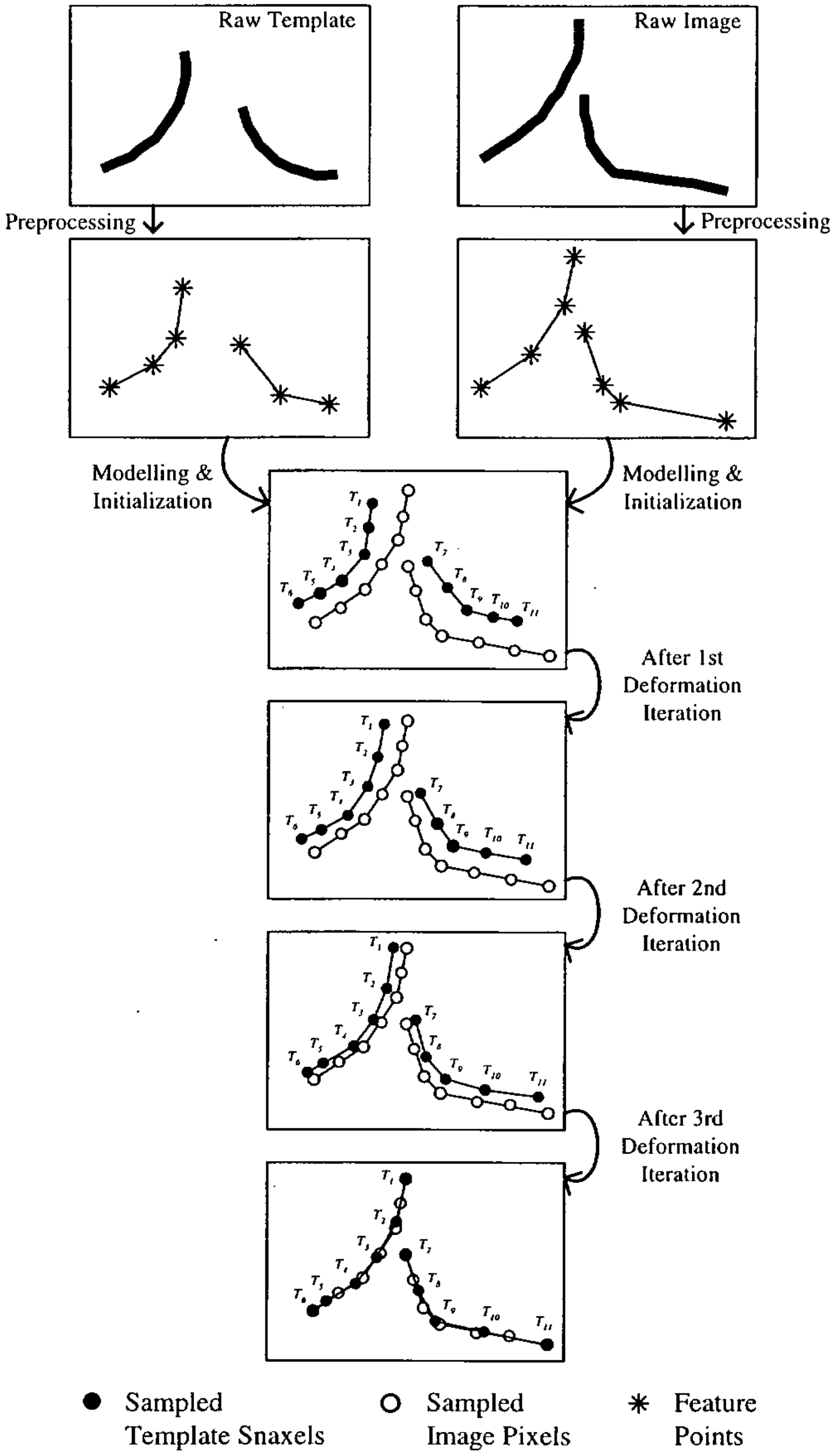

Raw Template
Raw Image
Preprocessing
Preprocessing
Modelling & Initialization
Modelling & Initialization
After 1st Deformation Iteration
After 2nd Deformation Iteration
After 3rd Deformation Iteration
T_1
T_2
T_3
T_4
T_5
T_6
T_7
T_8
T_9
T_{10}
T_{11}
● Sampled Template Snaxels
○ Sampled Image Pixels
✳ Feature Points

Figure 3.2 An overall process of the system using SDM on character recognition

In Figure 3.2, an example showing the overall process of the system using SDM on character recognition is depicted. Supposing that in the data acquisition module a raw template and a raw image are captured, the system is required to deform the template towards the image in order to extract some features of their dissimilarity for classification. In the preprocessing module, they undergo a noise removal and a thinning process. Besides, feature points are extracted and are denoted by stars in the figure. In the next module, the template pattern is modelled as a hierarchy of structure elements which will be described in Section 3.3 and is finally sampled by a set of snaxels denoted by solid circles. Based on the structure hierarchy, the deformation afterwards can be made accordingly. The image on the other hand is simply modelled as a set of sampled pixels (denoted by hollow circles) which are interpolated between feature points previously extracted and are kept fixed during the deformation process of the template. In this case, 11 snaxels and 14 pixels are sampled respectively. Then these two patterns are initialized such that the template pattern will be brought to the image as close as possible to facilitate the subsequent deformation. The resulted position of two patterns is shown in the figure. In the deformation module, an iterative application of energy minimization and smoothing process is carried out and three deformation snapshots are shown. Finally, features indicating the dissimilarity between two patterns from the final deformation snapshot like the resultant value of objective function are extracted for the Bayesian classification.

In the rest of this chapter, the preprocessing module will be described first. Details of the modelling & initialization module will be given in Section 3.3. Before elaborating the deformation and classification modules, various energy functionals which serve as an engine for giving the SDM ability to move (deform) will be presented in Section 3.4 and 3.5.

3.2 Preprocessing

3.2.1 Noise Removal

In order to incorporate the SDM with a capability to differentiate highly similar patterns, all pixels inside the image will be treated as valuable matching primitives. As a result, a noise removal process is considered necessary to be carried out beforehand. Due to its simplicity, the removal of salt-and-pepper noise is introduced in which a 3×3 window mask depicted in Figure 3.3 is required.

P_8	P_1	P_2
P_7	P_0	P_3
P_6	P_5	P_4

Figure 3.3 A 3 × 3 window mask

A neighbourhood count $\gamma(P_o)$ representing the total number of nonzero neighbours of P_0 is given by

$$\gamma(P_o) = \sum_{i=1}^{8} P_i \tag{3.1}$$

where $P_i \in \{0,1\}$. Salt-and-pepper noise is removed by sliding the window mask on the image from upper left to lower right and setting the center pixel P_0 according to the following rule:

$$P_o = \begin{cases} 1 & \text{if } \gamma(P_o) = 8 \\ 0 & \text{if } \gamma(P_o) = 0 \end{cases} \tag{3.2}$$

3.2.2 Thinning

Like the other DMs, the serious problem that exists in the proposed model is the great computational cost. To make it work as efficient as possible, a thinning process is introduced to reduce the total number of matching primitives in deformable matching. A fast parallel thinning algorithm proposed by Zhang and Suen [29] has been adopted for the efficiency and the effectiveness being claimed. For the sake of self-containment, a general concept of how it works is described. In this algorithm, a crossing count $\chi(P_o)$ representing the connectivity of P_0 is defined as

$$\chi(P_o)=\frac{1}{2}\sum_{i=1}^{8}|P_{i+1}-P_i| \tag{3.3}$$

where P_9 refers back to P_1 in Figure 3.3. The topological meaning of χ is clear and is shown in Table 3.1.

The method iteratively removes contour pixels in two sub-iterations: one aimed at deleting the south-east boundary points and the north-west corner points while the other one aimed at deleting the north-west boundary points and the south-east corner points. The iterations continue until no more pixels can be removed in either sub-iteration. In order to preserve end points of each skeleton line and pixel connectivity, some other constraints of pixel deletion are incorporated and are summarized in Table 3.2.

Table 3.1 Topological meaning of different values of $\chi(P_o)$

Value of $\chi(P_o)$	Meaning of P_0
0	Isolated point
1	End point
2	Link point
3	3-fork point
4	4-fork point

Table 3.2 Conditions of contour pixel deletion in thinning

P_0 is deleted if following conditions are satisfied :	
1st sub-iteration	2nd sub-iteration
a) $2 \le \gamma(P_o) \le 6$ and	a) $2 \le \gamma(P_o) \le 6$ and
b) $\chi(P_o) = 1$ and	b) $\chi(P_o) = 1$ and
c) $P_1 \times P_3 \times P_5 = 0$ and	c) $P_1 \times P_3 \times P_7 = 0$ and
d) $P_3 \times P_5 \times P_7 = 0$	d) $P_1 \times P_5 \times P_7 = 0$

3.2.3 Feature Points Detection

In order to make use of structural information inside the image being modelled, it is suggested to designate some primitives as ordinary feature points and some as perceptual important points based on their structure information embedded, rather than treating all of them as the same kind. It is this set of feature points that decomposes the model into a hierarchy of structure elements which in turn enables the model to deform according to the pattern structure. Ordinary feature points that we are in interests consist of end points, 3-fork points and 4-fork points. By using the crossing count χ defined in eq.(3.3) and according to the Table 3.1, ordinary feature points can be easily extracted.

Perceptual important points are also required by the modelling process in next module. They carry higher level of information compared with the ordinary one, and in our case are referred to those points with high curvature. They can be extracted by polygonal approximation of curves and the resultant vertices are the points we look for. Before the polygonal approximation of curves is carried out, it is necessary to extract all the curves terminated by ordinary feature points. It can be achieved by tracing through link points from each ordinary feature point to the others. By employing the algorithm proposed by Lowe [30], which is stated to be the most optimum algorithm in terms of fidelity and efficiency in polygonal approximation of curves [31], a set of straight-line segments for each curve are located. For the sake of self-containment, the general concept of Lowe's algorithm is described.

Lowe's algorithm approximates curves by straight line segments. Firstly, the significance of a straight line fit to a list of points is defined as a ratio of the length of the line segment divided by the maximum deviation of any point from the line. Then, a segment is recursively subdivided at the point with maximum deviation from a line connecting its endpoints. This process is repeated until each segment is no more than 4 pixels in length, producing a binary tree of possible subdivisions. Next, unwinding the recursion back up the tree, a decision is made at each junction as to whether to replace the current lower-level description with the single higher-level segment. If the maximum significance of any of the sub-segments is greater than the significance of the complete segment, then the sub-segments are returned; otherwise, the single segment is returned.

After approximating all curves by straight-line segments, perceptual important points correspond to the vertices of these line segments and are subsequently utilized by the next module.

3.3 Modelling and Initialization

3.3.1 Hierarchical Structure of SDM

Given an image to be modelled, in order to take its structure information into account, it is modelled as a hierarchy of structure elements depicted in Figure 3.4. The top-most level of the model (SDM) is the image itself. Inside the image, it may consist of a number of separated components. Each of them is referred to an object here. To relate all objects inside an image, pseudo-connections that will be described in Section 3.3.3 are employed. Each object is then represented as a set of active contours (snakes) that are structurally connected with each other. Traditionally, snake is composed of snaxels. But in the proposed model, one more encapsulation is introduced in between. It is segment and can be extracted via ordinary feature points and perceptual important points described previously. The former ones are used to decompose an object into a number of snakes while the latter ones are used to divide a snake into a number of segments. Finally, for each segment, it will be interpolated by snaxels characterized by their position vectors and associated edge orientation vectors.

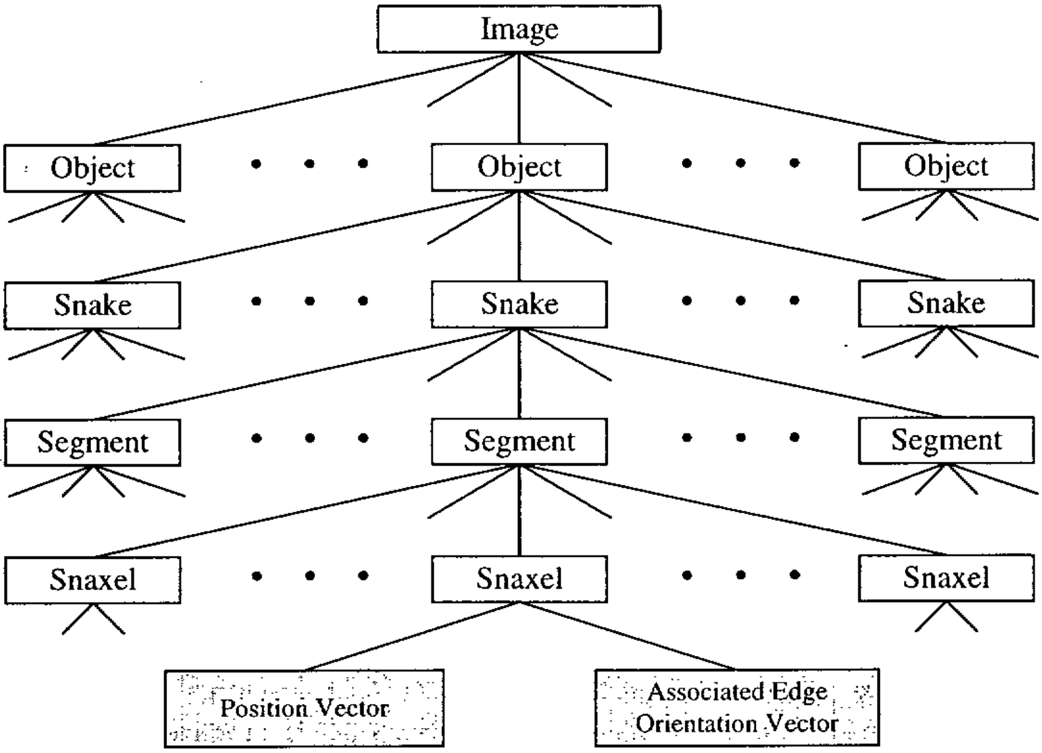
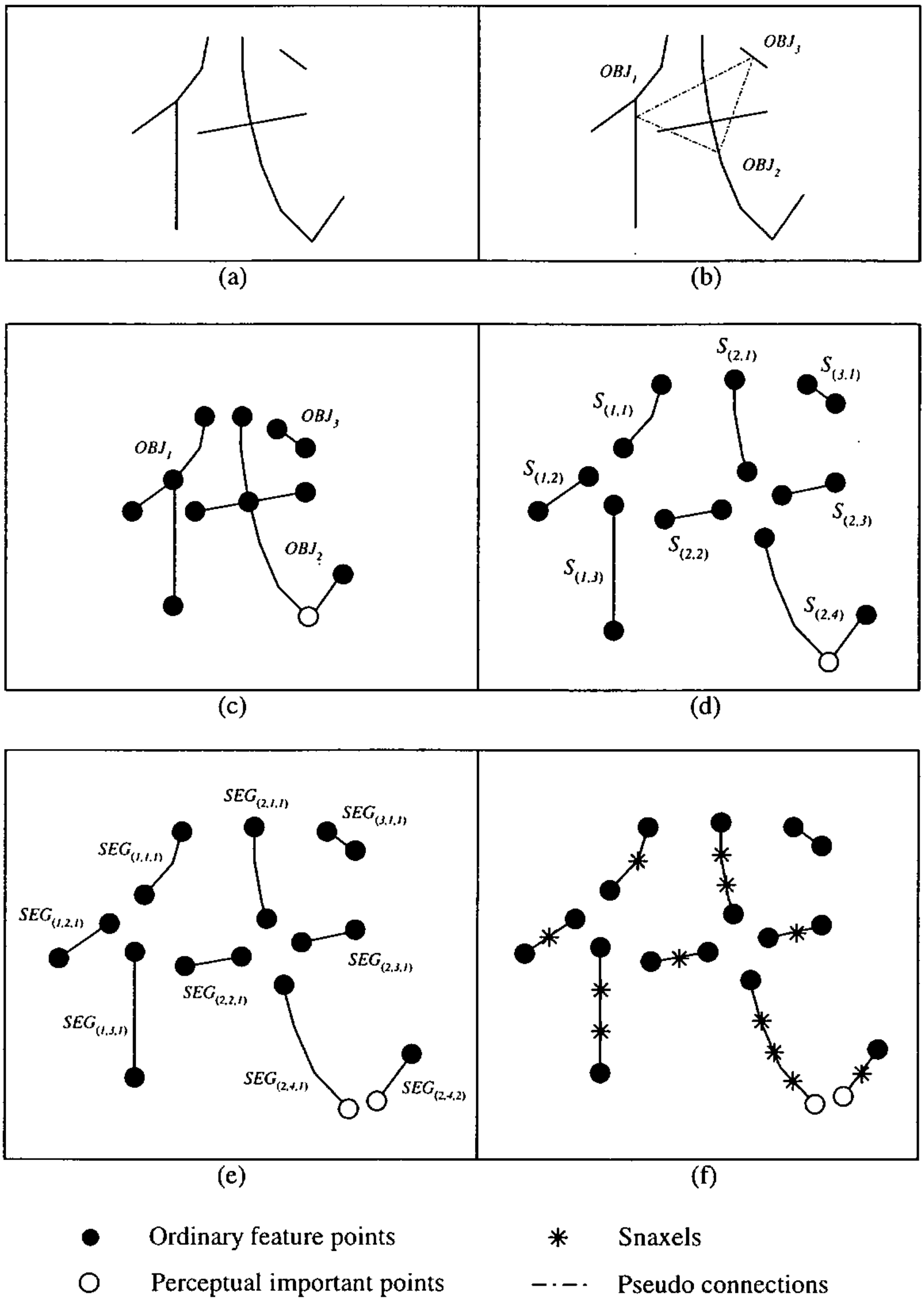


Figure 3.4 A hierarchical structure of SDM

To demonstrate the modelling process, let's consider a Chinese character image in Figure 3.5(a). Inside the image, three separated components are found and each of them is called an object OBJ_i. To maintain the inter-object structure, pseudo-connections joining the centroids of every two objects are introduced and are denoted by dotted lines in Figure 3.5(b). The next step is to extract all ordinary feature points and perceptual important points. They are denoted by solid circles and hollow circles respectively in Figure 3.5(c). After all the feature points are extracted, the image is ready to be decomposed into snakes and segments. Firstly, ordinary feature points will be utilized here for object decomposition. The existence of a 3-fork point will result in three separated snakes while that of a 4-fork point will result in four separated snakes. As shown in Figure 3.5(d), the object OBJ_1 is decomposed into three snakes, $S_{(1,1)}$, $S_{(1,2)}$ and $S_{(1,3)}$ while the object OBJ_2 is decomposed into four snakes, $S_{(2,1)}$, $S_{(2,2)}$, $S_{(2,3)}$ and $S_{(2,4)}$. For object OBJ_3, there is no 3-fork points or 4-fork points and only one snake $S_{(3,1)}$ is resulted. For snake decomposition, a snake will be bisected into two segments for every perceptual important point being found. As shown in Figure 3.5(e), the snake $S_{(2,4)}$ has a perceptual important point and hence it is decomposed into two segments, i.e., $SEG_{(2,4,1)}$ and $SEG_{(2,4,2)}$. As there is no perceptual important points in all other snakes, only one segment for each of them is resulted. The final step of the hierarchical structure decomposition is the interpolation of snaxels which are denoted by stars in Figure 3.5(f). In our experiments, the interpolation interval of snaxels and pixels are fixed at 1/10 of the image normalization size. Based on the structure hierarchy just developed, internal energy functional(s) can be formulated accordingly, giving the model an ability to deform according to the pattern structure.



(a) (b) (c) (d) (e) (f)

Figure 3.5 A hierarchical structure decomposition process of a Chinese character image. (a) The original image. (b) Image decomposition. (c) Feature points extraction. (d) Objects decomposition. (e) Snakes decomposition. (f) Snaxels interpolation.

3.3.2 Formal Representation of SDM

The formal representation of SDM is given here. According to the hierarchical structure of SDM presented in previous section, the SDM consisting N objects is given by

$$SDM = \left\{ \; OBJ_o \mid o = 1, \cdots, N \; \right\} \tag{3.4}$$

where OBJ_o is the o-th object in an image. With the help of ordinary feature points, each object can be decomposed into a set of inter-connected snakes and is given by

$$OBJ_o = \left\{ \; S_{(o,s)} \mid s = 1, \cdots, N_o \; \right\} \tag{3.5}$$

where $S_{(o,s)}$ is the s-th snake inside the object OBJ_o and N_o is the total number of snakes inside. Again, with the help of those perceptual important points, each snake is further decomposed into a set of segments, i.e.,

$$S_{(o,s)} = \left\{ \; SEG_{(o,s,g)} \mid g = 1, \cdots, N_{(o,s)} \; \right\} \tag{3.6}$$

where $SEG_{(o,s,g)}$ is the g-th segment in the snake $S_{(o,s)}$ and $N_{(o,s)}$ is the total number of segments inside. Finally, each segment is interpolated by snaxels characterized by their position vectors and associated edge orientation vectors, i.e.,

$$SEG_{(o,s,g)} = \left\{ \; \left(T_{(o,s,g,a)}, O\left(T_{(o,s,g,a)}\right)\right) \mid a = 1, \cdots, N_{(o,s,g)} \; \right\} \tag{3.7}$$

where $N_{(o,s,g)}$ is the total number of snaxels in segment $SEG_{(o,s,g)}$ and $T_{(o,s,g,a)}$ is the position vector of the a-th snaxel inside. Orientation vector $O(T_{(o,s,g,a)})$ associated with snaxel $T_{(o,s,g,a)}$ is defined as

$$O\left(T_{(o,s,g,a)}\right) = \begin{cases} T_{(o,s,g,a+1)} - T_{(o,s,g,a)} & \text{if } a < N_{(o,s,g)} \\ T_{(o,s,g,a)} - T_{(o,s,g,a-1)} & \text{otherwise} \end{cases} \tag{3.8}$$

In addition, our model also includes connectivity between snakes which is given by

$$C = \left\{ \begin{matrix} C_{hh}\left(S_{(o,s_1)}, S_{(o,s_2)}\right), C_{ht}\left(S_{(o,s_1)}, S_{(o,s_2)}\right), \\ C_{th}\left(S_{(o,s_1)}, S_{(o,s_2)}\right), C_{tt}\left(S_{(o,s_1)}, S_{(o,s_2)}\right) \end{matrix} \;\middle|\; o \in N; s_1, s_2 \in N_o \right\} \tag{3.9}$$

where $C_{hh}\left(S_{(o,s_1)}, S_{(o,s_2)}\right)$, $C_{ht}\left(S_{(o,s_1)}, S_{(o,s_2)}\right)$, $C_{th}\left(S_{(o,s_1)}, S_{(o,s_2)}\right)$ and $C_{tt}\left(S_{(o,s_1)}, S_{(o,s_2)}\right)$ are the head-head, head-tail, tail-head and tail-tail connectivity between the s_1-th snake and the s_2-th snake inside the object OBJ_o with value 1 means "connected" and 0 means "unconnected". In order to simplify the symbol indexing for the subsequent formulation of energy functionals, the SDM can be alternatively defined as a set of snaxels with structure implicitly associated and is given by

$$SDM = \left\{ T_i^k \mid i = 1,\cdots,N_k; k = 1,\cdots,M \right\} \tag{3.10}$$

where T_i^k, N_k, and M correspond to the position vector of snaxel *i* in segment *k*, the total number of snaxels in segment *k*, and the total number of segments in the image respectively.

