Abstract

Non-Rigid Point Matching: Algorithms, Extensions and Applications

Haili Chui

Yale University

2001

A new algorithm has been developed in this thesis for the non-rigid point matching problem. Designed as an integrated framework, the algorithm jointly estimates a one-to-one correspondence and a non-rigid transformation between two sets of points. The resulting algorithm is called “robust point matching (RPM) algorithm” because of its capability to tolerate noise and to reject possible outliers existed within the data points.

The algorithm is built upon the heuristic of “fuzzy correspondence”, which allows for multiple partial correspondences between points. With the help of the deterministic annealing technique, this new heuristic enables the algorithm to overcome many local minima that can be encountered in the matching process.

Devised as a general point matching framework, the algorithm can be easily extended to accommodate different speci£c requirements in many registration applications. Firstly, the modular design of the transformation module enables convenient incorporation of different non-rigid splines. Secondly, the point matching algorithm can be easily extended into a symmetric joint clustering-matching framework. It will be shown that by introducing a super point-set, the joint cluster-matching extension can be applied to estimate an average shape point-set from multiple point shape sets.

The algorithm is applied to the registration of 3D brain anatomical structures. We proposed in this work a joint feature registration framework, which is mainly based on the joint clustering-matching extension of the robust point matching. The proposed framework provides an effective and uni£ed way to utilize spatial relationship existed between different brain structural features to improve the brain anatomical registration/normalization. For the £rst time, a carefully designed synthetic study is carried out to investigate and compare different anatomical features’ abilities to achieve such an registration/normalization.

Other applications of the robust non-rigid point matching algorithm, such as key-frame animation and human face matching, will also be demonstrated in this work.

Non-Rigid Point Matching:

Algorithms, Extensions and Applications

A Dissertation

Presented to the Faculty of the Graduate School

of

Yale University

in Candidacy for the Degree of

Doctor of Philosophy

by

Haili Chui

Dissertation Director: Anand Rangarajan

January 2001

Acknowledgments

I extend warm thanks to my advisor, Professor Anand Rangarajan for his guidance and encouragement over all these years. The work presented in this thesis could not possibly be done without his inspirational ideas, valuable insights and consistent enthusiasm.

I would like to thank Professor James Duncan for his continuous support and for having given me the opportunity to come to work in the Yale Image Processing and Analysis group. It has been a great experience. I would also like to thank Professor Hemant Tagare and Professor Lawrence Staib for many stimulating discussions and for generously sharing their knowledge with me. Thanks also goes to Professor Peter Schulthesiss for being my Ph.D. programm committee member and for his valuable criticism.

I would also like to thank Professor Jitendra Malik at Univeristy of California at Berkeley for being the external reader for this thesis.

I thank Professor Robert Schultz, James Rambo, Joseph Walline and Lawrence Win for their much appreciated help in the process of data acquisition and analysis.

I also wish to thank the friends and fellow students at the IPAG group for their support. Thanks to Xiaolan Zeng and Ravi Bansal for many inspirational discussions. Xiaolan's previous work on segmentation and feature extraction has been a great help for my work. Thanks to Xenios Papademetris and Oskar Skrinjar for all the help and for sharing your visulization code. Thanks to Carolyn Meloling for all the administrative assistance and encouragement. Thanks to all the former and current members of IPAG, Pengcheng Shi, Reshma Munbodh, Thomas Casey, Gang Liu, Xiaoning Qian, Jing Yang, Ning Lin and Tao Zhong for making this experience fun. Thanks to my friends at the Yale Physics department, Hui Li, Bengyuan Liu and Changpu Xu for their friendship and for all the good time we had together.

The £nancial support for this work has been provided by grants from the Whitaker Foundation, National Science Foundation and National Institute of Health.

Finally, I would most like to thank my wife Jin and my family. They are the best part in my life.