With the SDM represented formally, the dynamic behavior such as the way they preserve their structure and what kinds of structural deformation are allowed should be discussed. In this regard, the following two subsections are devoted to describe the underlying structure preservation and deformation rationale from inter-object and intra-object viewpoint respectively.

3.3.3 Inter-object Structure

In the SDM, an image is composed of a number of isolated components and each of them is called an object. In order to preserve the infrastructure of the image, we seek to preserve the relative positions of all its underlying objects. It can be achieved by incorporating pseudo-connections as a web link among objects. For two objects, pseudo-connection is defined as a link joining their centroids.

An example is depicted in Figure 3.6(a) in which there are three objects OBJ_1, OBJ_2 and OBJ_3. Their centroids are denoted by C_1, C_2 and C_3 respectively. To maintain its inter-object structure, pseudo-connections joining every pair of centroids are introduced. By making the image translational and rotational invariant, in terms of objects distribution, we chose to preserve merely the length of pseudo-connections during deformation. As a result, the rotated version shown in Figure 3.6(b) can be obtained without experiencing any penalty. To achieve this goal, an energy functional is formulated and will be presented in Section 3.4.1.

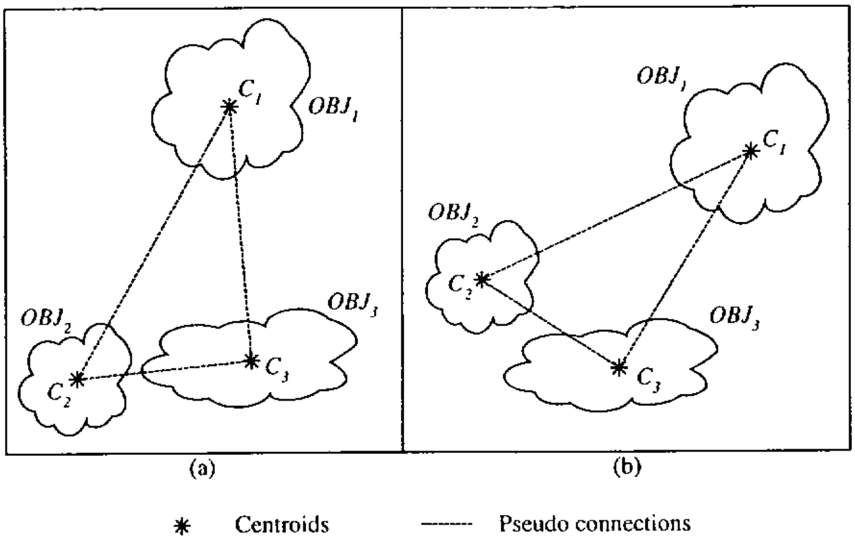


Figure 3.6 Pseudo-connections are constructed for an image consisting of three objects. (a)The original configuration. (b)The image is made rotational invariant in terms of objects distribution.

3.3.4 Intra-object Structure

Having mentioned the structure between objects, we shift our focus to that of the object itself. In the SDM, each object is modelled as a structural entity which consists of snakes, segments and snaxels. Preservation of intra-object structure requires preserving both the snaxel evenness in each segment and the snaxel edge orientation. In each segment, as long as the evenness of inter-snaxel distance and the snaxel edge orientation are maintained, we consider its structure still preserved. Considering a snake with five segments shown in Figure 3.7(a), some of the possible deformations while keeping its structure preserved are shown sequentially in Figure 3.7(b) - (f).

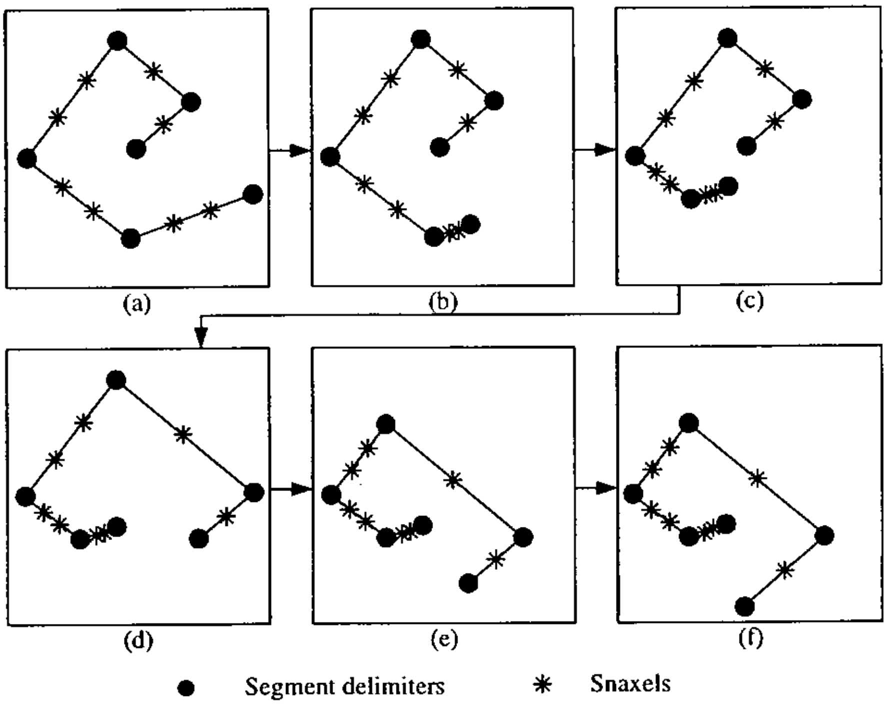


Figure 3.7 (a) A snake with five segments. (b)-(f) A sequence of its allowable deformations with structure preserved.

By representing this deformation process symbolically (with snaxel edge orientation hidden) in anti-clockwise direction, we can have:

(a) G___s___G___s___G___s___s___G___s___s___G___s___s___G

→(b) G___s___G___s___G___s___s___G___s___s___G_s_s_G

→(c) G___s___G___s___G___s___s___G__s__s__G_s_s_G

→(d) G___s___G______s______G___s___s___G__s__s__G_s_s_G

→(e) G___s___G______s______G__s__s__G__s__s__G_s_s_G

→(f) G_____s_____G_______s_______G__s__s__G__s__s__G_s_s_G

where segment-delimiter, snaxel and edge are denoted by "G", "s" and "_" respectively. In fact, syntactic approach is also using the similar technique to carry out flexible matching [1,4-5]. For example, given a class of the form ab^ncd^me, where a, b, c, d and e are primitives of the class, the allowable deformation will be:

abcde
→ abbcddddde
→ abbbbbbcdde
→ ...

This kind of "transformations" is actually a kind of structural deformation because deformation is performed on the structure of the pattern rather than on the spatial relationship among primitives. It is this structural deformation that we will apply on each object of the SDM.

The major difference between the structural deformation in syntactic approach and that in the SDM lies on the flexibility of deformation. In syntactic approach, once a class is given a specific form of structure, like ab^ncd^me, the pattern must deform accordingly without any deviation. However in SDM, because it is a kind of DM, it allows certain degree of deviation freedom. So, the following deformation is possible although the penalty of structure preservation in case 2 is greater than that in case 1:

case 1. G_s_s_G_s_G
→ case 2. G_s_s_____G_s_G

In the SDM, two internal energy functionals are required for the preservation of both the snaxel evenness in each segment and snaxel edge orientation and will be formulated in Section 3.4.2.

3.3.5 Initialization

Like the other DMs, a good initialization, i.e., a good starting point, can substantially off-load the minimization or deformation process and leads to better results. To our concern, two image patterns are given and it is required to determine whether they belong to the same class or whether they are in total match. So, initialization process serves the purpose of bringing them together as close as possible in order to facilitate the deformable matching in the next module. In this regard, an effective initialization scheme that combines scale and translation transformation has been developed. By first centering and scale-normalizing the input image to a predefined window size (the normalization window is fixed at 100x100 pixels in our experiments), the initialization of the model takes place as follows:

1. Center the model to the input image and uniformly scale it by a factor *F*, say 80%, of the normalization window. The value of *F* controls the degree that the pattern preserves its original aspect ratio. The smaller the value of *F*, the larger the allowance for flexible scaling in which the vertical and the horizontal scaling are independent to each other.

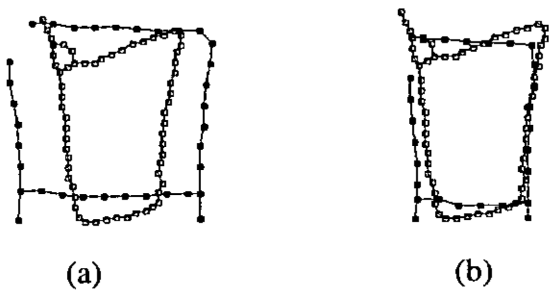
2. Translate the model by one step in each of the eight possible directions (left, top-left, etc.) and locate it to the position with the smallest evaluation energy. The evaluation energy is defined as the sum of the average distance from all pixels to their nearest snaxels and that from all snaxels to their nearest pixels. The translational step size, in terms of number of pixels, is denoted as SP_d.

3. Scale up the model horizontally in three scales and stop at a scale with the smallest evaluation energy. The horizontal scale-up step size is denoted as SP_{s1} (in terms of number of pixels)

4. Scale up the model vertically in three scales and stop at a scale with the smallest evaluation energy. The vertical scale-up step size is denoted as SP_{s2} (in terms of number of pixels)

5. Repeat step 2 to step 4 until no further modification.

This initialization scheme allows a certain degree of flexible scaling while at the same time carrying out translation transformation. The smaller the values of *F*, SP_d, SP_{s1}, and SP_{s2}, the more accurate the initialization, at the expense of computational cost. Figure 3.8 shows the difference between a simple uniform normalization and the proposed initialization scheme. Obviously, the later one provides a better starting point for the model to deform.



(a) (b)

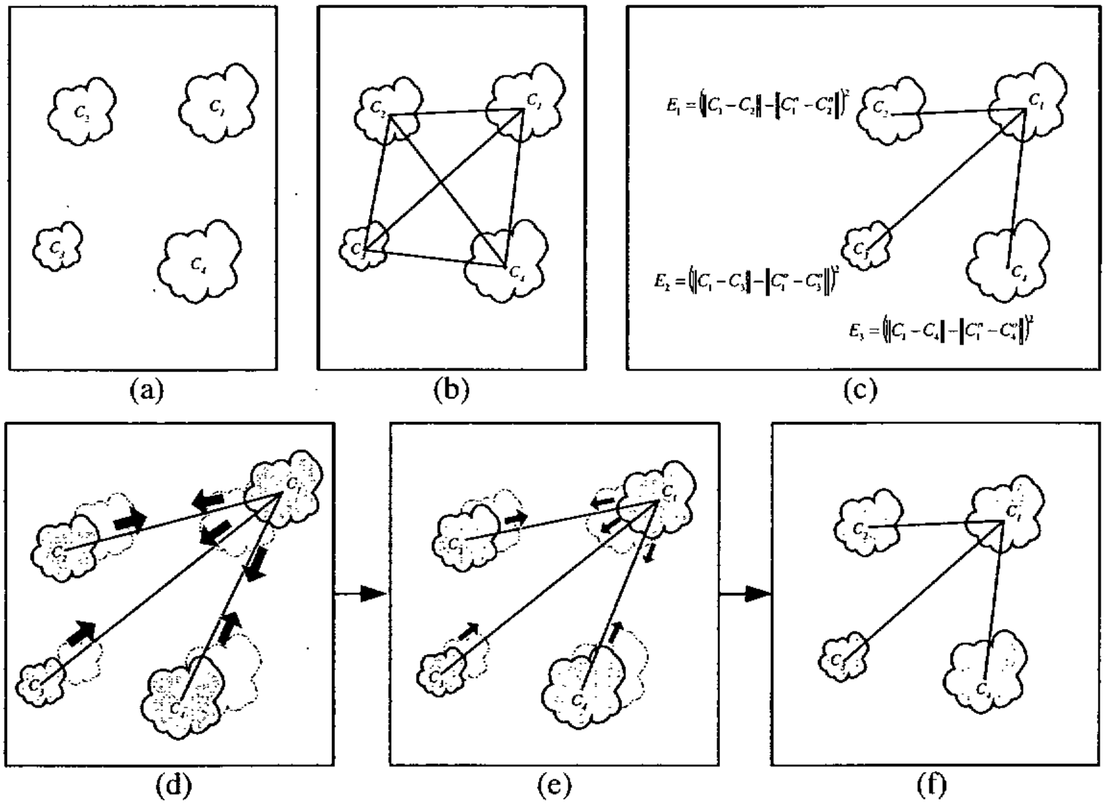
Figure 3.8 Initialization of SDM " □ " (solid square chain) to the input image " □ " (hollow square chain) by (a) uniform normalization and (b) the proposed initialization scheme.

3.4 Shape Preservation Criterion

Before proceeding to describe the deformation module, this two sections (i.e., Sections 3.4 and 3.5) are devoted to formulate the shape preservation criterion and data match criterion of the SDM respectively. For the shape preservation criterion, preserving the overall shape of the model requires preserving both its inter-object and intra-object structure. Subsections 3.4.1 & 3.4.2 below describe their formulation.

3.4.1 Inter-object Shape Preservation Criterion

As described in Section 3.3.3, pseudo-connections are employed to preserve the relative positions of all objects in an image. For every two objects, a pseudo-connection joining their centroids is constructed. For example, a web link of pseudo-connections for an image consisting of four objects (Figure 3.9(a)) is constructed as in Figure 3.9(b). To make the distribution of objects translational and rotational invariant, we do not take into considerations of the orientation of pseudo-connections. We try to preserve the inter-object structure by minimizing the length distortion of each pseudo-connection during deformation.


C_1
C_2
C_3
C_4
$E_1 = (\|C_1 - C_2\| - \|C_1^o - C_2^o\|)^2$
$E_2 = (\|C_1 - C_3\| - \|C_1^o - C_3^o\|)^2$
$E_3 = (\|C_1 - C_4\| - \|C_1^o - C_4^o\|)^2$

(a) (b) (c)

(d) (e) (f)

Figure 3.9 (a) A template consisting of four isolated objects. (b) Construction of pseudo-connections among objects. (c) Energy functionals are formulated for all connections. (d)-(f) The distorted template experiences restoration forces during energy minimization.

Taking the top-right object in Figure 3.9(b) as center, three pseudo-connections to the other objects are constructed (Figure 3.9(c)). In order to preserve the inter-object structure, the length distortion of each pseudo-connection should be minimized during deformation. It can be achieved by formulating an energy functional for each pseudo-connection to reflect the corresponding length distortion and have it minimized during deformation. In this case, an energy functional E_1 is formulated for the pseudo-connection between the object centroids C_1 and C_2, which corresponds to the squared difference between the current and the original length of the pseudo-connection. Similarly, energy functionals E_2 and E_3 are formulated for the other two pseudo-connections. Supposing that at iteration t they are moving far apart from each others as shown in Figure 3.9(d), minimizing E_1, E_2 and E_3 causes the induction of an inward force between every pair of objects which enables each pseudo-connection to restore to its original length. For the case shown in Figures 3.9(d)-(f), being caused by the inward forces, they keep moving until all the pseudo-connections are restored to their original length.

Taking the other objects as centers in turn, their energies can be generalized by

$$E = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\left\|C_i - C_j\right\| - \left\|C_i^o - C_j^o\right\|\right)^2 \qquad (3.11)$$

where

N is the total number of objects or centroids,

C_i is the current centroid position vector of the i-th object, and

C_i^o is the initial centroid position vector of the i-th object

In order to bias the influence of a close centroid, a weighting factor governed by a Gaussian window is introduced, w_{ij}. The smaller the distance between one and the centroid, the larger the value of the weighting factor and this in turn induces a greater force to preserve the relative displacement in between. The resultant internal energy functional for preserving the inter-object structure is given by

$$E_{pseudo} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left[w_{ij}\left(\left\|C_i - C_j\right\| - \left\|C_i^o - C_j^o\right\|\right)^2\right] \qquad (3.12)$$

where

$$w_{ij} = \frac{\exp\left(\dfrac{-\left\|C_i - C_j\right\|^2}{2\sigma_{pseudo}^2}\right)}{\sum_{k=1}^{N}\exp\left(\dfrac{-\left\|C_i - C_k\right\|^2}{2\sigma_{pseudo}^2}\right)} \qquad (3.13)$$

The parameter σ_{pseudo} is indeed the size of the Gaussian window which helps to define which is close centroid and which is not. One may imagine that only those centroids falling within the window (actually with a fuzzy boundary) of size σ_{pseudo} are regarded as close centroids that can provide a large preservation forces. Considering a strategy in which σ_{pseudo} is made decreasing during the course of deformation, the influence of neighbourhood centroids will become smaller and smaller and a global-to-local shape preservation is resulted. Such strategy of shape preservation is adopted by the SDM.

3.4.2 Intra-object Shape Preservation Criterion

Having defined the inter-object shape preservation criterion, we proceed to formulate energy functional(s) for intra-object shape preservation. In traditional snake model, two internal energy functionals are used to control its shape, i.e., snaxel evenness and contour smoothness. They are usually employed in contour detection problems in which these two energy functionals make the model able to align with nearby edges. However in our case, the smoothness energy functional is no longer suitable. Instead, the orientation of edges is an important indicator of whether the pattern's shape is distorted. Besides, an evenness energy functional is also required to give the model a flexibility of deformation according to snaxel evenness, i.e., a kind of structure we imposed as described in Section 3.3.4. The formulation of evenness and orientation energy functionals is presented as follows:

Evenness Preservation

In order to increase the flexibility of deformation according to the structure, the preservation of snaxel evenness in the SDM is carried out in each segment independently rather than in the whole snake like the case in traditional snake model. It enables snaxels in different segments (delimited by perceptual important points, i.e., high curvature points) to move independently.

An internal energy functional measuring the snaxel evenness distortion in segment k is defined as

$$E_k = \frac{1}{N_k - 1} \sum_{i}^{N_k - 1} \left(\left\| T_i^k - T_{i+1}^k \right\| - D_k \right)^2 \tag{3.14}$$

where

N_k is the total number of snaxels inside segment k,

T_i^k is the position vector of snaxel i in segment k, and

D_k is the average inter-snaxel distance of segment k

E_k is in fact corresponding to the average difference between the current and the average inter-snaxel distance. Generally for an image consisting of M segments, the generalized internal energy functional representing the overall snaxel evenness distortion is given by

$$E_{even} = \frac{1}{M} \sum_{k=1}^{M} \left(\frac{1}{N_k - 1} \sum_{i=1}^{N_k - 1} \left(\left\| T_i^k - T_{i+1}^k \right\| - D_k \right)^2 \right) \tag{3.15}$$

Orientation Preservation

As the edge orientation distortion is considered to be an important indicator of whether the object is distorted or not, an internal energy functional measuring the average edge orientation distortion is defined as

$$E_{orient} = \frac{1}{\sum_{k=1}^{M} (N_k - 1)} \sum_{k=1}^{M} \sum_{i=1}^{N_k - 1} \left[AD(T_i^k) \right]^2 \tag{3.16}$$

where

M is the total number of segments,

N_k is the total number of snaxels inside segment k, and

T_i^k is the position vector of snaxel i in segment k

The angle distortion $AD(T_i^k)$ is the angle difference between the current and the initial orientation of the edge associated with T_i^k. By using dot product rule, it is given by

$$AD(T_i^k) = \cos^{-1}\left[\frac{O(T_i^k)\cdot O^o(T_i^k)}{\left\|O(T_i^k)\right\|\cdot\left\|O^o(T_i^k)\right\|}\right] \tag{3.17}$$

where $O^o(T_i^k)$ is the initial position of the orientation vector $O(T_i^k)$. As our proposed system in current stage is targeted on handwritten Chinese character recognition and it is considered that most of Chinese character handwritings are not rotated, rotational invariance is not imposed. But in general, it can be easily achieved by incorporating a penalty-free affine transformation into the image frame during deformation.

Combined Preservation

As mentioned in Section 2.1.3, regularization is one of the most common ways to combine internal and external energies. However, even within the internal or external energy functional itself, there may consist of a number of sub-terms like E_{even} and E_{orient} in our case. Additional weighting factors may be required to integrate them before the final combination (of internal and external energies) takes place. In view of the difficulties to determine an optimal or sub-optimal weightings, attempts have been made to reduce the number of sub-terms in both energies. Recall that the shape preservation criterion involves inter-object and intra-object considerations. The former one is governed by energy E_{pseudo} while the latter one is governed by both E_{even} and E_{orient}. In this section, energy terms E_{even} and E_{orient} for intra-object shape preservation are integrated and the resultant energy functional is given by

$$E_{intra} = \frac{1}{\sum_k^M (N_k - 1)} \sum_k^M \sum_i^{N_k - 1} \left\| O(T_i^k) - \hat{O}^o(T_i^k) \right\|^2 \tag{3.18}$$

where

M is the total number of segments,

N_k is the total number of snaxels inside segment *k*, and

T_i^k is the position vector of snaxel *i* in segment *k*

$\hat{O}^o(T_i^k)$ denotes the vector in same direction as $O^o(T_i^k)$. If its length is fixed at a value the same as that of $O^o(T_i^k)$, minimizing E_{intra} will result in the preservation of original edge length and orientation. If its length is kept updating to reflect the up-to-date average inter-snaxel distance in its segment (i.e., segment k), minimizing E_{intra} will result in the preservation of snaxel evenness and edge orientation. Needless to say, the latter strategy will be adopted by the SDM. The details of this reformulation are attached in Appendix A.

Since the model structure is maintained merely by preserving snaxel evenness and edge orientation, it is possible that some segments may lengthen itself infinitely without any penalty introduced. So, apart from the snaxel evenness and edge orientation, the shape preservation scheme of the model should also maintain a certain degree of original edge length. It can be achieved by imposing an upper-bound and a lower-bound on the length of $\hat{O}^o(T_i^k)$ during its updates. In the proposed model, they are specified as the ratios of the length of $O^o(T_i^k)$ and are fixed at values 2.0 and 0.5 respectively in our experiments.

As the flexibility of deformation according to snaxel evenness is one of the largest gain from using the structural information of the image, its impact on the deformation is demonstrated by a real case of deformable matching as shown in Figure 3.10. Given two Chinese character patterns of the same class, "石", the system is required to deform the template towards the image to extract the dissimilarity in between. Pay attention to the top-most horizontal strokes of the image and the template that the length of the former is much longer than that of the latter. Two patterns are quite similar except for this horizontal stroke. For the matching with the flexibility to deform according to snaxel evenness (see the left column in the figure), achieved by updating the length of $\hat{O}^o(T_i^k)$ in each minimization iteration to reflect the up-to-date average inter-snaxel distance of their corresponding segments, it is observed that such horizontal stroke of the template can easily lengthen to cover the whole image stroke. For the matching without such flexibility (see the right column in the figure), achieved by fixing the length of $\hat{O}^o(T_i^k)$ equal to that of

$O^o\left(T_i^k\right)$, it is observed that although such horizontal stroke of the template has made every effort to cover the correspondence in the image, there is a little bit to the perfect coverage and it in turn gives rises to the dissimilarity measure eventually. Besides, since it doesn't have the flexibility to deform according to snaxel evenness, the resultant value of E_{intra} is found nearly double that of the former case. Hence, in this case, the incorporation of such flexibility really can enable the model to deform according to its structure without resulting in a high dissimilarity measure, like E_{intra}.