Contents

Chapter 1

Introduction to Image Registration

In this work, we intend to study the non-rigid point matching problem in an abstract setting. However, we are mainly interested in the application of non-rigid point matching to the problem of medical image registration — a sub-problem within the £eld of medical imaging. Before launching into a detailed description of non-rigid point matching, we brie¤y describe the fundamental concepts in the domain of image registration and discuss the reasons which motivated us to study the non-rigid point matching problem.

1.1 What Is Image Registration ?

Image registration [38, 11, 93, 57] generally refers to the process of identifying and subsequently aligning corresponding structures and/or objects from two image representations. To register two images, a transformation (spatial mapping) must be found so that each location in one image can be mapped to a new location in the second. This mapping should "optimally" align the two images wherein the optimality criterion itself depends on the actual structures/objects in the two images that are required to be matched.

These concepts can be best explained through an example as shown in Figure 1.1 and 1.2. It is a typical image registration task. Two source images are given. Within each image, there is an object with a label. When we align the two images, we intend to only align the objects and not the labels. This can be accomplished by rotating the £rst image with a certain angle. The alignment is shown in Figure 1.2.

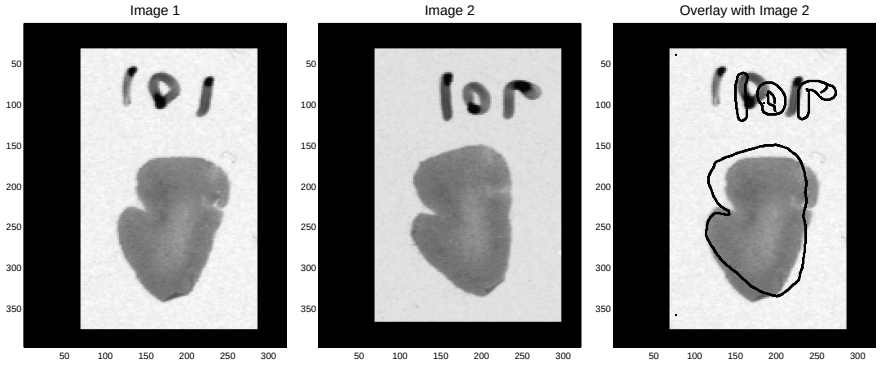

Figure 1.1: An example of image registration. Left and Middle: two source images are shown here. Each image contains an oval-like shape. The labels (seen upside down) are “101” and “106” respectively. Though the labels are also present in the images, only the oval objects need to be aligned. Right: the overlay of image 1 and the contour of image 2 is shown. There is clearly a misalignment.

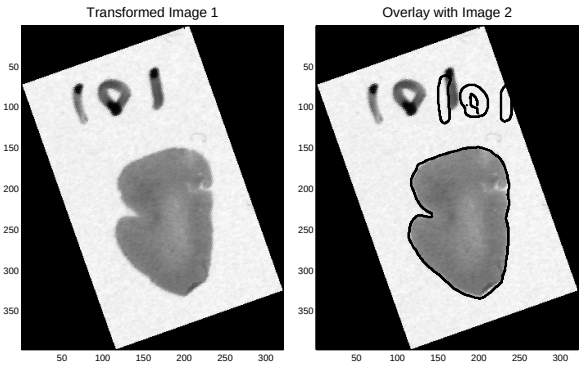

Figure 1.2: Alignment of the two images. The rotated image 1 is shown on the left. The overlay of the transformed image 1 and the contour of image 2 is shown on the right. The oval objects in the two images are clearly much better aligned whereas the labels are not.

1.2 Why Do We Need Image Registration ?

The need for image registration arises as a practical necessity in many diverse £elds. As pointed out in [11], a good alignment of images is necessary for i) integrating information from different images, ii) £nding changes in images taken at different times or under different conditions, iii) model based matching for the purpose of segmentation or object recognition. Image registration has a signi£cant impact on a wide range of various research £elds, such as computer vision, pattern recognition, medical image analysis and remotely sensed data processing. We include examples of some speci£c registration problems from these £elds in Tables 1.1, 1.2 and 1.3.

1.3 How Is Image Registration Normally Performed?

All image registration methods can be broken down into the following three steps:

1. Choosing the basis information best suited for matching.
2. Determining an adequate similarity metric or measure.
3. Searching for optimal or satisfactory values for the unknown variables characterizing the similarity measure.

The £rst step is to decide what kind of information can be extracted and used to represent the images for the purpose of matching. This information could very well be restricted to the set of image intensities. On the other hand, it could also be certain features extracted from the images. These feature can represent salient structures such as as strong edges, structural contours, line intersections and points of high curvature within the images.

In the second step, a similarity metric is chosen to provide a consistent way to evaluate the relative merit of a match. Such a choice is very dependent on the task at hand. The similarity measure is usually a function of some transformation parameters such rotation, translation etc.

Then, in third step, certain search strategies are implemented to search through the parameter space characterizing the similarity measure. The search is carried out until an adequate solution (again, whose de£nition depends on the task at hand) is found.

1.4 What Are the Different Types of Registration Methods ?

The choice of the basis information type is a fundamental aspect of all registration algorithms. It imposes an inordinate amount of in¤uence on the design of the similarity measure and subsequently the search strategy as well. It also

Integrating Information from Different Sources
Purpose: Registration of images of the same scene but acquired from different sensors for information fusion; registration of images taken from different viewpoints for 3D 3D reconstruction. **Example 1** Field: Medical Image Analysis Problems: (i) Integrate structural information from CT or MRI with functional information from other modalities such as fMRI, PET or SPECT; (ii) Align low quality treatment portal images with diagnostic high resolution planning CT images to improve the accuracy of radiotherapy. **Example 2** Field: Computer Vision Problems: Matching of images (e.g. stereo images) of the same 3D object but taken from different angles to recover the object's 3D shape. **Example 3** Field: Remote Sensed Data Processing Problems: Integrating images from different sensors (e.g., infared, visual or from other spectrum) for improved scene analysis.

Table 1.1: Image registration task class 1: multi-modality / viewpoint registration for the purpose of integrating information from different sources.

Finding Changes in Images Taken at Different Time
Purpose: Registration of images of the same scene but taken at different time for change detection or object tracking. **Example 1** Field: Medical Image Analysis Problems: Comparison of diagnostic images taken at different times to monitor normal/abnormal changes or growths: (i) Digital Substraction Angiography (DSA) - registration of images before and after radio isotope injections to identify functionality; (ii) Similar technology also apply for mammography for breast tumor detection, lung x-ray scans for cancer nodule detection and brain MRI scans for brain tumor detection. **Example 2** Field: Computer Vision or Remote Sensed Data Processing Problems: (i) Analysis of a moving object video sequence for object tracking; (ii) Surveillance of moving objects, weather pattern and natural resource distribution.

Table 1.2: Image registration task class 2: temporal registration for the purpose of £nding changes and tracking.

arguably sets an upper bound on the registration accuracy (but most likely only in a counterfactual sense). That is, if the “wrong” information type is chosen for matching, it may turn out that after the choice is set that even the best registration result is woefully inadequate in comparison to the choice of another “better” feature type. Based on these observations, we have used the choice of the basis information to orient us in our description of the taxonomy of registration methods.

As mentioned above, the basis information can just be the image intensity information at each pixel (voxel, if in 3D), or in some cases, a different intensity image produced via preprocessing by a £lter. Regardless of whether or not any £lter preprocessing is done, this type of method is fundamentally characterized by its direct reliance on pixel intensity information; consequently, they are referred to as *intensity-based* methods.

The usage of features leads to a different type of methods. With *salient* structures in the image represented

Model Based Matching
Purpose: Finding a match for a reference template (model) in an image **Example 1** Field: Medical Image Analysis Problems: Model based segmentation for locating a structure in an image that most closely resemble the chosen prior model. **Example 2** Field: Computer Vision and Pattern Recognition Problems: Shape/pattern matching for character recognition and signature veri£cation.