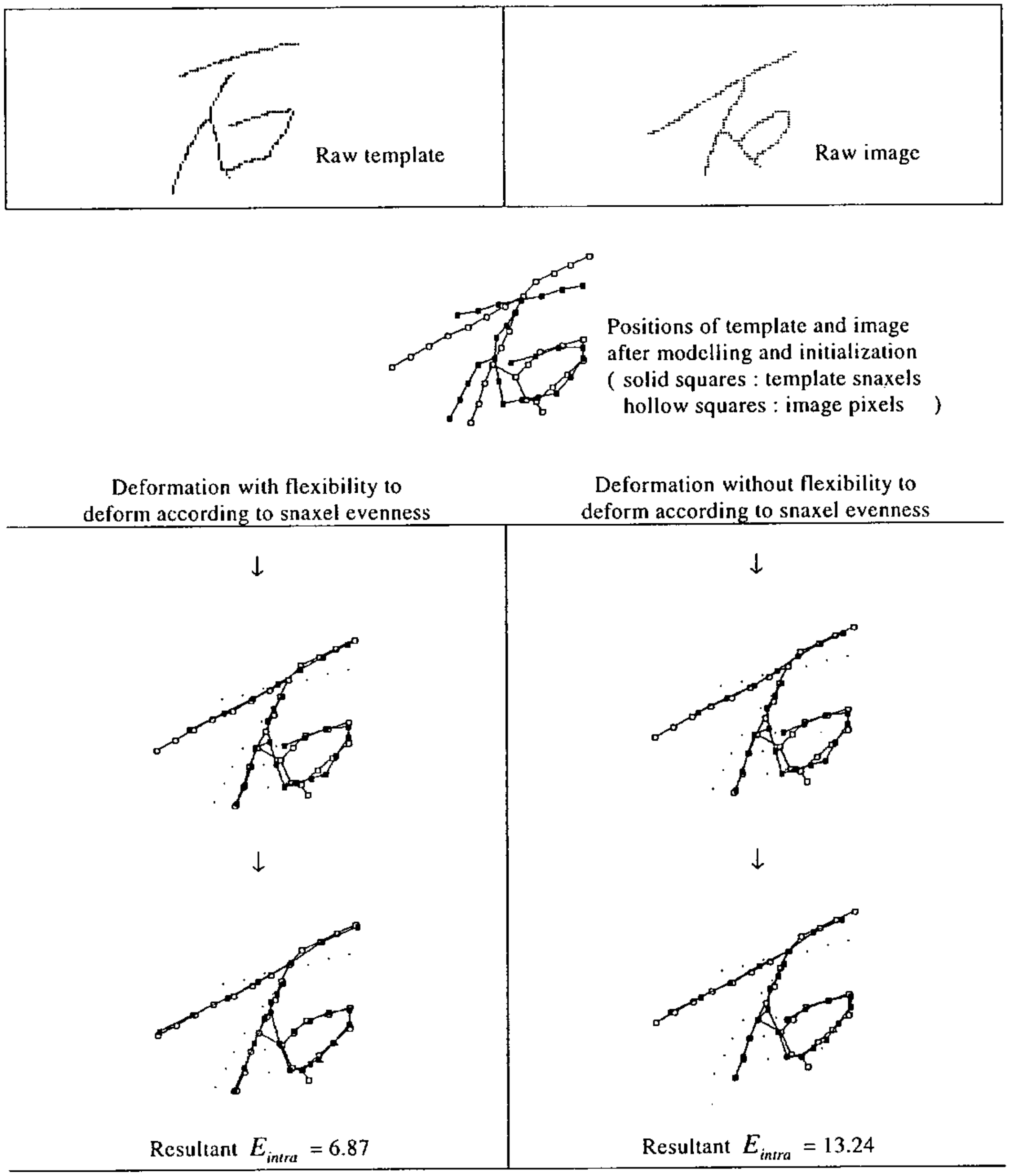

Raw template
Raw image
Positions of template and image after modelling and initialization
(solid squares : template snaxels
hollow squares : image pixels)
Deformation with flexibility to deform according to snaxel evenness
Deformation without flexibility to deform according to snaxel evenness
↓
↓
↓
↓
Resultant E_{intra} = 6.87
Resultant E_{intra} = 13.24

Figure 3.10 Demonstration of the deformable matching with (left column) and without (right column) the flexibility to deform according to snaxel evenness. Snapshots of deformation are shown in sequence from top to bottom.

3.5 Data Match Criterion

3.5.1 Point Patterns Matching and Line Patterns Matching

Before proceeding to formulate the external energy functional for SDM, let's first consider the simplest matching problem – point patterns matching. Two cases are described, namely, one point matching and two points matching. At the end, the line patterns matching, that our proposed model dealt with, is discussed.

Case 1: One Point Matching

The simplest case of point patterns matching is the one point matching. Considering a template with one snaxel T and an image with one pixel P as shown in Figure 3.11, we are required to match the template with the image by bringing the snaxel towards the pixel while keeping the pixel fixed. Tackling by energy minimization approach, its external energy functional can be formulated as

$$E_{ext1} = \|T - P\|^2 \qquad (3.19)$$

It actually corresponds to the squared distance in between. By minimizing it alone, the snaxel will experience a force towards the pixel (Figure 3.12). At subsequent iterations, the snaxel will move towards the pixel and in this case, they are overlapped finally.

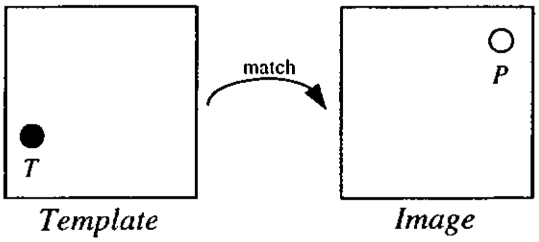


Figure 3.11 A template with one snaxel T is matched with an image with one pixel P.

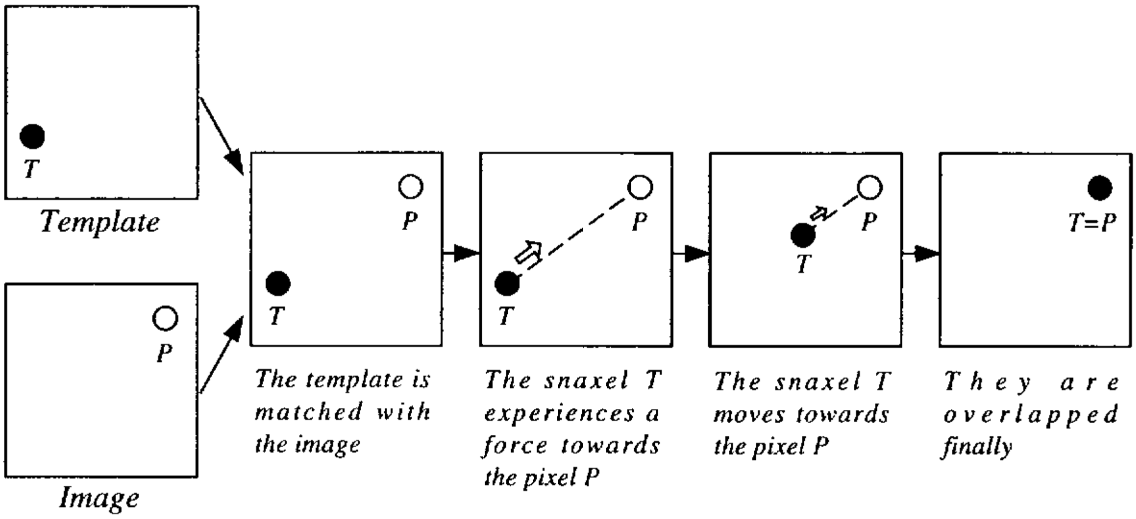

T
Template
P
Image
P
T
The template is matched with the image
P
T
The snaxel T experiences a force towards the pixel P
P
T
The snaxel T moves towards the pixel P
T=P
They are overlapped finally

Figure 3.12 The deformation process of one point matching

Case 2: Two Points Matching

The external energy functional for the case of two points matching is of a little bit more complicated than the previous one. Considering a template and an image as shown in Figure 3.13 in which each of them consists of two points, we are required to match the template with the image by bringing the snaxels towards the pixels while keeping pixels fixed. The dotted lines shown between snaxels and pixels are for demonstration purpose, which do not exist in reality.

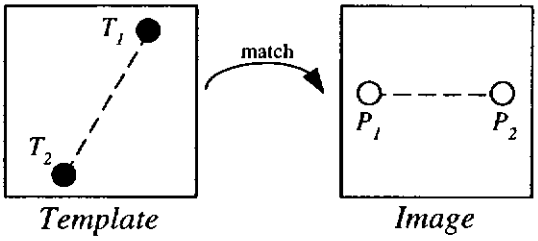


Figure 3.13 A template consisting of two snaxels (T_1 and T_2) is matched with an image consisting of two pixels (P_1 and P_2).

In this case, we do not know what pixel to which each snaxel is matched and it is assumed that we are now no need to preserve the shape of the template during deformation. In order to find a match for two snaxels, a strategy has been adopted in which each snaxel is tried to match with the nearest pixel. To implement this strategy, a fitness that the snaxel has found the nearest pixel should be formulated beforehand. Consider that the 2D image in Figure 3.13 is mapped to a 1D plane as shown in Figure 3.14. If a snaxel T is put somewhere between these two pixels, according to the above strategy, it will either move to the left towards P_1 or to the right towards P_2, depending on which pixel it is more closer to. So, a fitness graph that the snaxel has found the nearest pixel can be constructed as the one shown in the Figure 3.14 such that P_1 and P_2 are the positions where the snaxel loves to move to. The fitness equation corresponding to this graph can be formulated as

$$Gt = \frac{1}{2}\sum_{i}^{2} \mathbf{exp}\left(-\frac{\|T - P_i\|^2}{\sigma_{ext}^2} \right) \tag{3.20}$$

where σ_{ext} specifies the spread of the Gaussian window function. The overall fitness that two snaxels T_1 and T_2 have found their nearest pixels is defined as the geometric mean of their individual fitness (denoted by Gt_1 and Gt_2), that is,

$$F = \sqrt{Gt_1 \times Gt_2} \tag{3.21}$$

Finally, the external energy functional is defined as the negative log of the overall fitness and is given by

$$\begin{aligned} E_{ext2} &= -\mathbf{log}(F) \\ &= -\frac{1}{2}\sum_{i}^{2} \mathbf{log}\left[\frac{1}{2}\sum_{j}^{2} \mathbf{exp}\left(-\frac{\|T_i - P_j\|^2}{\sigma_{ext}^2} \right) \right] \end{aligned} \tag{3.22}$$

It is the smallest when all snaxels have found and matched with their nearest pixels successfully. Consider the Figure 3.13 again. By minimizing this external energy, each snaxel will try to look for the nearest pixel as shown in Figure 3.15. In this case, snaxel T_2 has found P_1 as its nearest pixel and T_1 has found P_2 as its nearest pixel. In subsequent iterations, T_1 and T_2 will move towards their nearest pixels, and eventually they are overlapped with their correspondences.

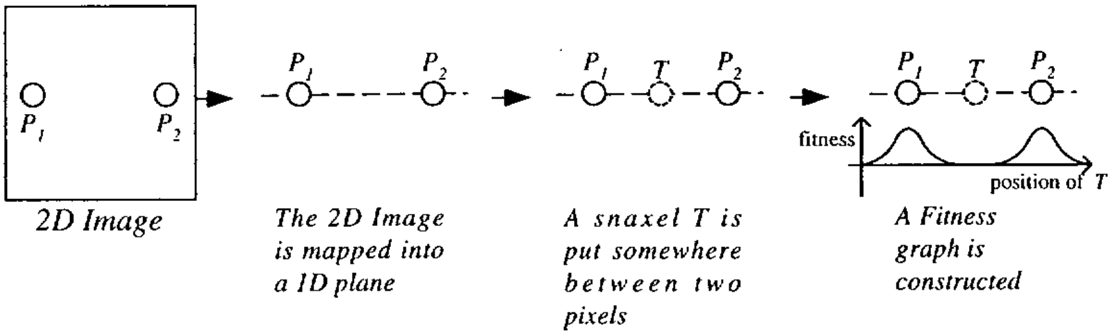

P1
P2
2D Image
P1
P2
The 2D Image is mapped into a 1D plane
P1
T
P2
A snaxel T is put somewhere between two pixels
P1
T
P2
fitness
position of T
A Fitness graph is constructed

Figure 3.14 The formulation of the fitness graph that a snaxel has found the nearest pixel

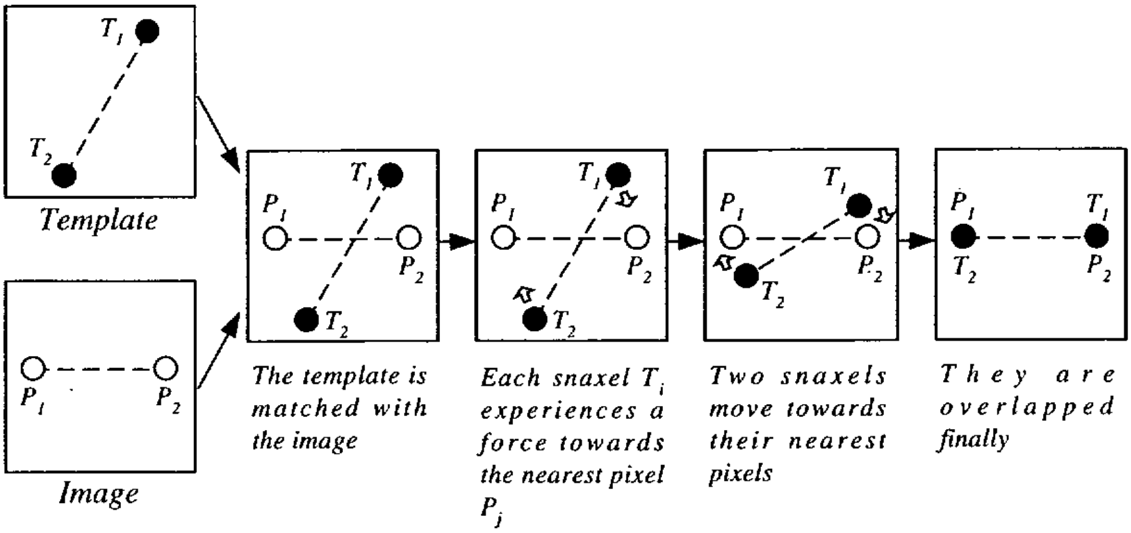

T1
T2
Template
P1
P2
Image
T1
P1
P2
T2
The template is matched with the image
T1
P1
P2
T2
Each snaxel Ti experiences a force towards the nearest pixel Pj
T1
P1
P2
T2
Two snaxels move towards their nearest pixels
P1
T1
T2
P2
They are overlapped finally

Figure 3.15 The deformation process of two points matching

Case 3: Line Patterns Matching

The final case is the line patterns matching. It is also the problem that our proposed model dealt with. Consider a template and an image shown in Figure 3.16 with each of them consisting of one curve line. We are required to match the template with the image while keeping the image fixed. In order to reduce the computational cost, both the template and the image will be interpolated by snaxels and pixels respectively. Please note that the total number of snaxels and pixels after interpolation may not equal all the times. In this case, four snaxels (T_1-T_4) and three pixels (P_1-P_3) are resulted. If using the previous points matching strategy to find a match, some undesirable image alignment are observed. According to the previous strategy, each snaxel tries to look for the nearest pixel, and in this case, snaxels T_2 and T_3 happen to have the same nearest pixel P_2. They will move towards P_2 in subsequent iterations and are eventually overlapped. It results in a severe distortion of model shape, particularly the snaxel evenness. However, for the line patterns matching, apart from points, there is an additional information – edge available for matching. So, instead of performing a point-to-point matching, a point-to-edge matching could be adopted as shown in Figure 3.16. In this case, four snaxels are not required to search for the nearest pixels but edges instead, giving them more freedom to move and finally the template aligns with the image without a severe distortion in model shape, particularly the snaxel evenness.

Compared with point-to-point matching, an uneven sampling of snaxels and pixels is possible here because each snaxel is not required to look for pixels but edges instead. Thus, some higher order of feature points can be incorporated as sampling points, like perceptual important points employed by SDM. With such sampling allowed, computational requirement could be substantially reduced. Compared with one-to-one edge matching, ours on the other hand does not require to extract an accurate one-to-one edge correspondence beforehand which is considered to be so difficult without the requirement of a sophisticated preprocessing.

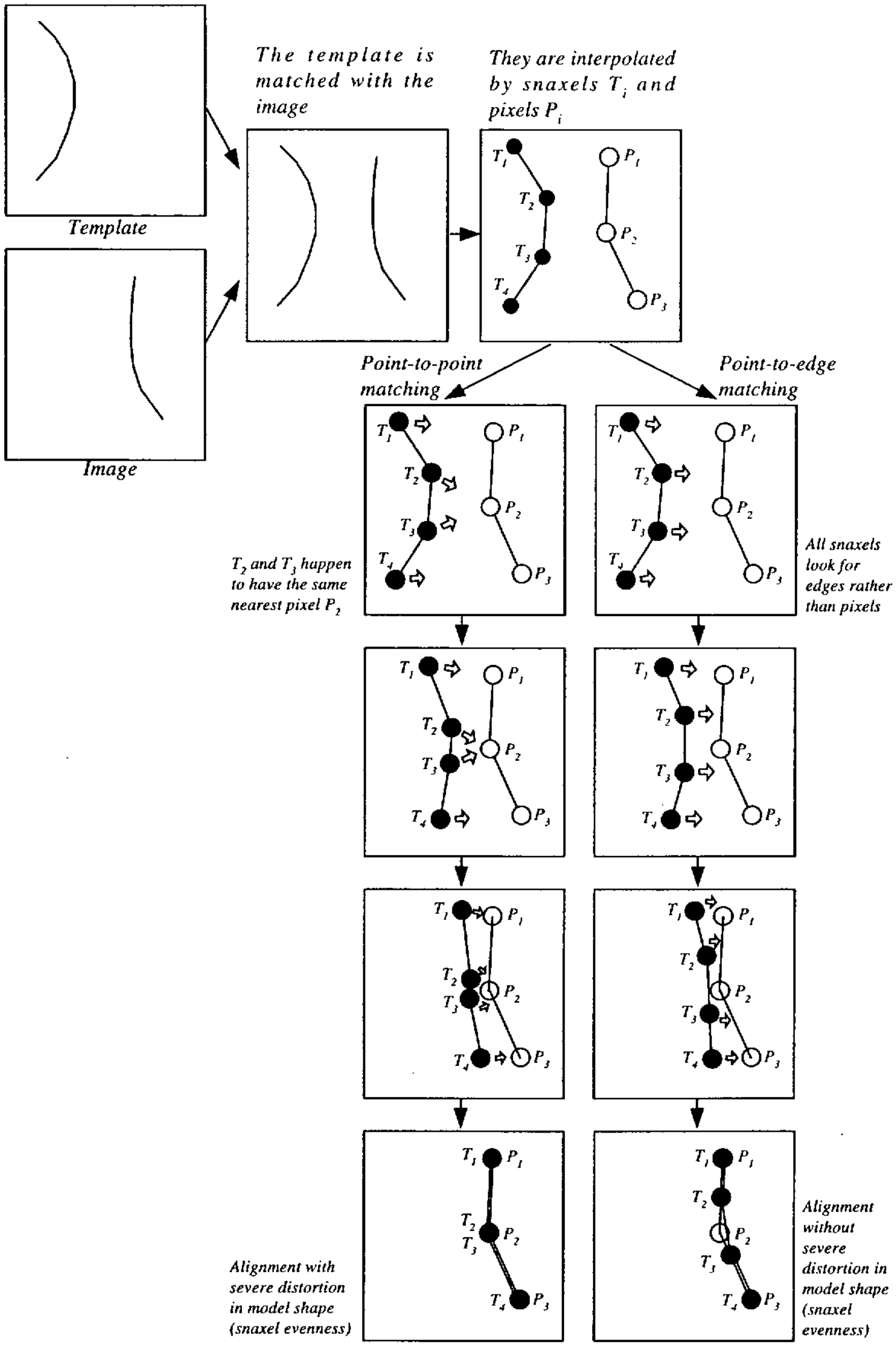

Template
Image
The template is matched with the image
They are interpolated by snaxels T_i and pixels P_i
T_1
T_2
T_3
T_4
P_1
P_2
P_3
Point-to-point matching
Point-to-edge matching
T_2 and T_3 happen to have the same nearest pixel P_2
All snaxels look for edges rather than pixels
Alignment with severe distortion in model shape (snaxel evenness)
Alignment without severe distortion in model shape (snaxel evenness)

Figure 3.16 A deformable matching of two curve lines. An unnecessary distortion of the model shape is observed in point-to-point matching but not in point-to-edge matching.

3.5.2 Point-to-edge Displacement Function

For the case 2 of point patterns matching in previous section, the term $\left\|T_i - P_j\right\|$ inside the external energy functional E_{ext2} defined in eq.(3.22) is actually corresponding to point-to-point displacement. If a point-to-edge matching is adopted, it should be replaced by a point-to-edge displacement function and the resultant external energy functional is given by

$$E_{ext3} = -\frac{1}{2}\sum_{i}^{2}\log\left[\frac{1}{2}\sum_{j}^{2}\exp\left(-\frac{h(T_i,P_j)}{\sigma_{ext}^2}\right)\right] \tag{3.23}$$

where $h(T_i,P_j)$ is the point-to-edge displacement function which measures the displacement between the snaxel T_i and an edge formed by P_j and its adjacent pixel. Generally for a template consisting of N sampled snaxels and an image consisting of M sampled pixels, the above external energy functional is generalized as

$$E'_{ext3} = -\frac{1}{N}\sum_{i}^{N}\log\left[\frac{1}{M}\sum_{j}^{M}\exp\left(-\frac{h(T_i,P_j)}{\sigma_{ext}^2}\right)\right] \tag{3.24}$$

In the SDM, $h(T_i,P_j)$ is made up of two measures, namely, positional difference and directional incompatibility as orientation is also an important cue for proper matching. Given a snaxel T and an image edge (P_1,P_2), positional difference is specified by a continuous and differentiable function $f(T,P_1,P_2)$ where

$$f(T,P_1,P_2) = \begin{cases} \left\|T-P_1\right\| & if\ \dfrac{(T-P_1)\cdot(P_1-P_2)}{\left\|P_1-P_2\right\|} \geq 0 \\ \left\|T-P_2\right\| & if\ \dfrac{(T-P_2)\cdot(P_2-P_1)}{\left\|P_2-P_1\right\|} \geq 0 \\ \dfrac{\left\|(T-P_1)\times(P_2-P_1)\right\|}{\left\|P_2-P_1\right\|} & otherwise \end{cases} \tag{3.25}$$

In fact, it is the perpendicular distance from T to the edge (P_1,P_2), but if T is not on top of the edge, it is defined as the distance to the nearest edge terminal, i.e., P_l or P_2. In Figure 3.17(a), the contour of function f with respect to an edge (P_1,P_2) is depicted. In this case, both snaxels T' and T'' lying on the same contour line share the same value of f to the edge, i.e., $f' = f''$.

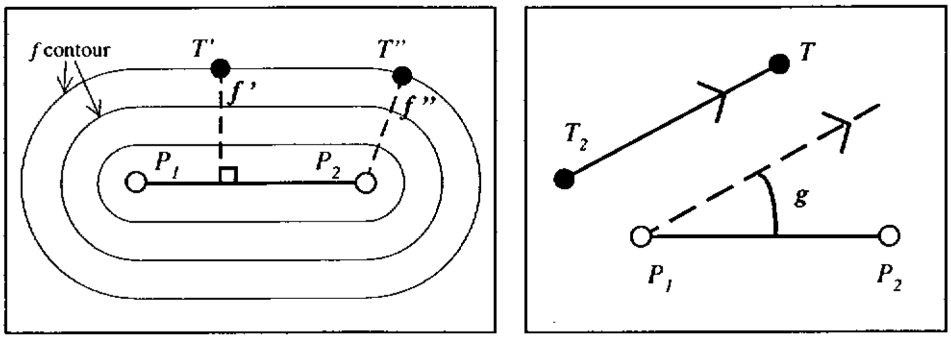
On the other hand, the directional incompatibility between a snaxel T and an edge (P_1, P_2) is defined as the angle difference between the edge formed by T and its adjacent snaxel and the edge (P_1, P_2), i.e.,

$$g(T, P_1, P_2) = \cos^{-1}\left[\frac{|O(T)\cdot(P_2 - P_1)|}{\|O(T)\|\cdot\|P_2 - P_1\|}\right] \tag{3.26}$$

where $O(T)$ is the associated orientation vector of T. It is illustrated in Figure 3.17(b) in which T_2 is the adjacent snaxel of T. The value of function g is just the angle difference between two edge vectors.

Finally, the overall point-to-edge displacement $h(T_i, P_j)$ between the snaxel T_i and the edge formed by P_j and its adjacent pixel (denoted by P_j') is simply defined as the sum of their square and is given by

$$h(T_i, P_j) = f(T_i, P_j, P_j')^2 + g(T_i, P_j, P_j')^2 \tag{3.27}$$



(a) (b)

● Snaxels ○ Image pixels

Figure 3.17 (a) The contours of the function $f(T, P_1, P_2)$. (b) Directional incompatibility between a snaxel T and an edge (P_1, P_2), i.e., $g(T, P_1, P_2)$

3.5.3 Model Guided Search and Image Guided Search

Finally, as mentioned in Section 2.3.1, the following two requirements are considered necessary for a total match:

1. Every part of the model should have some image primitives nearby, and
2. every part of the image should have some model primitives nearby.

So far, the first requirement has already been coped with but not the second one. In order to satisfy the second requirement, another part of the equation should be appended to the external energy functional E'_{ext3} just defined in eq.(3.24). It is exactly the same as the first part except that the role of snaxel and pixel are interchanged. The external energy functional is resulted to be

$$E''_{ext3} = -\frac{\alpha}{N}\sum_i^N \log\left[\frac{1}{M}\sum_j^M \exp\left(-\frac{h(T_i,P_j)}{\sigma_{ext}^2}\right)\right] - \frac{1}{M}\sum_i^M \log\left[\frac{1}{N}\sum_j^N \exp\left(-\frac{h(P_i,T_j)}{\sigma_{ext}^2}\right)\right] \tag{3.28}$$

In fact, the first part of equation enables each snaxel to search for the nearest pixel edge actively, i.e., model guided search (MGS) while the second part of equation enables each pixel to search for the nearest snaxel edge, i.e., image guided search (IGS). In other words, it integrates both the MGS and the IGS. A weighting factor α is introduced such that their weighting can be adjusted. The smaller the value of α, the less the contribution of MGS in the overall searching mechanism. In addition, the parameter σ_{ext} specifying the spread of the Gaussian window function can be interpreted as the size of a searching window such that snaxels (and pixels) will mainly look for pixel edges (and snaxel edges) falling inside. The energy functional E''_{ext3} is the resultant external energy functional adopted by the SDM and is denoted by E_{ext} in the rest of the thesis, i.e.,

$$\begin{aligned} E_{ext} &= E''_{ext3} \\ &= -\frac{\alpha}{N}\sum_i^N \log\left[\frac{1}{M}\sum_j^M \exp\left(-\frac{h(T_i,P_j)}{\sigma_{ext}^2}\right)\right] - \frac{1}{M}\sum_i^M \log\left[\frac{1}{N}\sum_j^N \exp\left(-\frac{h(P_i,T_j)}{\sigma_{ext}^2}\right)\right] \end{aligned} \tag{3.29}$$

Chapter 4

BAYESIAN FORMULATION, GLOBAL-TO-LOCAL DEFORMATION AND CLASSIFICATION

Having introduced the model representation and the deformation criteria in previous chapter, we proceed to formulate the deformation process of SDM under a Bayesian framework. Furthermore, the incorporation of a global-to-local deformation ability into the SDM as realized by a smoothing scheme will be described. At the end of the chapter, a classification scheme based on SDM will be elaborated.

4.1 Bayesian Formulation and Objective Function

Like the other DMs, the process of deformation (or deformable matching) of the SDM can be formulated under a Bayesian framework. Using Bayes rule, the prior probability of the deformed template $P(M)$ and the likelihood of the input image given the deformed template $P(I \mid M)$ can be combined to obtain the a posteriori probability density of the deformed template given the input image $P(M \mid I)$. Mathematically,

$$P(M \mid I)=\frac{P(I \mid M)P(M)}{P(I)} \tag{4.1}$$

As a result, deformation is realized by maximizing the a posteriori probability density. When the local or global maxima (if possible) is reached, the template is considered close to the image while being able to keep its shape in minimum distortion, and the deformation will be stopped. The formulation of prior distribution, likelihood and posterior probability density, followed by the minimization scheme adopted by SDM, are given below:

Prior Distribution

To bias the possible shape of the deformed template such that the template with no distortion in terms of inter-object structure and intra-object structure is the most favoured, the prior probability of the deformed template is defined as

$$P(M) = \beta \cdot \mathbf{exp}\left[-\frac{\left(E_{pseudo} + E_{intra}\right)}{\sigma_1} \right] \tag{4.2}$$

where β is the normalizing constant. In fact, $P(M)$ under the Bayesian scheme reflects the prior knowledge of the template shape.

Likelihood

The likelihood specifies the probability of observing an input image given a deformed template. Using the external energy defined in Section 3.5, the likelihood is defined as

$$P(I \mid M) = \gamma \cdot \mathbf{exp}\left(-\frac{E_{ext}}{\sigma_2} \right) \tag{4.3}$$

where γ is the normalizing constant. The maximum likelihood is achieved when it satisfies both requirements of the total match, i.e., each image pixel has got a snaxel edge nearby and each snaxel has got a pixel edge nearby.

Posterior Probability Density

Based upon Bayes rule, the prior probability of the deformed template in eq.(4.2) and the likelihood of the input image given the deformed template in eq.(4.3) can be combined to obtain the a posteriori probability density of the deformed template given the input image, i.e.,

$$\begin{aligned} P(M \mid I) &= \frac{P(I \mid M) \cdot P(M)}{P(I)} \\ &= C1 \cdot P(I \mid M) \cdot P(M) \\ &= C2 \cdot \mathbf{exp}\left(-\frac{E_{ext}}{\sigma_2}\right) \cdot \mathbf{exp}\left(-\frac{(E_{pseudo} + E_{intra})}{\sigma_1}\right) \end{aligned} \tag{4.4}$$

Taking natural logarithms on both sides results in:

$$\begin{aligned} \mathbf{log}(P(M \mid I)) &= \mathbf{log}(C2) - \frac{E_{ext}}{\sigma_2} - \frac{(E_{pseudo} + E_{intra})}{\sigma_1} \\ &= C3 - \frac{1}{\sigma_2}\left[E_{ext} + \frac{\sigma_2}{\sigma_1}(E_{pseudo} + E_{intra})\right] \\ &= C3 - \frac{1}{\sigma_2}\left[E_{ext} + \lambda(E_{pseudo} + E_{intra})\right] \end{aligned} \tag{4.5}$$

where

$$\lambda = \frac{\sigma_2}{\sigma_1}, \tag{4.6}$$

C1, *C2* and *C3* are constants.

Our objective is to maximize the a posteriori probability or, equivalently, minimize the following Bayesian objective function with respect to all snaxels positions:

$$\Psi = E_{ext} + \lambda(E_{pseudo} + E_{intra}) \tag{4.7}$$

where λ can be regarded as a relative weighting between the total internal energy $(E_{pseudo} + E_{intra})$ and the external energy (E_{ext}).

Minimization

With Ψ defined, the template can be deformed towards the image by minimizing this Bayesian objective function. Due to its simplicity, the steepest descent method is adopted. In view of its success in neural network training, a momentum term is included as well. The updating equation for each snaxel T_i at t-th iteration is given by

$$T_i(t+1) = T_i(t) + \Delta T_i(t) \tag{4.8}$$

where

$$\Delta T_i(t) = -\eta \frac{\partial(\Psi)}{\partial T_i(t)} + \zeta \cdot \Delta T_i(t-1) \tag{4.9}$$

In fact, $\Delta T_i(t)$ corresponds to the displacement vector of snaxel T_i. η and ζ denote the learning rate and momentum respectively. In each iteration, displacement vectors $\Delta T_i(t)$ of all snaxels are computed, aiming at decreasing Ψ. Such minimization process continues until Ψ cannot be further decreased for a certain period of time, i.e., the minimization process is considered converged. However, it is observed that if snaxels move merely according to these set of displacement vectors, the flexibility of deformation will be too large that the model may start to deviate significantly from its original shape and search locally once the deformation starts (A case will be given later). We found that it is essential to guide the model to deform globally at the very beginnings so as to escape from some local minima as introduced by the improper initialization. Deforming locally means that some model primitives do not follow the global movement of the model like the case shown in Figure 4.1(a) in which the snaxel T_2 tends to search/deform locally by moving backwards while all the other snaxels (T_1, T_3, T_4 and T_5 which may or may not be connected) tend to move forwards. We try to develop a scheme called smoothing in the following section which enables them to move globally (like the one shown in Figure 4.1(b)) initially and locally at the later stage of deformation.

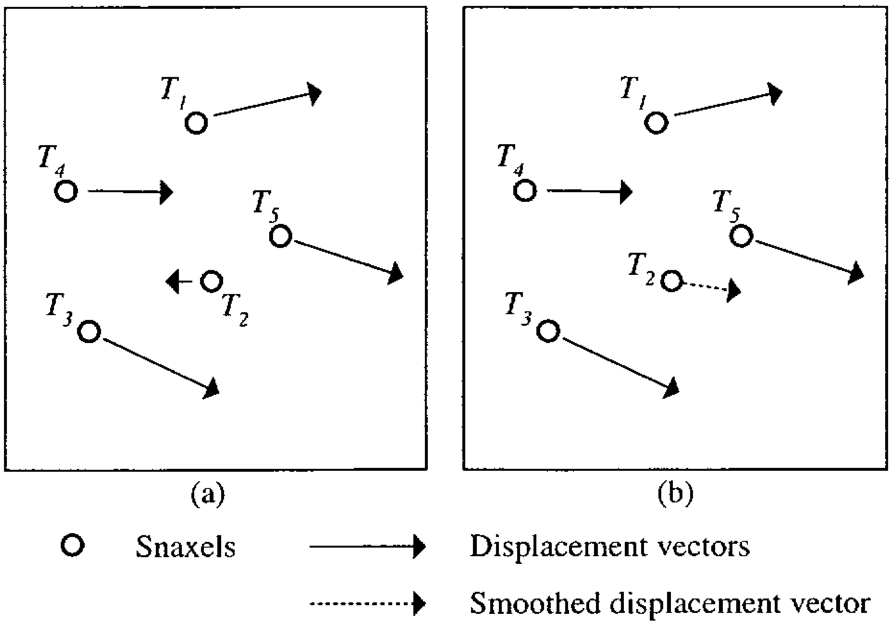


Figure 4.1 A template consisting of five snaxels which may or may not be connected. (a) Before a smoothing process is applied. (b) After a smoothing process is applied (particularly on snaxel T_2)

4.2 Smoothing Scheme

The smoothing scheme presented here aims at incorporating a global-to-local deformation ability into the SDM. The rationale behind is from the concept of ASP, described in Section 2.3.3, that it makes use of some schemes to preserve model shape actively. For examples, Burr's model [17] and Wakahara's model [11] are the ones who are based on ASP.

The smoothing scheme adopted by SDM mainly follows the idea in Burr's model [17] that the movement of model primitives (refer to snaxels here) is restricted according to the spatial relationship between primitives. The resultant flowchart of the deformation module is depicted in Figure 4.2. After each iteration of minimization, a set of displacement vectors, i.e., $\Delta T_i(t)$, for all snaxels is computed (according to eq.(4.9)). The smoothing process is introduced here to convert this set of displacement vectors to a set of smoothed displacement vectors (denoted here as

$\Delta^s T_i(t)$) before the actual movement of snaxels taking place. $\Delta^s T_i(t)$ is defined as the weighted average of neighbouring displacement vectors. To provide the effect that nearby neighbours influence more than far away ones, a Gaussian weighting function is employed. Finally, all snaxels are updated according to $\Delta^s T_i(t)$ before next minimization iteration starts. Refer to Figure 4.1 again. Actually, the displacement vectors and the smoothed displacement vectors are corresponding to solid arrows and dotted arrows respectively here. In this case, the smoothing process is applied to the displacement vector of T_2, making the snaxel to follow the global movement of the other snaxels as shown in Figure 4.1(b).

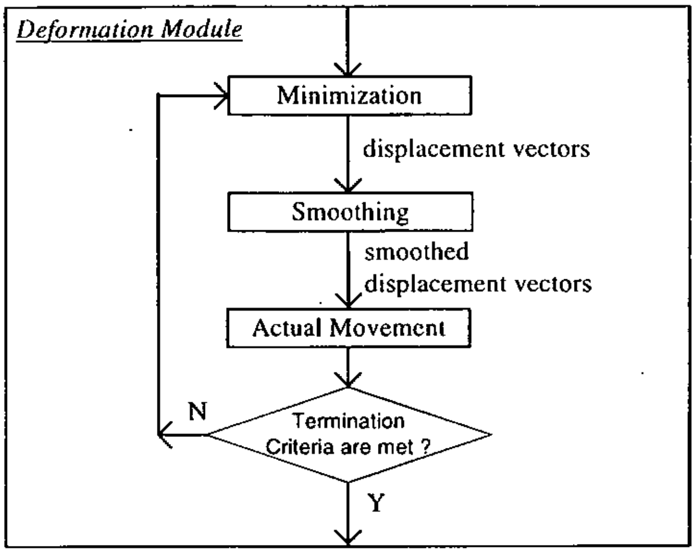


Figure 4.2 The flowchart of deformation module

Here we start to formulate the smoothing process mathematically. Based on the displacement vectors given by eq.(4.9), the smoothed displacement vector of snaxel T_i at t-th iteration is defined as

$$\Delta^s T_i(t) = \frac{\sum_j \left[\, W(T_i, T_j) \cdot \Delta T_j(t) \,\right]}{\sum_j W(T_i, T_j)} \tag{4.10}$$

where the Gaussian weighting function $W(T_i,T_j)$ is given by

$$W(T_i,T_j)=\exp\left[-\frac{\left\|T_i-T_j\right\|^2}{\sigma_{smooth}^2}\right] \tag{4.11}$$

The parameter σ_{smooth} specifies the size of the window in which snaxels will provide significant neighbourhood smoothing forces to the current snaxel T_i. It is this parameter which governs the global-to-local deformation of the SDM. The larger the value of σ_{smooth}, the more global the deformation. If σ_{smooth} is equal to zero, a pure local deformation is resulted. So, a global-to-local deformation can be achieved by starting σ_{smooth} at a higher value and reducing its value gradually during the course of deformation.

In view of the hierarchical structure of SDM that each separated component inside an image is treated as an object and the preservation of inter-object structure has already been coped with by pseudo connections, neighbourhood of influence here is further modified to be restricted to the same objects. Thus, the smoothed displacement vector in eq.(4.10) is fine-tuned as

$$\Delta^s T_i(t)=\frac{\sum_{j\in obj(i)}\left[\ W(T_i,T_j)\cdot\Delta T_j(t)\ \right]}{\sum_{j\in obj(i)}W(T_i,T_j)} \tag{4.12}$$

of which the implementation is much more easier and efficient. With the smoothing process introduced, each snaxel will be updated according to its smoothed displacement vector, i.e.,

$$T_i(t+1)=T_i(t)+\Delta^s T_i(t) \tag{4.13}$$

To demonstrate the impact of adopting the smoothing process in reality, a real case of deformable matching is shown in Figure 4.3. Given two Chinese character patterns of the same class, "四", the system is required to deform the template towards the image to extract the dissimilarity features in between. Supposing in this case that the initialization process fails to bring them together resulting in a far apart distance in between, the burden is shifted to the deformation process that the model needs to move a long way and tries its best to escape from many local minima in order to align with the image. For the case without the incorporation of smoothing (see the right column in the figure), the template starts to search locally and deviates much from its original shape in just a few iterations of deformation. Although in this case it is so lucky to align with the image properly, there are cases that the template corrupted eventually. On the contrary, if smoothing is incorporated (see the left column in the figure), the deformation of the template is properly guided such that its global shape can be well maintained during the deformation process. In the proposed model, such kind of deformation is required so that the model can deform globally at the very beginnings so as to escape from some local minima as introduced by the improper initialization, like this case.

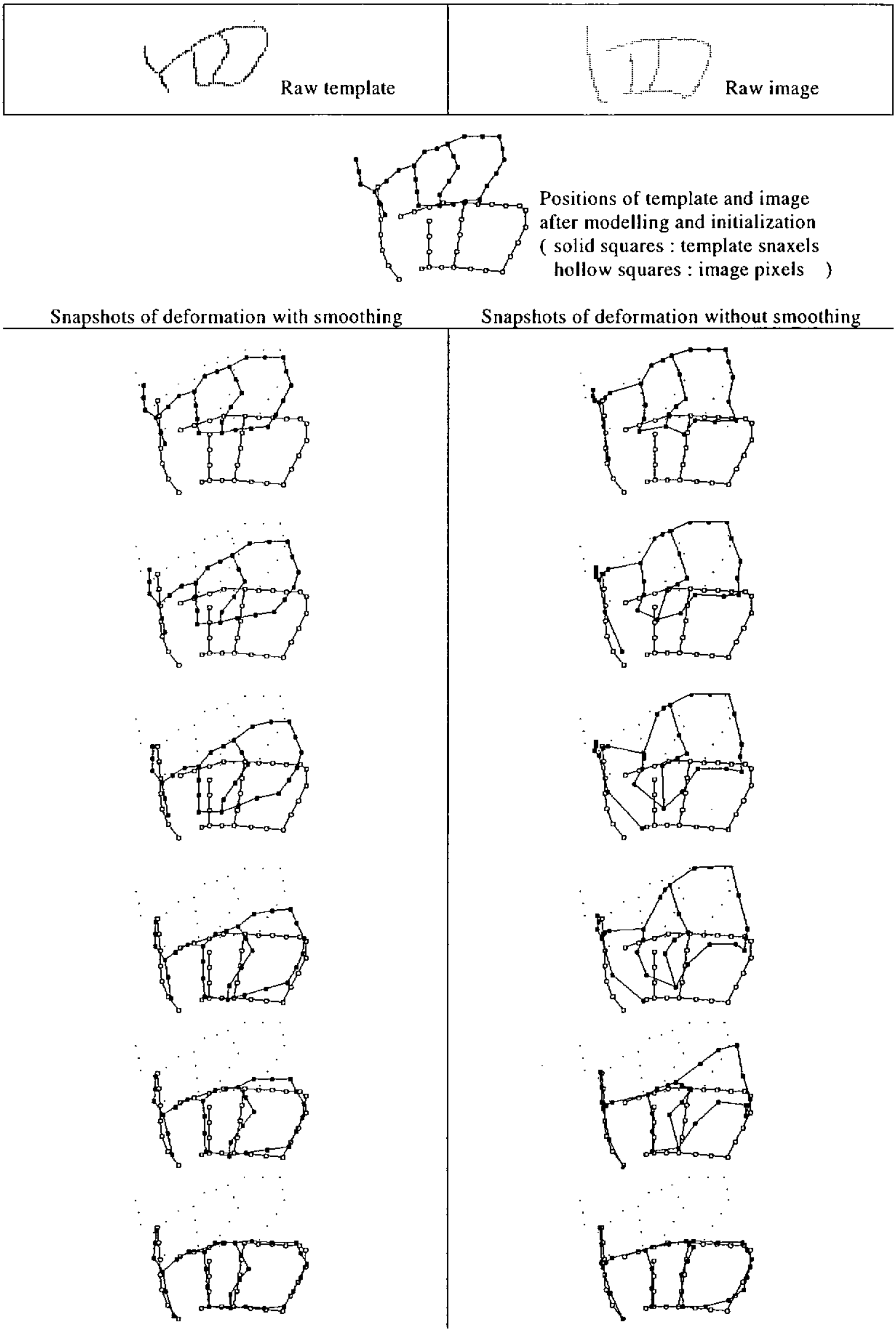

Raw template
Raw image
Positions of template and image
after modelling and initialization
(solid squares : template snaxels
hollow squares : image pixels)
Snapshots of deformation with smoothing
Snapshots of deformation without smoothing

Figure 4.3 Demonstration of the deformable matching with smoothing(left column) and without smoothing(right column). Sequence is from top to bottom.

4.3 Feature Extraction and Classification

The most straightforward way to carry out classification using SDM/DMs is to compare the objective function values Ψ (after the minimization process) for the pattern classes of concern and pick the one with the lowest value. It does not require any sophisticated feature extraction step and in fact an important merit of using DMs for classification tasks. In fact, one may view differently that the resultant objective function value is the most obvious feature being extracted by a deformable matching. In other words, SDM can be considered as a feature extraction method which includes the resultant objective function value as the most representative feature. Such a view is adopted, elaborated by introducing other features and described below.

4.3.1 Feature Extraction

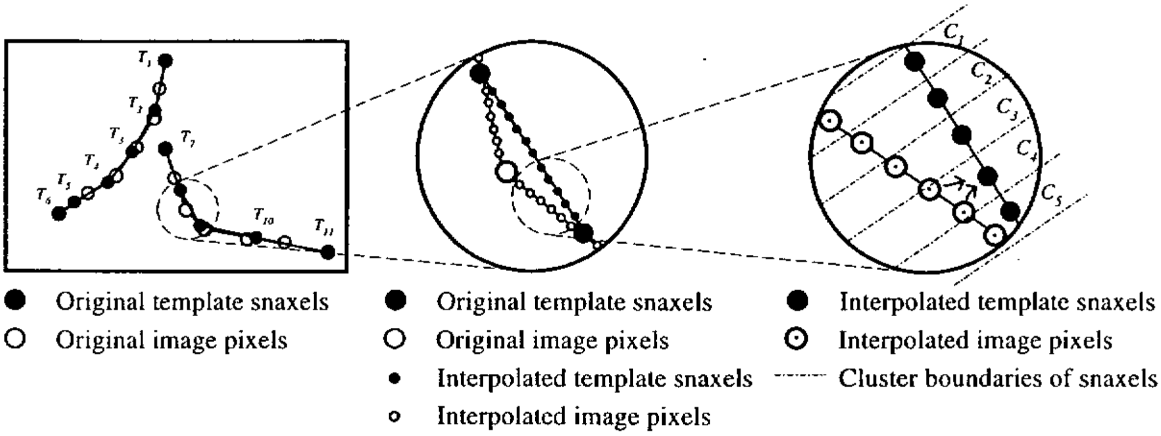
Although the classification can be carried out simply by using the resultant objective function value, it is found insufficient because it doesn't tell us explicitly how good the template aligns with the image which is considered to be an important indicator of whether two images are in good match. In this regard, an additional feature measuring the alignment fitness is extracted – clustering error. It is in fact a counter counting the total number of unmatched pixels and snaxels after deformation. To sum up, in the proposed model, two features are extracted for classification:

1. resultant objective function value, and
2. clustering error.