Table 1.3: Image registration task class 3: template registration for the purpose of recognition and/or segmentation.

as compact geometrical entities (e.g., points, lines, curves or surfaces), these methods seek to achieve registration indirectly through the alignment of the geometrical features. They are thus commonly called *feature-based* methods. Though the features require more effort to extract than image intensities, they provide a better representation of the image's intrinsic structures since the usage of features generally reduces the effects of scene and sensor noise. Further more, features offer a much more compact form of representation than the original intensity image, hence greatly reduce the computational cost in evaluating the similarity measure and subsequently facilitate the search process.

Therefore, we will focus on the feature-based approach in this work.

1.5 Focus of This Thesis.

We are mainly interested in one type of feature-based registration — non-rigid point matching. More speci£cally, in this work,

1. We focus on the feature-based approach.
2. We intend to use the point feature.
3. We seek to develop an algorithm for the feature point registration problem that is capable of handling non-rigid

transformations.

We just discussed some bene£ts of the feature-based approach. Note that the feature-based approach is not restricted to using one and only one type of feature. We choose the point feature because of several of its advantages which will be enumerated below.

Essentially represented by points' coordinates, the point feature is the simplest form of all features. In fact, it often serves as the basis upon which other more sophisticated feature representations, such as curves and surface, are built to incorporate additional information besides the coordinate information. With point feature being the foundation of all features, the point matching problem can be regarded as the most fundamental problem in the domain of feature-based registration as well. A careful investigation of point matching in the abstract setting should not only result in improved point feature-based registration but should also provide opportunities for advances in other feature-based registration methodologies as well.

Because of the bene£t and simplicity of using point features in image registration, point feature matching has attracted a lot of attention. An enormous amount of research has been done in the £eld of point matching. However, most previous effort has been concentrated on the simpler cases where only rigid transformations are involved.

Non-rigid transformations are needed for image registration tasks for deformable objects. Analyzing non-rigid motion has becomes a major research problem not only in computer vision and image processing but also, more dominantly, in medical imaging, where most of the objects being studied are deformable. As we demonstrate in the subsequent sections, the few existing algorithms that are capable (to different extents) of handling the non-rigid point matching problem have various shortcomings. Consequently, a major goal in this work is to develop a general purpose non-rigid point matching algorithm which is also well suited to real-world image registration tasks.

Chapter 2

Non-Rigid Point Matching Problem: A De£nition and Review

In this chapter, we begin by de£ning the non-rigid point matching problem and discuss why it is a dif£cult problem to solve. We then present a comprehensive review of the previous work that is related to this problem. At the end of this chapter, we brie¤y discuss the approach taken in this work and some of the resulting improvements.

2.1 The Non-Rigid Point Matching Problem

2.1.1 Transformation and Correspondence

A basic non-rigid point matching problem can de£ned as follows: given two sets of points (essentially their coordinates), we would like to £nd the *non-rigid transformation* that best maps one set of points onto the other and/or the set of *correspondence* (including possible outliers) between the points.

The point-sets can be aligned with a good transformation/spatial mapping, which takes one set of points and warps them closely onto the other set. The correspondence basically informs us as to which point in one set is the counterpart of a point in the other set. Usually an answer for the transformation/correspondence is considered to be a good answer when identical or similar structures presented within the two point-sets are aligned together or identi£ed to be corresponding.

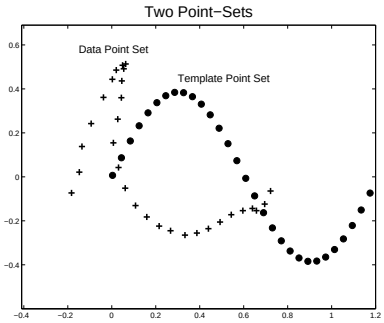

Figure 2.1: A simple non-rigid point matching problem. Two point-sets are shown. The goal is to align these two point-sets by deforming one of them (the template point-set depicted using dots) onto the other (the data or target point-set depicted using crosses).

The transformation and the correspondence are normally regarded as the two unknown variables[1] in a point-matching problem. They share an intimate relationship. Once one variable is known, the solution for the other is actually mostly trivial. Given the set of correspondences (including the set of outliers), £nding a good transformation is often a straightforward least-squares problem. On the other hand, given a transformation, we can apply it to one point-set and determine the set of correspondences using some proximity criteria. Consequently, if either variable is deemed known, the point matching problem is considered solved. This is the main reason why the point matching problem can represented as a problem using either variable (transformation or correspondence), or both. While it may seem simpler to de£ne the problem using a single variable, we will see that it is not necessarily the case for non-rigid point matching.

An simple example is shown to illustrate a non-rigid point matching problem in Figure 2.1. The desired correspondence and transformation are shown in Figure 2.2.

2.1.2 Why Is the Non-rigid Point Matching Problem Dif£cult ?

A search through the variables' parameter spaces is required in order to obtain a reasonable solution to non-rigid point matching. The search becomes computationally very expensive when the dimensionality of the parameter space is high.

Correspondence is one of such high dimensional variables. Even without bringing outliers into the picture, a simple problem of matching of a set of N points to another set of N points would lead to a total of $N!$ permutations.

[1] They probably should be called two sets of unknown parameters. For simplicity, we used the term "variable" here.

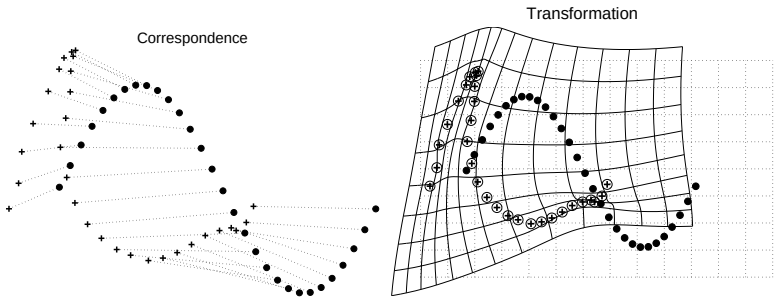

Figure 2.2: The correspondence and the transformation between the two point-sets. The set of correspondences is shown on the left. A link is shown between the correspondending point pairs. The transformation is shown on the right. The transformed template points (shown as big empty circles) now overlap perfectly with the target points. To illustrate the way that the space has been deformed, we also draw a deformed grid (solid curves) and compare it to the original a regular grid (dotted lines).

Any of these permutations can be the possible best solution for correspondence. As the value of N increases, it quickly leads to a combinatorial explosion, making the problem impossible for any simple exhaustive search strategy. Once outliers are included, the situation can be even worse due to increased number of possibilities. Consequently, as we will discuss later in this chapter, the solution for correspondence has alway been carried out with some simplifying assumptions to cut down the search space in all previous methods.

Turning to the transformation, insofar as only rigid mappings are considered, the search often gets restricted to four parameters (rotation, translation and scale) in 2D and nine parameters in 3D. However, we are primarily interested in non-rigid transformations; in this case, the number of parameters grows with the number N of points that are being matched. Further more, the non-rigid mapping parameters are real-valued making the search more dif£cult.

In short, the high dimensionality of the two unknown variables, the non-rigid transformation and the correspondence, composes the main intrinsic[2] factor that makes non-rigid point matching a hard problem. In addition to this intrinsic dif£culty, there are other factors that can be encountered in real-world situations that pose further challenges.

The ideal situation for point matching is when the data sets are "*exactly matchable*". There are two ways of construing that term. First, it means that each point in one set has exactly one counterpart in the other. There are no spurious points (outliers) in either set. The second sense of the term is that there is no noise. Thus, through an

[2]It is "intrinsic" because these variables are indispensable for the non-rigid point matching problem. Other factors such as data quality (e.g. noise) are application dependent and not considerred intrinsic.