The objective function values Ψ in eq.(4.7) can be obtained easily after the deformation process. The clustering error on the other hand is described below. Although a cleaner formulation can be obtained by incorporating the clustering error into the criteria used by the SDM, it is found difficult to do so because some of the dissimilarity measure are considered too complex to be formulated as a general function which is continuous and differentiable and can in turn be minimized. Even, some of them are really processes rather than functions, like the calculation of clustering error which is in fact a process that investigates on the clustering statistics and can only be performed after the deformation process.

Similarity measurement between two binary images (image and template) can be regarded as a clustering problem. Both image pixels and snaxels can be treated as clusters which group snaxels and pixels respectively. If both parties can obtain an even clustering, i.e., the number of clustered points is the same for each pixel/snaxel, high similarity exists between both images. In the proposed model where point-to-edge matching is adopted, snaxels and image pixels have been sampled to reduce the computational cost. To carry out clustering in a more detail manner, snaxels and image pixels should be added back by interpolation between adjacent sampled snaxels and pixels. The clustering process remains the same except that the total number of clusters is increased by those interpolated snaxels and pixels. As long as the interpolation rate of both parties are the same, the requirements of the total match mentioned in Section 2.3.1 can be redefined such that all clusters are required to have one and only one clustered point for the total match. In this regard, a clustering error is considered if either of the following cases happens:

1. some cluster(s) have more than one clustered points, and
2. some cluster(s) are empty.



T_1
T_2
T_3
T_4
T_5
T_6
T_7
T_{10}
T_{11}
C_1
C_2
C_3
C_4
C_5
● Original template snaxels
○ Original image pixels
● Original template snaxels
○ Original image pixels
• Interpolated template snaxels
∘ Interpolated image pixels
● Interpolated template snaxels
⊙ Interpolated image pixels
------ Cluster boundaries of snaxels

Figure 4.4 An undesirable clustering error

However, by using the SDM for pattern matching, some undesirable clustering errors are observed, like the one shown in Figure 4.4. The left-most picture is the final deformation snapshot extracted from Figure 3.2. Visually, two strokes of the template are mapped properly into those in the image. Since it is a point-to-edge mapping, it is possible that one snaxel edge may be mapped to two pixel edges, like the case highlighted in this picture. In this case, they are considered to be properly aligned (at least in the case of one-to-two edge mapping) and there is no severe distortion in model shape, i.e., the evenness of snaxels and orientation of edges. However, after an even interpolation of snaxels and pixels, it is found that some clusters have more than one clustered points (snaxels are treated as clustering party). In this case, among clusters C_1-C_5, the cluster C_4 has two clustered elements. Hence a clustering error occurs according to its definition stated in above. In order to reduce such kind of undesirable clustering error if strokes have been considered to be properly mapped, an increase in the interpolation rate for the clustering party is proposed. Nevertheless, such proposal will result in a number of clusters with no elements (i.e., the second condition of clustering error) even if strokes have been considered to be properly aligned, making it difficult to differentiate the case when the image is found to be the sub-part of the template. In fact, the second condition of the clustering error can be neglected by treating both snaxels and pixels as clusters in turn. By treating pixels as clusters, if the image is really a sub-part of the template, then there should exist some pixel clusters with more than one elements, i.e., the first condition of clustering error. To summarize, the clustering error can be calculated by the following steps:

1. Treating snaxels as the clustering party.

2. Interpolating snaxels and pixels among the existing ones. Interpolation rate of the clustering party (i.e., snaxels) should be set higher than (say, double) that of the other.

3. Clustering all the pixels and then counting the total number of pixels in each cluster.

4. If the total number of pixels in a particular cluster is greater than 1, then the clustering error associated with that cluster is equal to the total number of pixels in that cluster minus 1, or else it is equal to zero.

5. Summing up all the clustering error of all clusters and a total clustering error by treating snaxels as the clustering party is resulted.

6. Treating pixels as the clustering party and repeating steps 2-5 with the role of snaxel and pixel interchanged, and finally another total clustering error is resulted. The overall clustering error is simply defined as their sum.

Mathematically, the clustering error of the SDM is given by

$$E_{clustering} = \sum_{i}^{M} \Gamma_i \tag{4.14}$$

where

$$\Gamma_i = \begin{cases} N(C_i) - 1 & if\ N(C_i) > 1 \\ 0 & otherwise \end{cases} \tag{4.15}$$

The function $N(C_i)$ measures the total number of clustered elements in the i-th cluster, i.e., C_i, which can be either a snaxel cluster or a pixel cluster (including those sampled and interpolated ones). Γ_i denotes the clustering error in each cluster while M denotes the total number of snaxel and pixel clusters, and finally the overall clustering error $E_{clustering}$ is formulated as the summation of all the underlying individuals.

4.3.2 Classification Scheme

The classification algorithm adopted by SDM is simply a statistical one, i.e., a Bayes classifier. The overall classification scheme of the SDM is depicted in Figure 4.5. It consists of two stages, feature extraction stage and classification stage. In the feature extraction stage, an unknown image is required to match with all N class templates and a 2-D feature vector is resulted after each. Let $Y_i = (y_{i1}, y_{i2})$ be the 2-D feature vector resulted after the deformable matching with template *i*. In the classification stage, all feature vectors $(Y_1,\ldots,Y_N)$ will be fed to the Bayes classifier in which the discriminant function for class *i* is given by (let *x* be the unknown image)

$$d_i(x)=p(x\mid\omega_i) \tag{4.16}$$

where

$$p(x\mid\omega_i)=(2\pi)^{-1}\left|\Sigma_i\right|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right] \tag{4.17}$$

Note here that we have assumed the probabilities of the occurrence of all pattern classes, i.e., $P(\omega_i)$, are the same. In fact, the term $(x-\mu_i)$ is actually corresponding to Y_i previously extracted, which reflects the distance between the image and the cluster center in feature space. By substituting Y_i into the term $(x-\mu_i)$ inside $p(x\mid\omega_i)$, $d_i(x)$ can be determined and the image will be classified to the class with the largest $d_i(x)$.

For each $d_i(x)$, the covariance matrix Σ_i should be determined before classification taking place. In other words, a training process is required to construct the density distributions of all pattern classes. Suppose in class *i*, there are N_i training samples and one of them is chosen as the representative template. By deforming it with the other training samples, (N_i-1) feature vectors will be resulted and let them be $(Z_1,\ldots,Z_{N_i-1})$ where $Z_i=(z_{i1},z_{i2})$. Assuming that (z_{i1},z_{i2}) are stochastically independent , Σ_i can be calculated by

$$\Sigma_i=\frac{1}{N_i-1}\sum_{j=1}^{N_i-1}\left[(Z_j)\cdot(Z_j)^T\right] \tag{4.18}$$

with covariance elements, i.e., σ_{jk} for all $j\neq k$ set to zero. In our implementation, template selection is performed by a cross-over deformation among different training samples within each class. The template with a density distribution which maximize the probability of observing all the other training samples within the same class is selected.

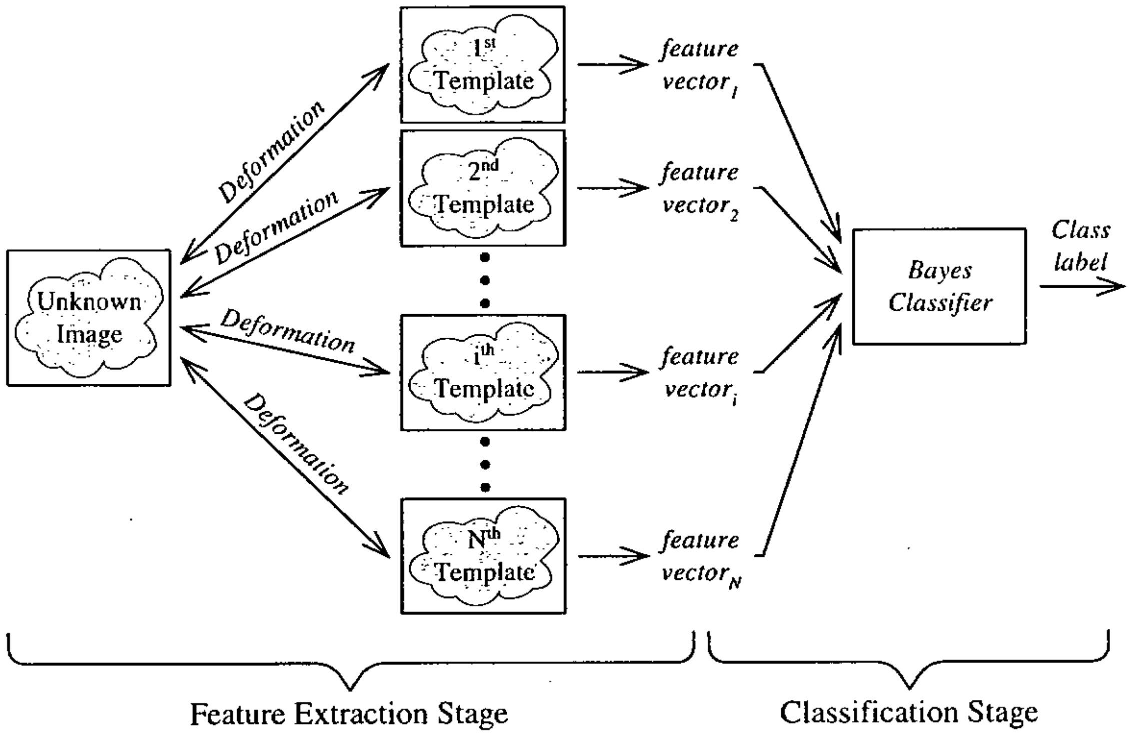


Figure 4.5 A pictorial view of the classification scheme using SDM

Chapter 5

EXPERIMENTAL RESULTS

5.1 Character Image Database

In this chapter, the performance of the proposed model is examined through various experiments. All the training and testing samples employed are extracted from a public database "IPTP CD-ROM2" of handwritten KANJI character images [32]. Inside the database, all the samples were digitized from actually used postcards with 400-dpi and 8-bit resolution, and have already been segmented into characters and binarized. The typical recognition rate on the database with first choice is 78.6 %[33].

Like the other DMs, SDM also has the same problem – a great computational cost. So, instead of being proposed as an alternative recognizer, the SDM is mainly proposed as a post-processor of other recognition systems, i.e., it helps to verify the candidates they produced. Investigation will be carried out on the performance of the SDM by simulating it as

1. a post-processor of a system adopting structural approach, and
2. a post-processor of a system adopting statistical approach.

All training and testing samples are coming from 50 character categories as shown in Table 5.1. Ten training samples and ten testing samples are extracted exclusively for each category. For each row (Gi) in the table, there are ten different character categories which are specially selected from the database with high similarity in structure. These five groups of characters, i.e., G1-G5, are simulated as the candidates given by a structural based system since it is expected that recognition using structural approach will result in characters with high similarity in structure. Another five groups of character categories corresponding to the five columns of characters in Table 5.1, i.e., V1-V5, are simulated as the candidates produced by those systems adopting statistical approach. Since the statistical based recognizer classifies characters based on a statistical framework, it does not guarantee to produce candidates with similar structure. However, as statistical approach may also find some character candidates in similar structure, one adjacent character category in each row is added to the corresponding candidate set and finally, the groups V1-V5 are formed. For each group, whether it is Gi or Vi, a one-out-of-ten classification is performed.

Before proceeding to the performance analysis on the database in detail, a parameter sensitivity analysis, a feature extraction analysis as well as a functionality analysis are carried out aiming at looking for the optimum parameters for subsequent experiments and investigating the performance of different functional components and mechanism being adopted in the model.

Table 5.1 Character categories for performance analysis

	V1		V2		V3		V4		V5	
G1	田	早	由	白	甲	四	面	日	口	石
G2	目	里	月	見	有	真	直	井	百	甘
G3	大	木	天	六	太	人	文	八	入	久
G4	十	千	中	子	丁	小	七	上	土	下
G5	王	玉	生	二	五	立	平	牛	三	手

5.2 Parameter Sensitivity Analysis

A sensitivity analysis of the parameters adopted by the proposed model is reported in this section. Before the initialization process taking place, all images are pre-normalized to size of 100x100 pixels. The interpolation intervals of all snaxels and pixels are fixed at 1/10 of the image normalization size. Initially, the following parameter values are chosen: $\eta = 0.3$, $\zeta = 0.7$, $\lambda = 1.75$, $\alpha = 1$, $\sigma_{ext} = 0.7$, σ_{smooth} and σ_{pseudo} are decreased from 1/3 of the image normalization size by 0.3 each iteration. The parameter sensitivity analysis is carried out by observing the performance of the SDM against each parameter. All the recognition rate reported here is the average accuracy of recognizing images in groups G1-G5 and V1-V5 with first choice. Parameters being investigated are summarized in Table 5.2 and are described in detail as follows:

Table 5.2 Parameters to be examined

Parameters	Meaning	Refer to equation(s)
λ	Weighting between internal and external energies	(4.7)
α	Weighting between two searching strategies, MGS and IGS	(3.29)
σ_{ext}	The size of searching window for snaxels and pixels	(3.29)
σ_{smooth} and σ_{pseudo}	Neighbourhood of influence	(4.11) and (3.13)

<u>*Weighting between internal and external energies*</u> *(λ)*

The first parameter to be analyzed is the relative weighting λ between internal and external energies in eq.(4.7). Values $\{0.5, 1, 1.75, 2.5, 3.25\}$ are examined and the corresponding performance of SDM is plotted in Figure 5.1. It is observed that the SDM is not quite sensitive to the values of λ but the values between 1 and 2.5 are the preferred setting to obtain an acceptable performance .

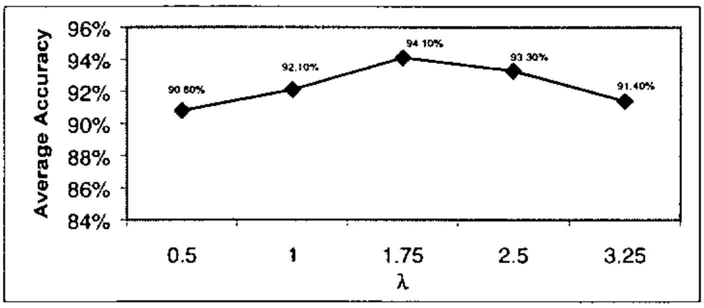


Figure 5.1 The performance of SDM against λ

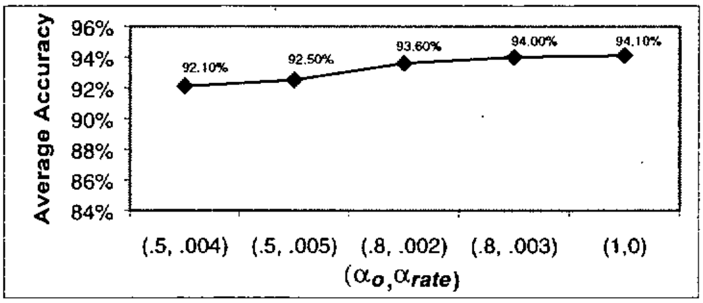


Figure 5.2 The performance of SDM against $(\alpha_o, \alpha_{rate})$

Weighting between MGS and IGS (α)

The second parameter to be examined is the relative weighting α between two searching strategies, MGS and IGS in eq.(3.29). How the weighting of MGS and IGS influences the performance of SDM is investigated by using the following updating equation for α :

$$\alpha(t) = \min((\alpha_o + t \cdot \alpha_{rate}), 1) \tag{5.1}$$

where t is an index of deformation iterations, α_o and α_{rate} are the initial value and the increasing rate of α respectively. In fact, a heuristic is adopted in which the template pattern is required to be distributed evenly on the image before it actively searches for the image. Five sets of value $(\alpha_o, \alpha_{rate})$, i.e.,

$$\{(0.5, 0.004), (0.5, 0.005), (0.8, 0.002), (0.8, 0.003), (1, 0)\}$$

ranging from a slow active search to a fast active search are examined. The corresponding performance of the SDM is plotted in Figure 5.2. It is observed that the SDM is not sensitive to how α is set and what strategy is used. So, the simplest way is to fix α at 1 during deformation.

Searching window for snaxels and pixels (σ_{ext})

Next, the parameter σ_{ext} specifying the size of searching window for snaxels and pixels in eq.(3.29) is examined. Values $\{0.3, 0.5, 0.7, 1, 1.3\}$ were examined and the corresponding performance of the proposed model is plotted in Figure 5.3. It is found that the model is not performing quite good when σ_{ext} is small because the searching windows are so small that most of snaxels and pixels are not able to locate their nearest neighbours and consequently, the template cannot align well with the image. It is suggested to set σ_{ext} within the range from 0.5 to 1 for an acceptable performance.

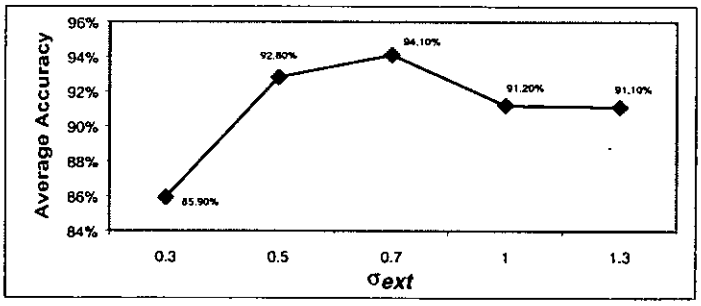


Figure 5.3 The performance of SDM against σ_{ext}

Neighbourhood of influence (σ_{smooth} *and* σ_{pseudo})

Inside the smoothing process and the inter-object structure preservation criterion, their neighbourhood of influence are governed by σ_{smooth} in eq.(4.11) and σ_{pseudo} in eq.(3.13) respectively. For simplicity, they share the same value and are here denoted by σ_{sp}. In order to carry out a global-to-local deformation during the course of matching, σ_{sp} is updated by the following equation at each iteration of energy minimization:

$$\sigma_{sp}(t) = \mathbf{max}\left(\left(\frac{1}{3}(\text{Normalization Size}) - t \cdot \sigma_{spRate}\right), 0\right) \tag{5.2}$$

where t is an index of deformation iterations and σ_{spRate} is the decreasing rate of σ_{sp}. Thus, the neighbourhood of influence at the very beginning, i.e., global deformation stage, is one-third of the image normalization size. Values $\{0.3, 0.5, 0.7, 1, 1.3\}$ ranging from a small to a large decreasing rate are examined. The first one will make σ_{sp} drop to zero in about 110 iterations while the last one in about 25 iterations. The corresponding performance of the SDM is plotted in Figure 5.4 and it is found that the proposed model is not sensitive to how σ_{sp} is adjusted.

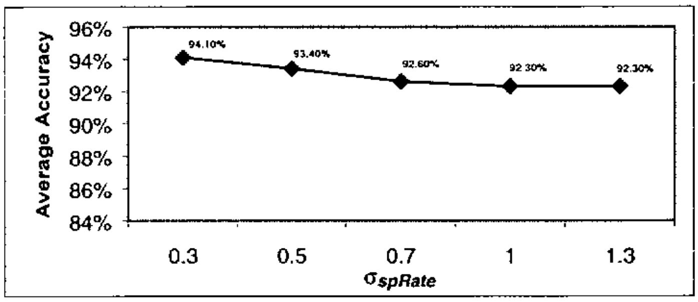


Figure 5.4 The performance of SDM against σ_{spRate}

Based on the results, the optimum values of parameters are { $\lambda = 1.75$, $\alpha_o = 1$, $\alpha_{rate} = 0$, $\sigma_{ext} = 0.7$ and $\sigma_{spRate} = 0.3$ } which will be adopted by the SDM in subsequent analysis.

5.3 Feature Extraction Analysis

In the SDM, two features are extracted, namely, the resultant value of objective function and the clustering error. In fact, the former one is the core feature indicating the dissimilarity between two images while the latter one is an assistant feature which provides one more information (alignment fitness) to the classification. By combining these two features, a 2D feature classification is carried out. In this section, the performance of the SDM with and without the assistant feature is investigated, that is, 2D and 1D feature classification. It is summarized in Table 5.3 and is plotted in Figure 5.5. It is observed that the performance of the SDM increases when an additional feature, i.e., the alignment fitness, is introduced. In other words, the extraction of clustering error as an additional feature can actually improve the model performance.

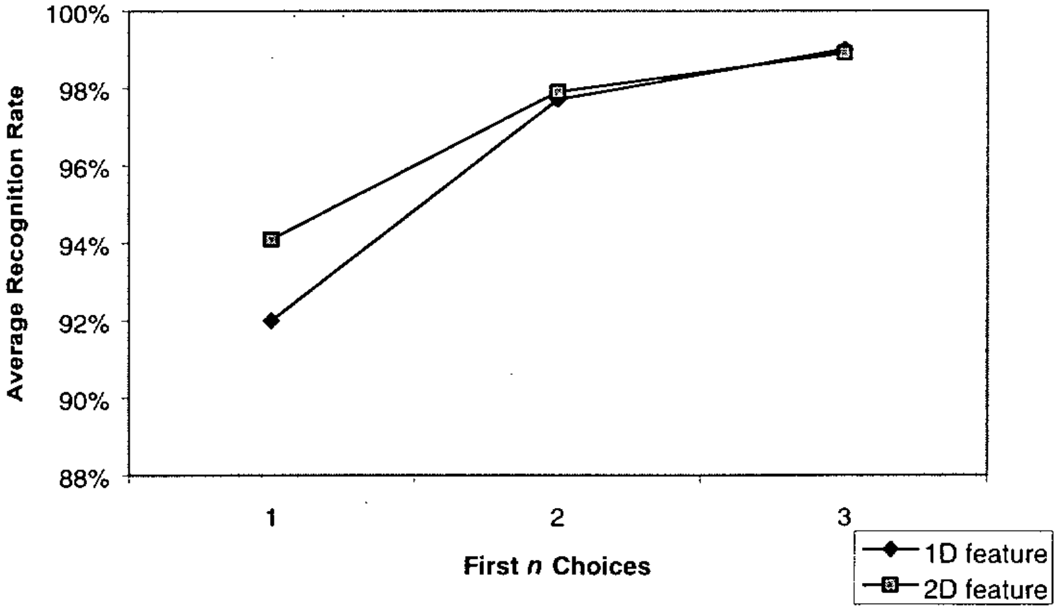


Figure 5.5 The performance of SDM with one and two features extracted

Table 5.3 Details on the recognition rate of SDM with one and two features extracted

	First choice		First 2 choices		First 3 choices	
	1D	2D	1D	2D	1D	2D
Post-processing of structural based system (avg of G1-G5)	88.8 %	92.2 %	96.8 %	97.6 %	98.6 %	98.8 %
Post-processing of statistical based system (avg of V1-V5)	95.2 %	96.0 %	98.6 %	98.2 %	99.4 %	99.0 %
Average recognition rate	92.0 %	94.1 %	97.7 %	97.9 %	99.0 %	98.9 %

5.4 Functionality Analysis

Apart from the parameter sensitivity analysis and feature extraction analysis, a functionality analysis is carried out as well to examine the impact of various components like structural deformation and global-to-local deformation ability on the SDM. In the SDM, the structural deformation is realized by inter-object and intra-object structure preservation while the global-to-local deformation ability is realized by a smoothing process. It is interesting to know whether they actually improve the performance of a DM and hence a set of experiments was conducted to find out their corresponding impact. For comparison purpose, the performance of the SDM without structural deformation and global-to-local deformation ability is also examined. It is achieved by having the model to preserve merely the original edge length and the orientation of each snaxel edge. The flexibility of deformation according to snaxel evenness is turned off. The resultant performance of the SDM is summarized in Table 5.4 and is plotted in Figure 5.6.

It is observed that the performance of the proposed model increases with each functionality introduced, in particular for the components, inter-object preservation and smoothing process, they improve the performance significantly. As a result, it is reasonable to conclude that the incorporation of structural deformation and global-to-local deformation abilities into a DM is justifiable and worthwhile.

Table 5.4 Details on the recognition rate of SDM in functionality analysis

Structural Deformation		Global-to-local Deformation	Recognition Rate		
Intra-object structural preservation (evenness)	Inter-object structural preservation (pseudo connection)	Smoothing scheme	Post-processing of structural based systems (average of G1-G5) First choice	Post-processing of statistical based systems (average of V1-V5) First choice	Average accuracy
✗	✗	✗	89.6 %	94.8 %	92.2 %
✓	✗	✗	89.2 %	95.4 %	92.3 %
✓	✓	✗	90.6 %	95.6 %	93.1 %
✓	✓	✓	92.2 %	96.0 %	94.1 %

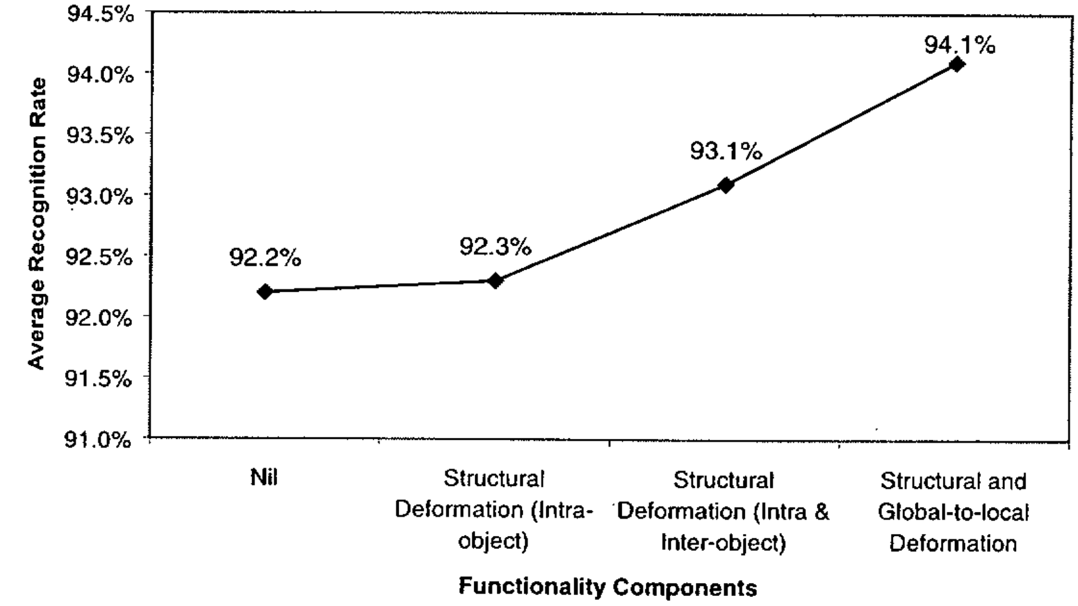


Figure 5.6 The performance of the SDM with different functional components

5.5 Performance Analysis of Chinese Character Recognition

Having reported the parameter sensitivity analysis as well as the impact of different functional components and mechanism on the proposed model, a detail analysis of the model performance on Chinese character recognition is presented in this section. Apart from simulating the SDM as a post-processor of a structural recognizer and a post-processor of a statistical recognizer, an additional set of experiment has been conducted as well to investigate the performance of SDM by treating it as a stand-alone recognizer. A 1-out-of-50 stand-alone classification by utilizing all 50 character categories in Table 5.1 is carried out. All the characters will be recognized without any rejection. In order to compare with a DM, the rubber sheet model proposed by Jain and Zongker [10] has been implemented for performance comparison. The results are presented accordingly. In the rest of this chapter, the rubber sheet model is denoted by "RSM".

5.5.1 Recognition as Post-processing of Structural Based Systems

Figure 5.7 shows the recognition rate by treating the SDM and the RSM as post-processors of a structural based recognizer. All data points shown were recorded as an average of five readings obtained from performing recognition in five groups G1-G5. Table 5.5 shows the details of their performance. It is observed that our model outperforms the RSM and has obtained a 92.2 % accuracy for first choice.

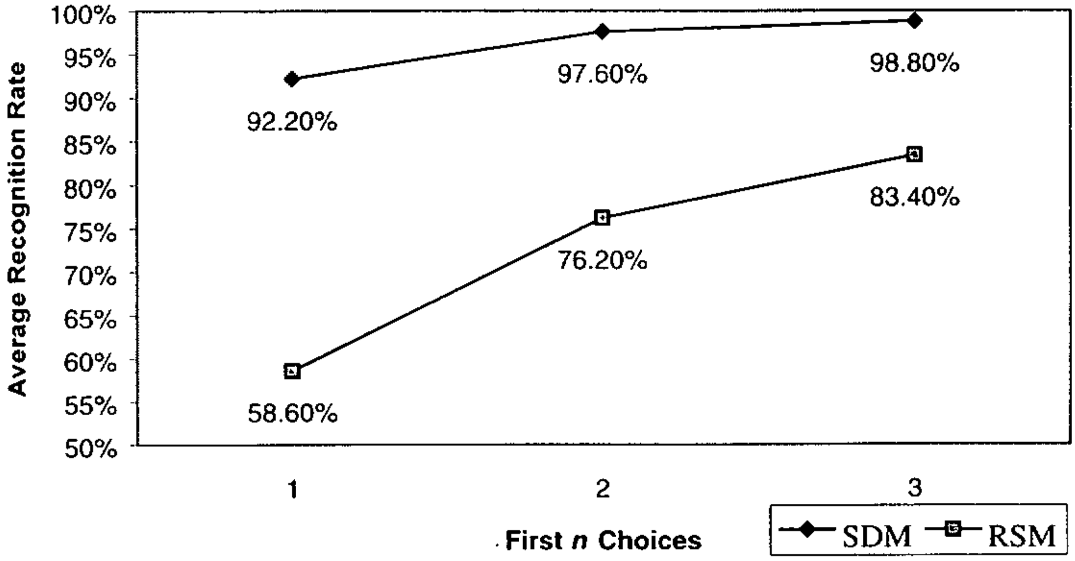


Figure 5.7 The performance comparison of the SDM and RSM : post-processing of structural based systems

Table 5.5 Details on the recognition rate of the SDM and RSM : post-processing of structural based systems

Groups	First choice		First 2 choices		First 3 choices	
	SDM	RSM	SDM	RSM	SDM	RSM
G1	89 %	64 %	96 %	77 %	98 %	85 %
G2	93 %	60 %	100 %	79 %	100 %	85 %
G3	91 %	33 %	96 %	53 %	98 %	63 %
G4	92 %	71 %	97 %	86 %	99 %	91 %
G5	96 %	65 %	99 %	86 %	99 %	93 %
Average	92.20 %	58.60 %	97.60 %	76.20 %	98.80 %	83.40 %

In order to demonstrate the performance of our model visually, five cases of correct classification extracted from each group are shown in Figure 5.8 – 5.12 respectively. For each figure, a particular image is classified and the deformation of the resultant top three choices of template candidates are shown. In Figure 5.8, an image is fed to the system. As the proposed model adopted a deformable matching approach rather than a statistical one, the first few choices of template candidates should exhibit a certain degree of structure similarity with the image, which is reflected by this example. Also in this case, the image will not be wrongly classified as and because penalty is induced for their existence of extra snaxel strokes as compared with the input image. It is reflected in Table 5.6 that their final external energy and the clustering error are much higher than that of the winning case.

The second case of correct classification is shown in Figure 5.9 and Table 5.7. Like the previous one, three template patterns with similar structure as that of the image are resulted. Again, the second and the third choices of template candidates cannot be the winner due to their existence of extra snaxel edges. It leads to a high clustering error obtained eventually (see $E_{clustering}$ in Table 5.7). These extra snaxel edges also tried to align with some pixel edges by altering its internal shape. It is reflected by their high internal energies E_{int} (i.e., $E_{pseudo} + E_{intra}$) as compared with that of the winner class. However, due to the need of shape preservation, some snaxel edges may not be able to situate on any pixel edges and hence, their external energies are also greater than that of the winner one.

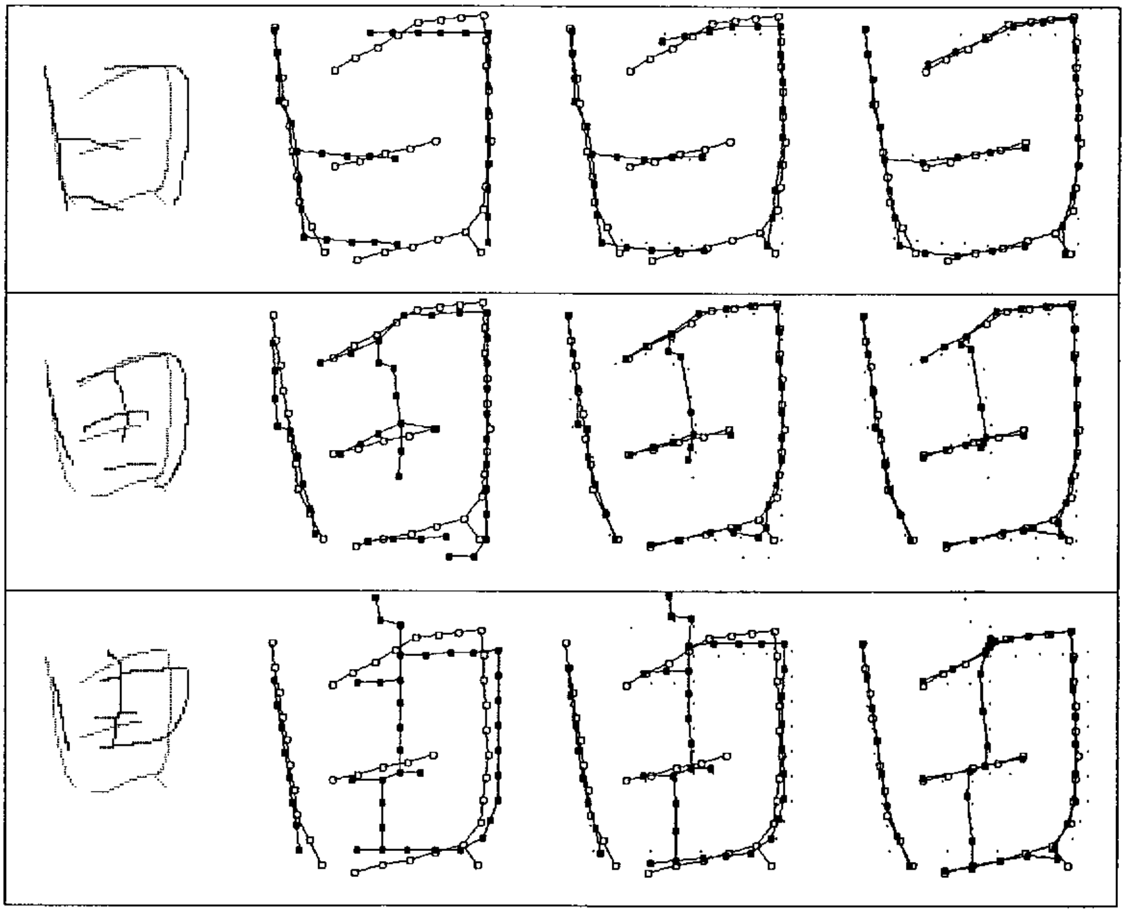


Figure 5.8 These three rows correspond to the deformation of top three choices of template candidates, 日 , 田 and 由 (in dark colour) respectively when feeding an image 日 (in light colour). From left to right, they are the original position of two patterns, their configurations after initialization (solid and hollow squares are snaxels and pixels respectively), a snapshot during deformation and the resultant configuration after deformation. Note that the original position of each snaxel is shown by a small dot.

Table 5.6 Energy values after deformation in Figure 5.8

	$E_{int}\left(E_{pseudo}+E_{intra}\right)$	E_{ext}	Ψ	$E_{clustering}$
日 - 日 matching	6.48	11.32	22.66	13
日 - 田 matching	5.05	24.01	32.83	38
日 - 由 matching	6.89	34.40	46.45	55

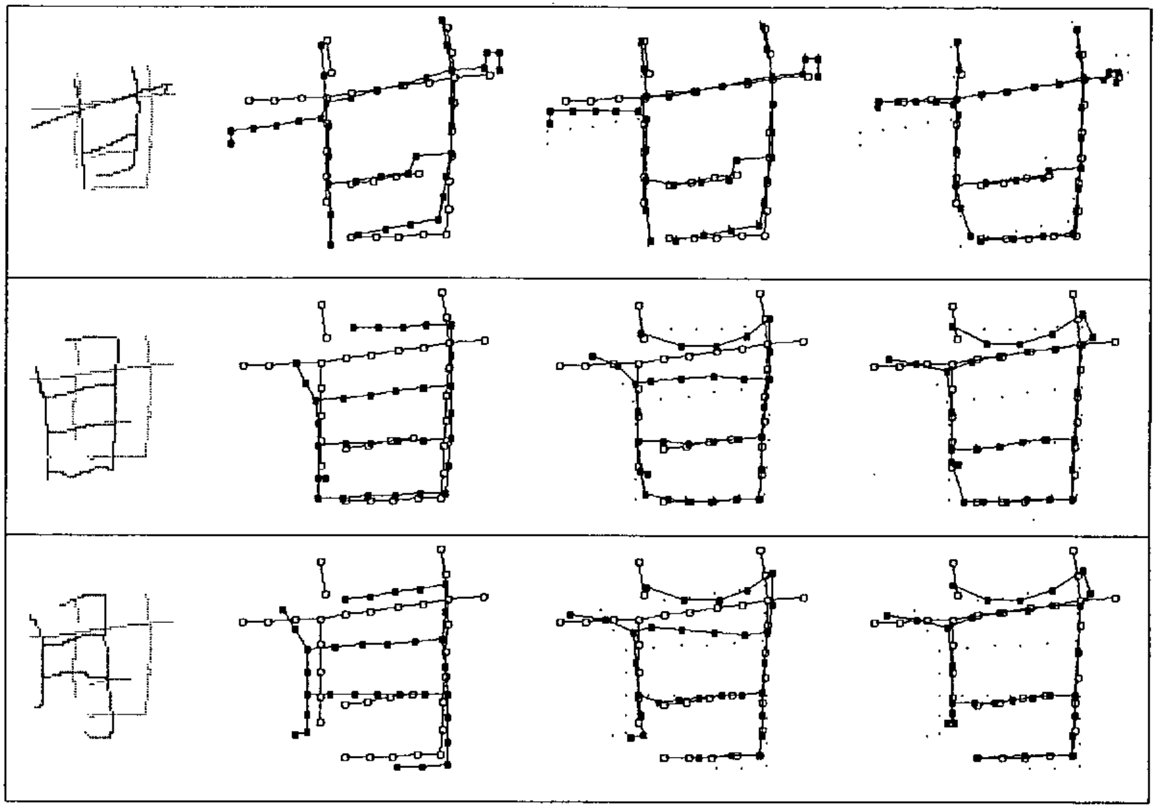


Figure 5.9 The deformation of top three choices of template candidates, 甘, 目 and 月 when feeding an image 甘.

Table 5.7 Energy values after deformation in Figure 5.9

	$E_{int}\left(E_{pseudo}+E_{intra}\right)$	E_{ext}	Ψ	$E_{clustering}$
甘-甘 matching	1.94	9.63	13.03	31
甘-目 matching	12.73	17.88	40.16	57
甘-月 matching	11.99	16.78	37.77	54

Another example of correction classification is shown in Table 5.8 and Figure 5.10 in which the image 太 has found three template patterns (太, 六 and 木) with highest discriminant function in the proposed classification scheme. In this case, 木 cannot be the winner because its extra vertical stroke in the middle causes it to have a high internal and external energy and also a high clustering error. For the template 六, our structural deformation scheme is demonstrated by its ability to shorten its longest horizontal stroke and to lengthen its bottom two strokes to match with the image without a much increase in the internal energy E_{int} (i.e., $E_{pseudo} + E_{intra}$). However, as it does not have an extra structure component to cater for the short edge at the bottom of the image (reflected by a high external energy and clustering error), it fails to be the winner.

Next, the correct classification of an image 十 has been shown in Figure 5.11 and Table 5.9. Consider the case when the template 千 is deformed to match with 十. Under the influence of our structural deformation scheme, the stroke at the head of 千 has an ability to shorten itself as long as an evenness of snaxels is maintained without a much increase in the internal energy. However, in this case, such behaviour is undesirable because all the concerning snaxels are overlapped together. Although it cannot be reflected by the internal energy as well as the external energy, it is tackled by the clustering error measure which indicates the alignment fitness. As reflected in Table 5.9, its clustering error is much higher than that of the winning case, that is, a bad alignment is found. Consider the case when the template 丁 is deformed to match with 十. Although the proposed scheme of the structural deformation gives it an ability to shorten its tail, it can by no means to cater for the head of 十 without an increase in internal energy.

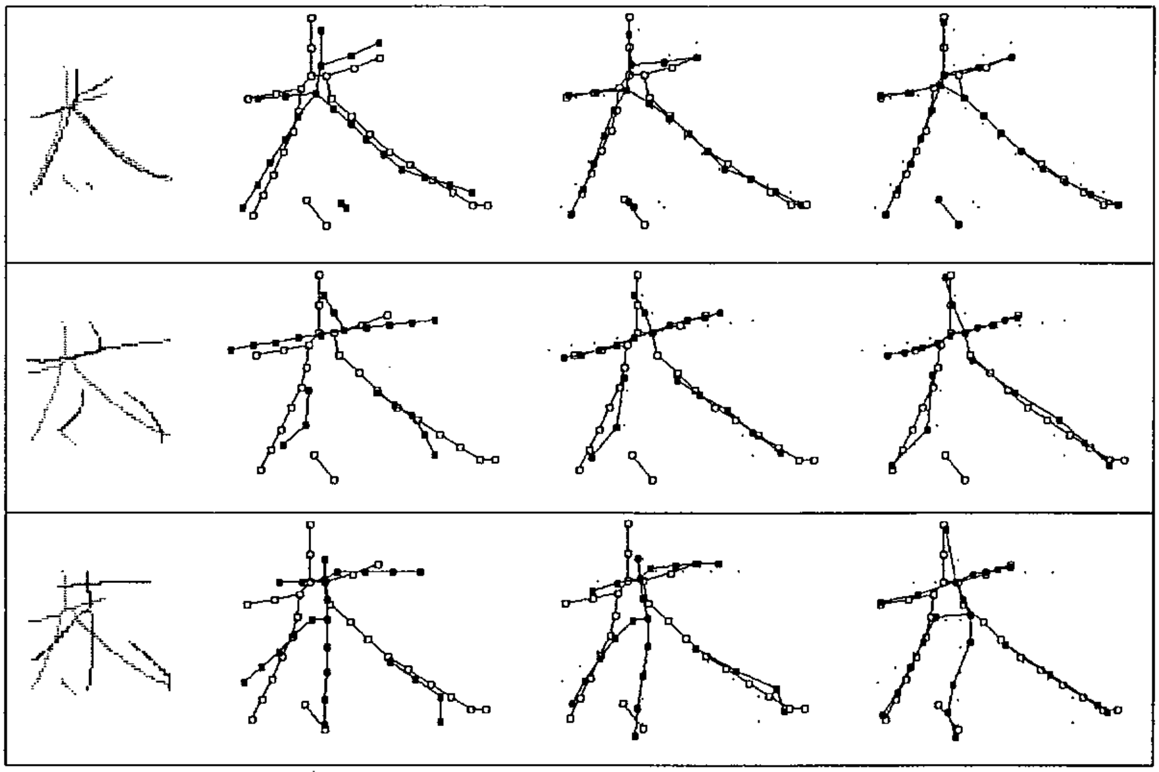


Figure 5.10 The deformation of top three choices of template candidates, 太, 六 and 木 when feeding an image 太.

Table 5.8 Energy values after deformation in Figure 5.10

	$E_{int}\left(E_{pseudo}+E_{intra}\right)$	E_{ext}	Ψ	$E_{clustering}$
太-太 matching	1.91	7.56	10.91	5
太-六 matching	4.73	33.41	41.69	20
太-木 matching	10.81	30.33	49.25	34

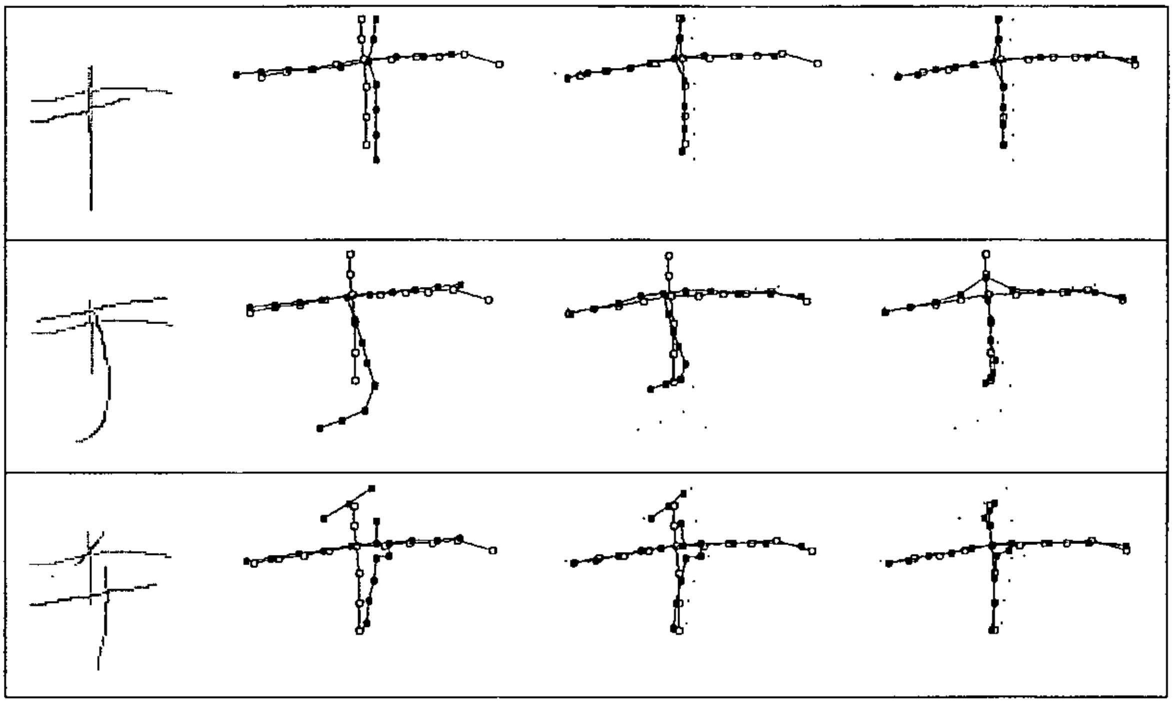


Figure 5.11 The deformation of top three choices of template candidates, 十, 丁 and 千 when feeding an image 十.

Table 5.9 Energy values after deformation in Figure 5.11

	$E_{int}\left(E_{pseudo}+E_{intra}\right)$	E_{ext}	Ψ	$E_{clustering}$
十 - 十 matching	0.80	6.07	7.48	0
十 - 丁 matching	7.14	11.09	23.59	21
十 - 千 matching	1.71	6.15	9.15	17

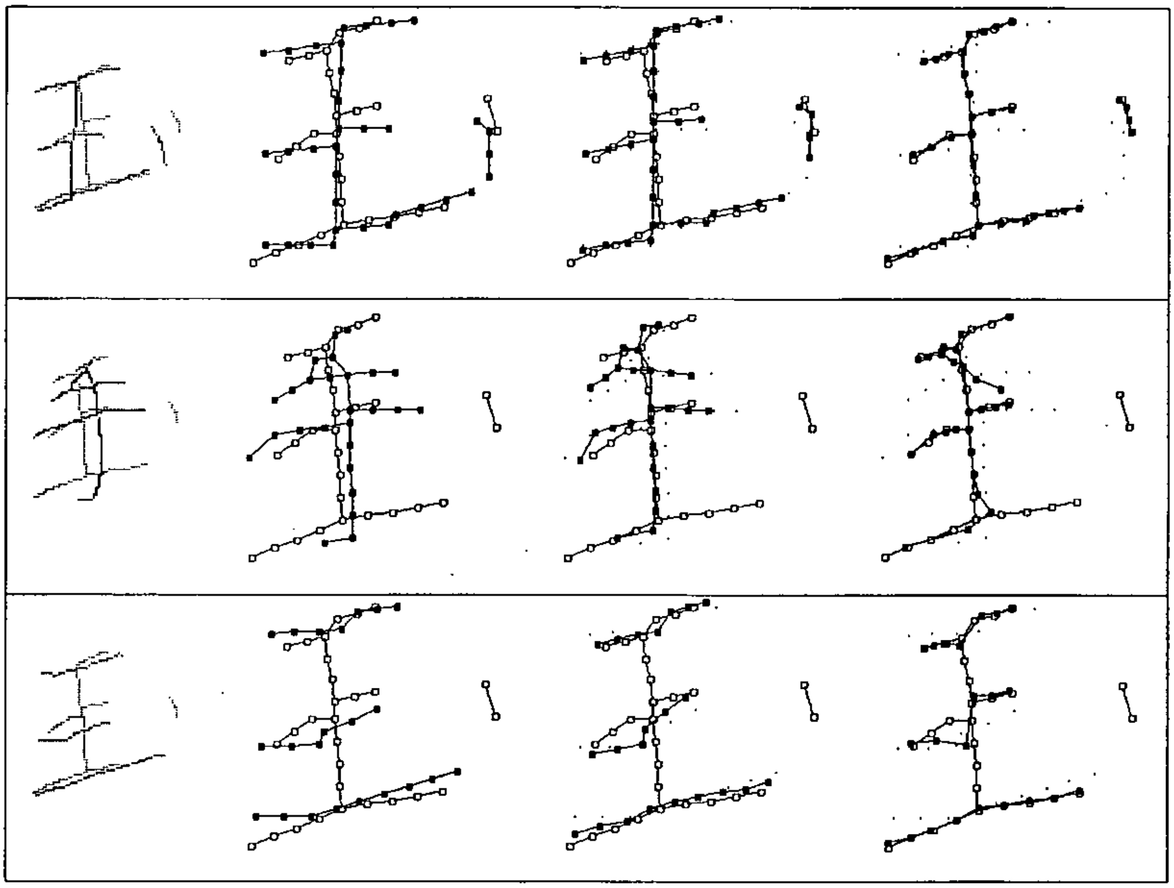


Figure 5.12 The deformation of top three choices of template candidates, 玉, 手 and 三 when feeding an image 玉.

Table 5.10 Energy values after deformation in Figure 5.12

	$E_{int}\left(E_{pseudo}+E_{intra}\right)$	E_{ext}	Ψ	$E_{clustering}$
玉 - 玉 matching	2.28	7.35	11.34	1
玉 - 手 matching	10.79	37.72	56.60	66
玉 - 三 matching	3.62	20.96	27.30	46

The last example of correct classification we demonstrate is depicted in Figure 5.12 and Table 5.10. In this case, the second and the third choices of the template candidates cannot be the winner as they are not capable to handle the right-most stroke of the image 玉, resulting in a high external energy and clustering error shown in Table 5.10. In addition, the point-to-edge matching strategy can be demonstrated by this example. Considering the right-most strokes of both the winning template 玉 and the image 玉, the former is composed of three snaxel edges while the latter is composed of one pixel edge. Since each snaxel and each pixel are required to perform a point-to-edge searching rather than a point-to-point searching, they can easily stay on the edge, and eventually a three-to-one edge mapping is resulted without any unnecessary overlapping of snaxels. Please note that the snaxel evenness still maintained even if the stroke is shortened.

Table 5.11 Confusion matrices of groups G1-G5 for SDM

input	G1 Recognition Result									
image	0	1	2	3	4	5	6	7	8	9
0	6		4							
1	1	8		1						
2			10							
3				9				1		
4					9		1			
5			1			9				
6	1						9			
7	1							9		
8									10	
9										10

input	G2 Recognition Result									
image	0	1	2	3	4	5	6	7	8	9
0	5		5							
1		9		1						
2			9	1						
3				10						
4					10					
5						10				
6							10			
7								10		
8									10	
9										10

input	G3 Recognition Result									
image	0	1	2	3	4	5	6	7	8	9
0	10									
1		9			1					
2			10							
3				10						
4		2			8					
5						10				
6	1						9			
7						2		8		
8	2								8	
9				1						9

input	G4 Recognition Result									
image	0	1	2	3	4	5	6	7	8	9
0	9				1					
1		10								
2			7	3						
3				10						
4	1				8					1
5						10				
6							10			
7								10		
8								1	9	
9					1					9

input	G5 Recognition Result									
image	0	1	2	3	4	5	6	7	8	9
0	9	1								
1		10								
2			10							
3				10						
4		1			9					
5						10				
6							9	1		
7								9		1
8									10	
9										10

Regarding the misclassification cases, Table 5.11 lists out the confusion matrix of each group, i.e., G1-G5, for SDM. It is found that most of the misclassification is mainly due to the large distortion between the desirable template pattern and the image, and also the high deformation flexibility of our model such that some undesirable template patterns may deform into an image easily without any severe penalty introduced. One of the cases is shown in Figure 5.13 in which an image 毋 is misclassified as the class 由. It is mainly due to the fact that the top horizontal stroke of the template 毋 is much longer than the corresponding part in the image and that cannot compensate the penalty introduced by the top vertical stroke in the template 由 which has been overlapped with other snaxel edges eventually. It is reflected by the values of clustering error $E_{clustering}$ and the external energy E_{ext} shown in Table 5.12 that they are much higher in "毋-毋" matching than in "毋-由" matching. Although in the proposed model, this horizontal stroke can be shortened without any penalty as long as it can maintain its evenness, it depends on the actual case. If snaxels inside find some other pixel edges which are much closer, they will be attracted towards them. Like in this case, a left-most vertical stroke in the image is found to be much closer and as a result this horizontal stroke is lengthened rather than shortened.

Another misclassification example is shown in Figure 5.14 in which an image 目 is misclassified as the class 月. In this case, the lower right stroke of the template 月 consists of two segments (right vertical and bottom horizontal) since a high curvature point exists at the corner. As in SDM, each segment is allowed to lengthen or shorten itself as long as it can maintain the evenness of its internal snaxels, this bottom horizontal segment can align with the bottom stroke of the image 目 easily by lengthening itself without much penalty introduced in E_{int} (see Table 5.13). Even in this case, its internal energy E_{int} is smaller than that of " 目 - 目 " matching as the template 目 may need to cater for other image distortions. The misclassification is mainly because the deformation flexibility of the SDM is so great that an undesirable template can be deformed towards an image easily without a large penalty encountered in the finer stage of matching. Although the smoothing scheme is introduced to incorporate some global deformation ability to the model, it applies in coarse-to-fine manner that local deformation should be resulted finally. This kind of misclassification is considered uneasy to be tackled because local deformation should be allowed for detail matching.

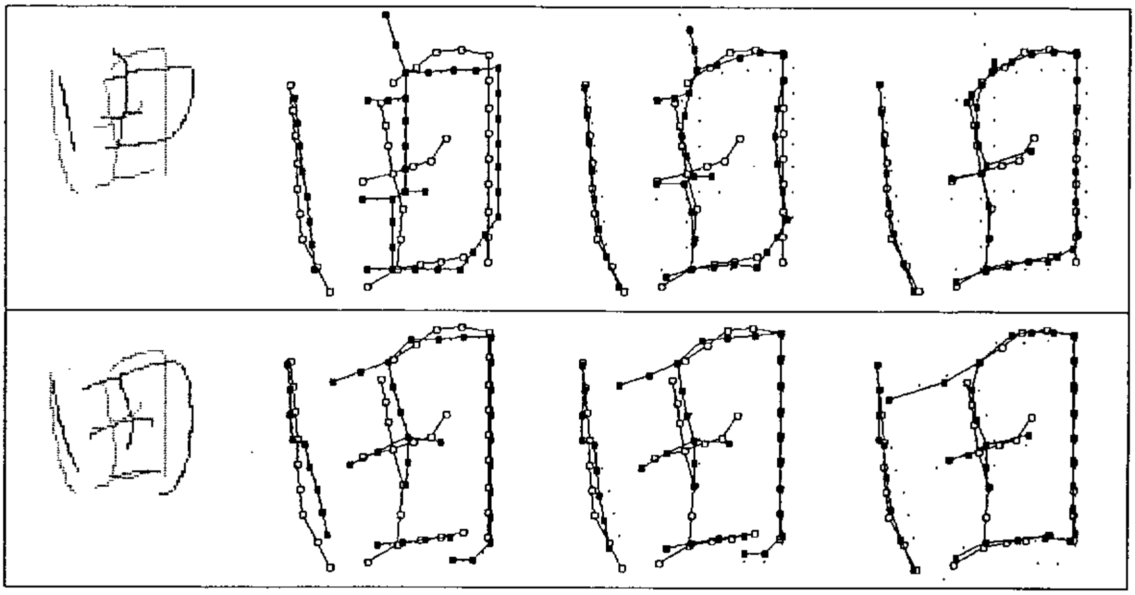


Figure 5.13 These two rows are corresponding to the deformation of two template patterns, 由 and 田 respectively when feeding an image 田.

Table 5.12 Energy values after deformation in Figure 5.13

	$E_{int}\left(E_{pseudo}+E_{intra}\right)$	E_{ext}	Ψ	$E_{clustering}$
田-由 matching	5.55	9.40	19.10	31
田-田 matching	4.93	22.33	30.97	49

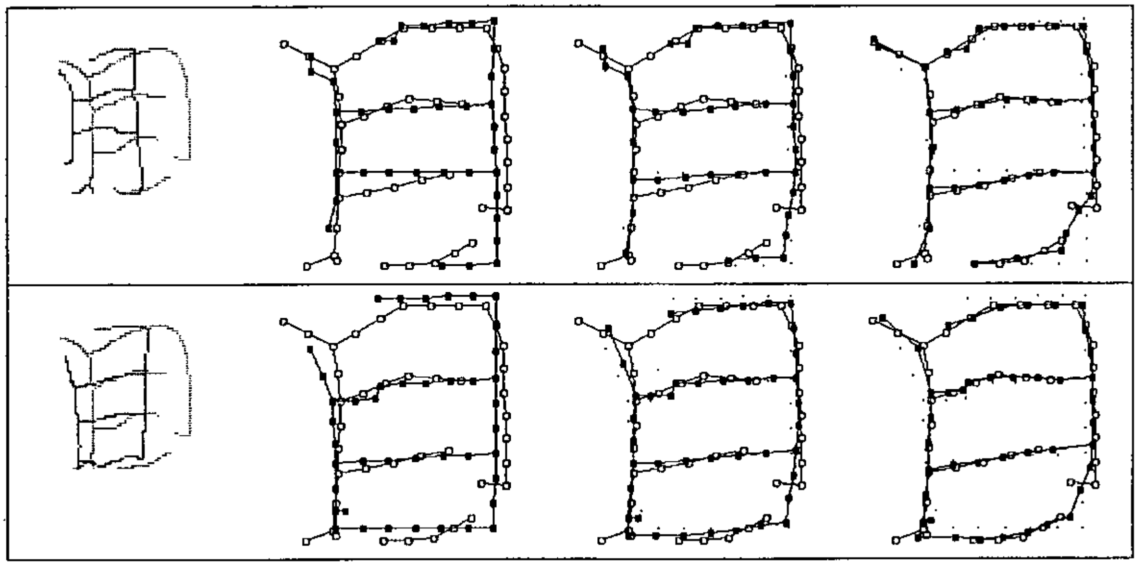


Figure 5.14 Deformation when feeding an image 目 to template patterns 月 and 目 .

Table 5.13 Energy values after deformation in Figure 5.14

	$E_{int}\left(E_{pseudo}+E_{intra}\right)$	E_{ext}	Ψ	$E_{clustering}$
目 - 月 matching	3.86	14.06	20.80	27
目 - 目 matching	5.85	16.13	26.38	56

5.5.2 Recognition as Post-processing of Statistical Based Systems

Figure 5.15 shows the recognition rate obtained by the SDM and RSM when both are treated as a post-processor of statistical based recognizers. All data points shown here were recorded as an average of five readings obtained from the five groups V1-V5. Details of the recognition rate and the confusion matrices of groups V1-V5 are shown in Table 5.14 and 5.15 respectively. It is observed that the proposed model outperforms the RSM by attaining a 96.0 % accuracy for the first choice. Since the candidates produced by a statistical based system do not ensure to have high similarity in structure, the SDM can easily get them differentiated and so a high recognition rate compared with that of the previous section is resulted. This is also true for RSM.

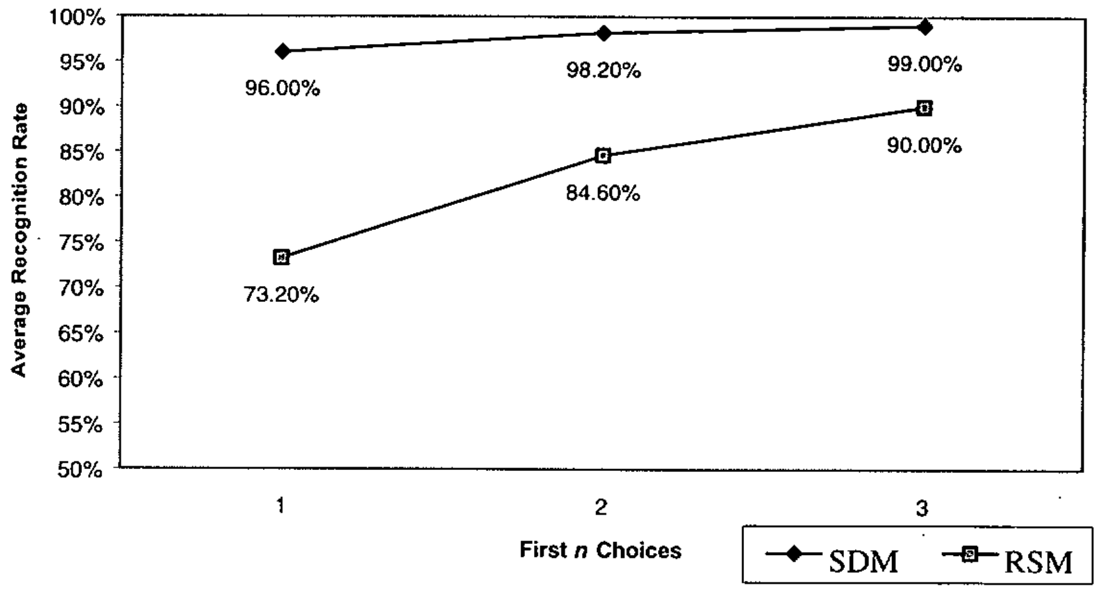


Figure 5.15 The performance comparison of the SDM and RSM : post-processing of statistical based systems

Table 5.14 Details on the recognition rate of the SDM and RSM : post-processing of statistical based systems

Group	First choice		First 2 choices		First 3 choices	
	SDM	RSM	SDM	RSM	SDM	RSM
V1	95 %	72 %	98 %	84 %	99 %	89 %
V2	95 %	74 %	98 %	83 %	99 %	89 %
V3	99 %	71 %	99 %	86 %	100 %	91 %
V4	95 %	72 %	99 %	83 %	99 %	88 %
V5	96 %	77 %	97 %	87 %	98 %	93 %
Average	96.00 %	73.20 %	98.20 %	84.60 %	99.00 %	90.00 %

Table 5.15 Confusion matrices of groups V1-V5 for SDM (For each group shown in Table 5.1, character categories are numbered from left to right and top to bottom)

input	V1 Recognition Result									
image	0	1	2	3	4	5	6	7	8	9
0	10									
1		10								
2			10							
3				10						
4					10					
5						10				
6							10			
7	1					3		6		
8									9	1
9										10

Input	V2 Recognition Result									
Image	0	1	2	3	4	5	6	7	8	9
0	8	2								
1		9	1							
2			9	1						
3				10						
4					10					
5						10				
6							10			
7				1				9		
8									10	
9										10

input	V3 Recognition Result									
image	0	1	2	3	4	5	6	7	8	9
0	9	1								
1		10								
2			10							
3				10						
4					10					
5						10				
6							10			
7								10		
8									10	
9										10

input	V4 Recognition Result									
image	0	1	2	3	4	5	6	7	8	9
0	9		1							
1		10								
2			10							
3				10						
4					10					
5						10				
6							10			
7								10		
8			1						8	1
9				2						8

Input	V5 Recognition Result									
Image	0	1	2	3	4	5	6	7	8	9
0	10									
1		10								
2		1	9							
3				10						
4					10					
5						10				
6							10			
7								8		2
8									10	
9			1							9

5.5.3 Stand-alone Recognition

An additional set of experiment is also conducted to investigate the performance of SDM and RSM by treating them as stand-alone recognizers. A 1-out-of-50 stand-alone classification by utilizing all 50 character categories in Table 5.1 is carried out. The overall performance is plotted in Figure 5.16 and it is observed that SDM has achieved a 87.80% accuracy for the first choice and outperforms the RSM again.

Although RSM can model almost any 2D image pattern, nearly all structural information inside the pattern has been ignored. Consequently, deformation can only be made based on the spatial relationship between primitives rather than on the their structure. Comparatively, the deformation flexibility of SDM is much higher than that of the RSM since the deformation of SDM is made according to the image structure, and that in turn makes it more able to cope with the complex distortion in the image during matching. It is reflected by the experimental results reported in these three sections that the SDM has attained a superior performance compared with RSM.

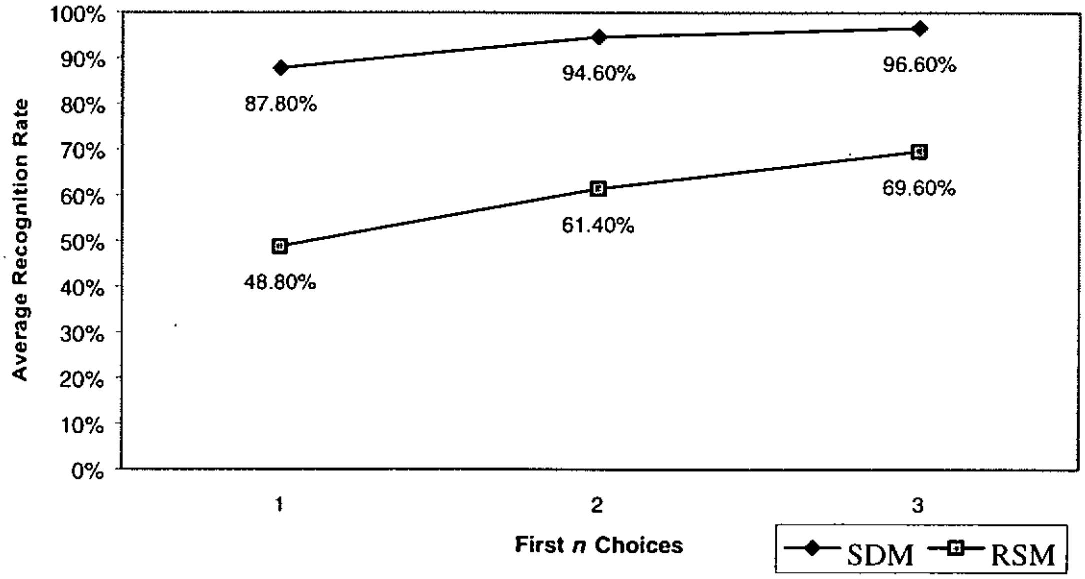


Figure 5.16 The performance comparison of the SDM and RSM : stand-alone recognition

5.6 Large Scale Experiment

A large scale experiment which tries to justify the practical use of the proposed model has been conducted. 200 Chinese character categories (including 50 in Table 5.1) from the same database have been extracted and there are altogether 4000 image samples for verification. These characters are grouped into four tables. Like the first table as shown in Table 5.1, the other three (summarized in Appendix B) are also divided into five structural groups (G1-G5) and five statistical groups (V1-V5). Inside each structural group, characters are specially selected from the database with high similarity in structure (say, with same radical) and are simulated as the candidates from structural based systems while the characters inside each statistical group are simulated as the candidates from statistical based systems.

In order to compare with a statistical recognizer, the model proposed by Li and Yu [34] has also been implemented and it is denoted by "STAT" in the following. As a result, the performance of three approaches, namely, SDM, RSM and STAT can be compared and are summarized in Table 5.16. Among these 4000 samples, half of them are testing samples and another half is training samples. The performance of these three approaches when they are used to verify the candidates from testing samples and from training samples are recorded. The result of the former one is plotted in Figure 5.17 in which the SDM outperforms the other two approaches in recognizing unknown images. However, the performance of the SDM in recognizing training samples is not as good as that of the STAT. It is because the calculation of Gaussian density distribution in each class by STAT is much more accurate than that by SDM in which the mean cannot be accurately calculated but is implicitly represented by the position of the template. Finally, the average performance of three approaches in recognizing both testing and training samples is plotted in Figure 5.18. Although it is observed that the performance of the proposed model is comparable with that of STAT and much higher than that of RSM, it can merely be proposed as a post-processor of other recognizers due to its great computational cost.

Table 5.16 Details on the recognition rate of STAT, SDM and RSM for a large scale Chinese character recognition

	Post-processor of structural based systems (average of G1-G5) First choice			Post-processor of statistical based systems (average of V1-V5) First choice			Overall Average
	Testing samples	Training samples	Average	Testing samples	Training samples	Average	
STAT	80.8 %	97.6 %	89.2 %	88.4 %	99.3 %	93.9 %	91.53 %
SDM	88.8 %	93.2 %	91.0 %	93.3 %	97.1 %	95.2 %	93.05 %
RSM	49.7 %	54.8 %	52.2 %	62.5 %	64.3 %	63.4 %	57.80 %

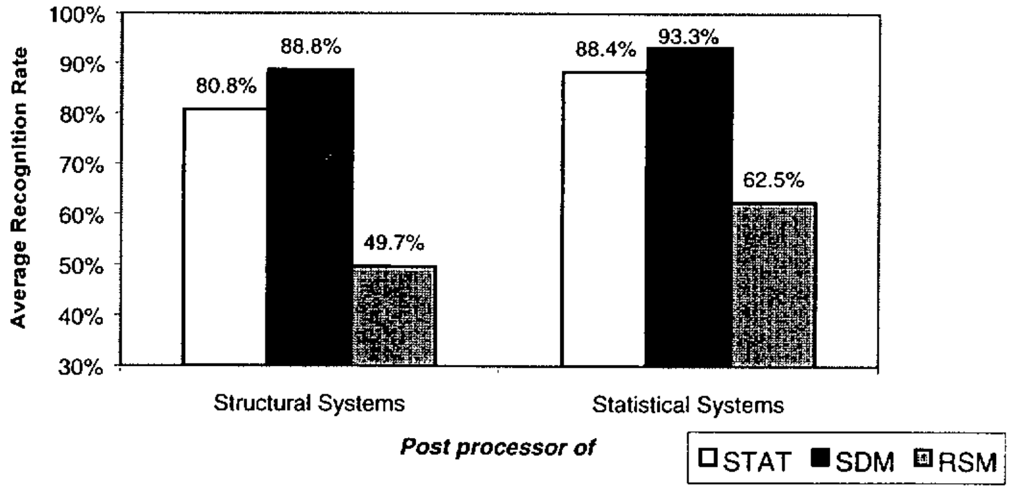


Figure 5.17 The performance of STAT, SDM and RSM in verifying testing samples

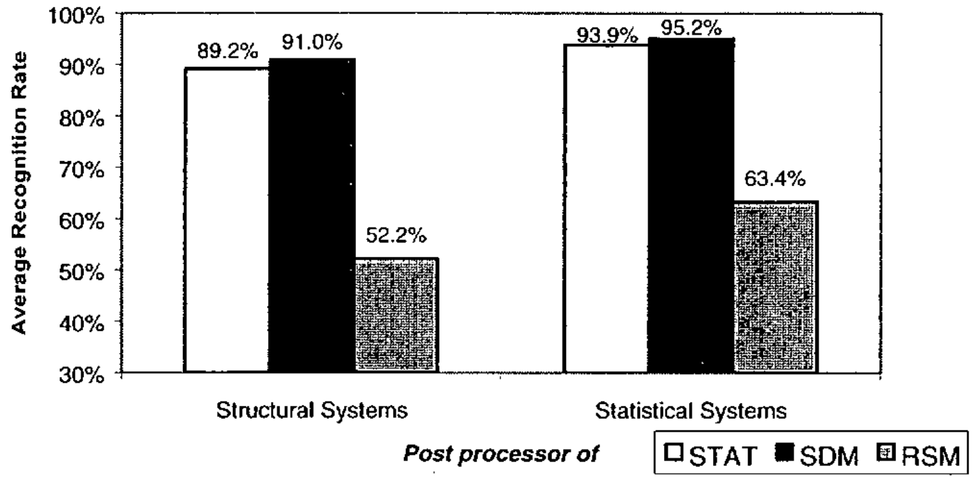


Figure 5.18 The performance of STAT, SDM and RSM in verifying both the testing and training samples

Chapter 6

CONCLUSIONS

6.1 Contributions

In this work, a SDM which explicitly takes structural information into its modelling process and is able to deform in a well-controlled manner through an ability of global-to-local deformation has been developed. The preliminary versions of SDM have been published in [35-36]. The main contributions of this work can be summarized as follows:

Incorporation of Structural Information

It is considered that if deformation can be made according to the pattern structure, the flexibility of deformation will be increased, making the model more able to cope with the complex distortion in the image during matching. However, it is observed that most of the existing DMs do not incorporate structural information into the model, and can merely deform according to the spatial relationship between primitives rather than the structure of patterns. To address this issue, a new class of DMs called structural deformable model (SDM) which explicitly takes the structural information into accounts and is able to model complex structure has been proposed. It works by modelling an image as a structured entity which is constructed by a hierarchy of components, namely, image, objects, snakes, segments and snaxels that are structurally connected with each others. With proper formulation of an internal energy functional to maintain its inter-object and intra-object structure, SDM results to have flexible deformation based on the pattern structure.

Global-to-local Deformation Ability

It is observed that if SDM deforms merely according to its structure, the flexibility of deformation would be too large that it may start to deviate significantly from its original shape and search locally once the deformation starts. In order to guide the model to deform globally at the earlier stage and locally at the latter stage of the deformation process, a smoothing scheme has been proposed to actively restrict the movement of model primitives according to their spatial relationship. Some parameters are used to adjust the relative weighting between the global and the local deformation on the model such that a coarse-to-fine matching can be achieved. As a result, the model has not only been given a high flexibility of deformation according to the structure, but also a capability of global-to-local deformation.

Formulation of Deformable Matching as a Feature Extraction Method

Feature extraction is both an important and difficult process in pattern recognition. Difficulties do not lie on how features are extracted but what they are as there is no general method for choosing or designating features. In this work, the SDM has been newly developed as a feature extraction method which is expected to be able to give a more separable clustering in feature space. Apart from the resultant value of objective function which corresponds to the core feature of the dissimilarity between images, an additional feature being extracted called the clustering error is newly introduced aiming at providing the information - alignment fitness to the classification. From the experiments, this additional feature is found to be able to give rises to the model performance, which further increases the capability of the model to differentiate between like images.

6.2 Limitations and Suggestions for Further Research

Multi-resolution Matching Strategy

Almost all applications adopting deformable model based approach suffer from the same problem - a great computational cost. The SDM also cannot escape from this and takes about 10 seconds in average for each deformable matching (excluding preprocessing) in SUN Ultra5 workstations. In order to speed-up the process, a multi-resolution matching strategy is suggested for its ability of skipping most of local minima and deforming to the desired object quickly. Currently in the SDM, the sampling rates of snaxels and pixels are kept fixed during the course of deformation. In fact, a speed-up on the deformable matching can be achieved by starting the sampling rates at small values (coarse stage) and increasing gradually.

Incorporation of Noise Model

SDM has a capability to differentiate highly similar patterns as it treats all pixels or pixel edges in the image as valuable matching primitives. It is this capability that makes the SDM sensitive to noise. In order to equip it with both the capability of differentiation between highly similar patterns and the ability to be insensitive to noise, a modification to the model can be made by introducing a noise model in the data match formulation, like the one proposed by Revow *et al.* [12]. For this kind of noise model, a value called "noise ratio" can be adjusted to specify the amount of noise existing in the image. It is in fact a probability imposed on all image pixels or pixel edges to give each of them the possibility to be a valuable matching primitive or a noise. The trade-off can be made by tuning this noise ratio. The larger its value, the more the model is noise insensitive but at the same time, the less it is capable to differentiate highly similar patterns.

Writing Styles Extraction

SDM, apart from its ability to do pattern matching, is a good tool in extracting writing style from human handwritings for its shape varying capability. This kind of writing style can be used in many ways, especially in enhancing recognition rate by influencing the way how DM deforms and its template selection process. However, such writing style is quite difficult to define and is considered as an interesting topic worth for further investigation.

Using Both Classification Rankings

As a post-processor of other recognition systems, the classification scheme of the SDM can be enhanced by considering both the original ranking from other recognition systems and that resulted from the SDM itself. In this way, the classification result will not be biased by the SDM. Even, it can give rises to the confident level of the correctness of the result because the class label should be agreed by both recognition schemes before it wins.

Investigation on Feature Designation

In the current stage for the SDM, only two features are extracted and employed in classification; namely, the resultant value of objective function and the clustering error. As it is possible that there are still many features which can give a more well separated clustering, a possible direction for further research is to investigate on the feature designation. Although previously feature designation is stated to be a difficult task, it only applies to statistical approach but not deformable matching approach. It is because choosing features to represent the dissimilarity between images (deformable matching approach) is easier than choosing features to characterize an image (statistical approach).

Appendix A

INTEGRATION OF E_{even} AND E_{orient} FOR INTRA-OBJECT SHAPE PRESERVATION CRITERION

The integration of E_{even} and E_{orient} for intra-object shape preservation criterion is presented here in detail. Whether it is minimizing E_{even} or E_{orient} alone, a force will be induced to bring the model to a low energy state. Let them be F_{even} and F_{orient} respectively. For a force, it has both the magnitude and direction. However, even if the magnitude of F_{even} and F_{orient} are neglected, it is not certain whether their directions are the same all the time. It is possible that sometimes they are 180° out of phase. An example is depicted in Figure A.1 in which the edge (P_1,P_2) is deformed in a way that two induced forces, F_{even} and F_{orient} are nearly in opposite direction. (The original edge and the deformed edge are denoted as $(\underline{P}_1,\underline{P}_2)$ and (P_1,P_2) respectively) As it becomes shorter (assuming the average length remains unchanged), F_{even} is pointed outwards. As it deviates from the original orientation, F_{orient} is pointed inwards. In this way, although both energies are constructed for preserving intra-object shape, they could not cooperate all the time and sometimes, their forces are cancelled out against each another. It is another reason for us to integrate E_{even} and E_{orient} apart from the requirement of parameters determination.

Considering the energy functional E_{even} defined in eq.(3.15), D_k is the averaged inter-snaxel distance in each segment. After each deformation iteration, D_k should be updated to reflect the up-to-date averaged inter-snaxel distance in each segment. In fact, if D_k is fixed during the course of deformation, it is a special case of maintaining evenness that it actually preserves original edge length. In the following, let us first consider that D_k is fixed and as a result, integrating E_{even} and E_{orient} will result in a process of preserving original edge length and orientation. As it is translational invariant, all edges can be modelled as vectors and the problem is then reduced to the formulation of a single force for keeping the size and direction of each vector preserved.

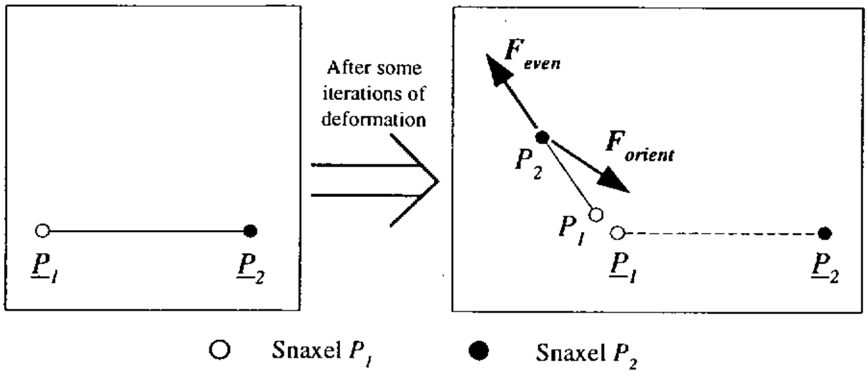


Figure A.1 The possibility of the cancellation of forces induced by E_{even} and E_{orient}

Consider a case that there is an edge inside a template pattern. The original positions of its two terminal snaxels are denoted by two vectors, P_1^o and P_2^o. At t-th iteration of deformation, their positions are denoted by P_1^t and P_2^t, and are shown in Figure A.2. Having modelled by vectors L^o and L^t respectively, the preservation of the original edge length and orientation can be achieved by simply minimizing the distance d which is given by

$$\begin{aligned} d &= \left\| \; L^t - L^o \; \right\| \\ &= \left\| \; \left(P_2^t - P_1^t\right) - \left(P_2^o - P_1^o\right) \; \right\| \end{aligned} \qquad \text{(A.1)}$$

By minimizing the distance d, a force F will be induced following the path of d down to the original vector L^o. Note that the preservation of original edge length and orientation is eventually enabled by only one force shown in the figure. Refer back to the energy functional E_{even}. So far, D_k is kept fixed during the course of deformation. In order to have the model preserving the evenness of inter-snaxel distance, D_k should be updated at each deformation iteration to reflect the up-to-date average inter-snaxel distance in its segment. To incorporate this process into the existing mechanism, it can be implemented by periodically updating the length of L^o like that of D_k while keeping its direction unchanged.

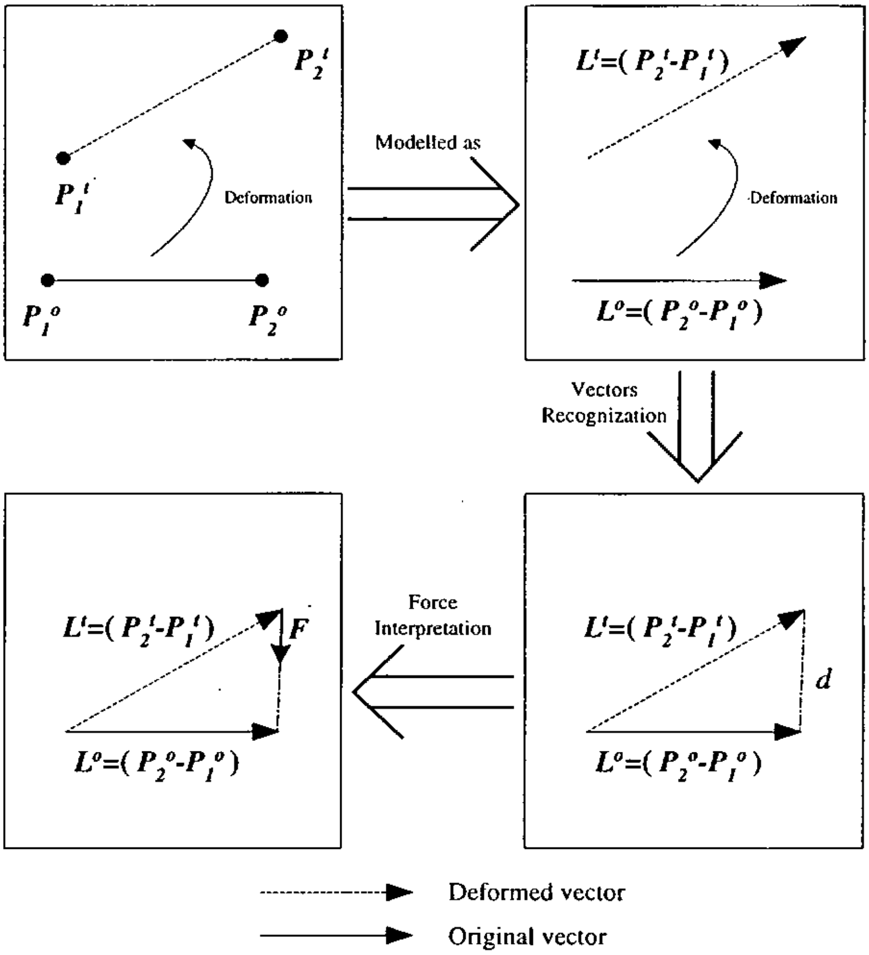


Figure A.2 Integration of forces, F_{even} and F_{orient}

To make it generalized for the whole image pattern, the preservation of both the evenness of inter-snaxel distance within each segment and edge orientation can be achieved by minimizing an energy functional E_{intra} which is given by

$$E_{intra} = \frac{1}{\sum_{k}^{M}(N_k - 1)} \sum_{k}^{M} \sum_{i}^{N_k - 1} \left\| O(T_i^k) - \hat{O}^o(T_i^k) \right\|^2 \tag{A.2}$$

where

M is the total number of segments,

N_k is the total number of snaxels inside segment k, and

T_i^k is the position vector of snaxel i in segment k

$\hat{O}^o(T_i^k)$ denotes the vector in same direction as $O^o(T_i^k)$ which is defined in Section 3.4.2. If its length is fixed at a value the same as that of $O^o(T_i^k)$, minimizing E_{intra} will result in the preservation of original edge length and orientation. If its length is kept updating to reflect the up-to-date average inter-snaxel distance in its segment (i.e., segment k), minimizing E_{intra} will result in the preservation of snaxel evenness and edge orientation. Needless to say, the latter strategy will be adopted by the SDM.

Appendix B

LARGE SCALE DATABASE

Totally 200 KANJI character categories are extracted from the database "IPTP CD-ROM2" [32]. They are grouped into four tables. Table 5.1 is the first one while the other three are summarized below as Table B.1 – B.3. Similar to Table 5.1, each of them is divided into five structural groups (G1-G5) and five statistical groups (V1-V5). Inside each structural group, characters are specially selected from the database with high similarity in structure, such as the ones with same radical or with similar patterns. For example, characters with the same radical "宀", "亻" and "艹" are grouped respectively in Table B.1 G2, Table B.2 G1 and Table B.3 G2. They are simulated as the candidates from structural based systems while the characters inside each statistical group are simulated as the candidates from statistical based systems.

Table B.1 The second table of characters inside the large database

	V1		V2		V3		V4		V5	
G1	内	向	南	岡	門	間	関	園	国	西
G2	宇	宿	字	宮	富	安	室	守	家	寒
G3	合	分	今	谷	金	倉	会	市	奈	矢
G4	沿	清	波	池	江	沢	沼	浦	河	湯
G5	津	浜	海	港	深	濃	渡	淀	滑	瀬

Table B.2 The third table of characters inside the large database

	V1		V2		V3		V4		V5	
G1	伏	仁	任	仙	佐	代	伊	保	住	伯
G2	松	横	根	相	杉	札	村	林	板	植
G3	橋	柳	柏	梅	枝	森	秋	利	和	穂
G4	地	場	堀	垣	堺	城	塚	坂	塩	増
G5	原	府	鹿	摩	庄	麻	庫	唐	磨	厚

Table B.3 The fourth table of characters inside the large database

	V1		V2		V3		V4		V5	
G1	吉	台	名	吹	古	岩	知	品	加	砂
G2	長	若	葉	草	荒	落	芦	芸	荘	葛
G3	市	高	京	方	前	花	並	美	益	道
G4	萩	藤	蒲	菊	蔵	茅	苫	笠	笹	篠
G5	部	都	郡	郷	那	阪	阿	陽	院	御

REFERENCES

[1] R. Schalkoff, *Pattern recognition – Statistical, structural and neural approaches.* Wiley, 1992.

[2] K. Fukunaga, *Introduction to statistical pattern recognition.* Academic Press, 1990.

[3] J.T. Tou and R.C.Gonzalez, *Pattern recognition principles.* Addison-Wesley, 1981.

[4] K.S. Fu, *Syntactic pattern recognition and applications.* Prentice-Hall, 1982.

[5] R.C. Gonzalez and M.G. Thomason, *Syntactic pattern recognition – An introduction.* Addison-Wesley, 1982.

[6] T.F. Cootes, C.J. Taylor, D.H. Cooper and J. Graham, "Active shape models – Their training and applications," *Computer Vision and Image Understanding,* vol.61, no.1, pp.38-59, 1995.

[7] L.H. Staib and J.S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.14, no.11, pp.1061-1075, Nov. 1992.

[8] M.D. Jolly, S. Lakshmanan and A.K. Jain, "Vehicle segmentation and classification using deformable templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.18, no.3, pp.293-308, Mar. 1996.

[9] A.K. Jain and A. Vailaya, "Shape-based retrieval : a case study with trademark image databases," *Pattern Recognition*, vol.31, no.9, pp.1369-1390, 1998.

[10] A.K. Jain and D. Zongker, "Representation and recognition of handwritten digits using deformable templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.19, no.12, pp.1386-1391, Dec. 1997.

[11] T. Wakahara, "Shape matching using LAT and its application to handwritten numeral recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.16, no.6, pp.618-629, Jun. 1994.

[12] M. Revow, C.K.I. Williams and G.E. Hinton, "Using generative models for handwritten digit recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.18, no.6, pp.592-606, Jun. 1996.

[13] J. Bertille, "An elastic matching approach applied to digit recognition," *Proc. Second Int. Conf. on Document Analysis and Recognition*, pp.82-85, 1993.

[14] H. Nishida and S. Mori, "Algebraic description of curve structure," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.14, no.5, pp.516-533, 1992.

[15] K.W. Cheung, D.Y. Yeung and R.T. Chin, "A unified framework for handwritten character recognition using deformable models," *Proc. Second Asian Conf. on Computer Vision*, Singapore, pp.344-348, 1995.

[16] B. Widrow, "The rubber-mask technique - I. Pattern measurement and analysis," *Pattern Recognition*, vol.5, no.3, pp.175-197, 1973.

[17] D.J. Burr, "Elastic matching of line drawings," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.3, no.6, pp.708-713, Nov. 1981.

[18] M. Kass, A. Wttkin and D. Terzopoulos, "Snakes: active contour models," *Int. Journal of Computer Vision*, pp.321-331, 1988.

[19] A.K. Jain, Y. Zhong and S. Lakshmanan, "Object matching using deformable templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.18, no.3, pp.267-277, Mar. 1996.

[20] Y. Gong, *Intelligent Image Databases: Towards Advanced Image Retrieval.* Kluwer Academic, 1998.

[21] A.A. Amini, T.E. Weymouth and R.C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.12, no.9, Sep. 1990.

[22] D.J. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *CVGIP: Image Understanding*, vol.55, no.1, pp.14-26, Jan. 1992.

[23] C.T. Tsai, Y.N. Sun and P.C. Chung, "Minimising the energy of active contour model using a hopfield network," *IEE Proceedings-E*, vol.140, no.6, Nov. 1993.

[24] K.W. Cheung, *Bayesian frameworks for deformable pattern recognition and retrieval.* PhD Thesis, Department of Computer Science, Hong Kong University of Science and Technology, Jan. 1999.

[25] K.W. Cheung, D.Y. Yeung and R.T. Chin, "A bidirectional matching algorithm for deformable pattern detection with application to handwritten word retrieval," To appear in *Proc. Seventh IEEE Int. Conf. on Computer Vision*, Kerkyra, Greece, 20-27 Sep. 1999.

[26] S. Ranade and A. Rosenfeld, "Point pattern matching by relaxation," *Pattern Recognition*, vol.12, no.4, pp.269-275, 1980.

[27] T. Wakahara and K. Odaka, "On-line cursive kanji character recognition using stroke-based affine transformation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.19, no.12, pp.1381-1385, Dec. 1997.

[28] A.B. Wang, K.C. Fan and J.S. Huang, "Recognition of handwritten Chinese characters by modified relaxation methods," *Image and Vision Computing,* vol.12, no.8, pp.509-522, Oct. 1994.

[29] T.Y. Zhang and C.Y. Suen, "A fast parallel algorithm for thinning digital patterns," Communications of the ACM, vol.27, no.3, pp.236-239, Mar. 1984.

[30] D.G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence,* vol.31, pp.355-395, 1987.

[31] P.L. Rosin, "Techniques for assessing polygonal approximations of curves," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.19, no.6, pp.659-666, Jun. 1997.

[32] IPTP CD-ROM2. Technology Development Research Center, Institute for Posts and Telecommunications Policy, Japan Ministry of Posts and Telecommunications.

[33] T. Tsutsumida, F. Kawamata, S. Yamaguchi, K. Nagata and T. Wakahara, "The third IPTP character recognition competition and study on multi-expert systems for handwritten Kanji recognition," *Progress in Handwriting Recognition.* A.C. Downton and S. Impedovo Ed., World Scientific, pp.299-304, 1997.

[34] T.F. Li and S.S. Yu, "Handprinted Chinese character recognition using the probability distribution feature," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol.8, no.5, pp.1241-1258, 1994.

[35] C.K.Y. Tsang and F.L. Chung, "Development of a structural deformable model for handwriting recognition," *Proc. Int. Conf. on Pattern Recognition,* Brisbane, Australia, pp.1130-1133, 16-20 Aug. 1998.

[36] C.K.Y. Tsang and F.L. Chung, "A stroke-based structural deformable model for handwriting recognition," *Proc. Fourth Joint Conf. on Information Sciences,* North Carolina, 23-28 Oct. 1998.