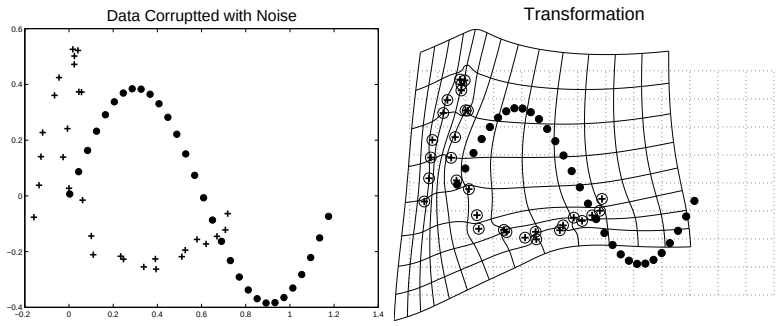

Figure 2.3: Data points can be corrupted due to noise. Noise causes the point locations to be slightly “jittered”. Left: Corruptted data points are shown as dark crosses. Right: When we force the template points to exactly match the noisy data points exactly, the deformation will be unnecessarily distorted.

appropriate non-rigid transformation, we can map a point exactly onto its counterpart. The residue error will be zero in this case. The example shown in Figure 2.1 and 2.2 represents such an ideal case. In real-world situations, since the feature points themselves are obtained through feature detectors from possibly corrupted images, there usually is a certain irreducible amount of point “jitter” and outliers. The feature points are therefore not corrupted and are not “exactly matchable”.

One common problem is noise that arises from the processes of image acquisition and feature extraction. Noise adds jitter (disturbance) to the feature point location. The consequence is that the exact mapping in the ideal point matching scenario no longer holds. To prevent over£tting (£tting not only the data but also the noise), the algorithm needs to £gure out where the true locations of the data points are and conduct the estimation of the non-rigid mapping accordingly. Deciding on the extent to which the data points should be matched is not an easy problem especially when taking into account of the enormous ¤exibility of the non-rigid transformations involved here. We include one example to illustrate the effect of noise here (Figure 2.3).

Another even more dif£cult factor is the presence of outliers — points in one point-set that have no suitable counterparts in the other and hence need to be rejected during the matching process. To be able to handle outliers, the algorithm needs to match a subset of one point-set with an appropriate subset of the other. The size of the subsets is not known is advance. This further increases the already enormous parameter space of the correspondence, making the problem more dif£cult. An illustrative example of non-rigid point matching with outliers is given in Figure 2.4.

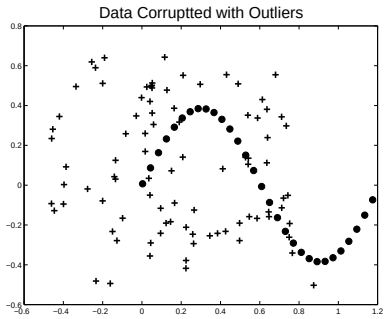

Figure 2.4: Data points can also be corrupted with outliers. We added outlier points to only one point-set so that the effect is clearly discernible. The corruptted points are shown as dark crosses. Clearly, these outliers have to be rejected for the matching to be successful.

2.1.3 A General De£nition For the Non-Rigid Point Matching Problem

Taking these real-world considerations into account, we de£ne a general non-rigid point matching problem to be the following: given two point-sets with only the points' location/coordinate information, we are required to £nd the correspondences between the two point-sets, reject possible outliers and determine a good non-rigid transformation that can map one point-set onto the other, while the mapping should not be adversely affected by the possible existing noise.

2.2 Review of Point Matching Algorithms

As explained above, the two variables, the transformation and the correspondence, in point matching problem are closely related. Once one variable is known, solution for the other is mostly straight forward. Consequently, if either variable can be *independently* determined, the matching problem can be considered solved. There are methods in the literature that are designed to take this kind of independent approach — namely, solve for one of the variables alone without even introducing the other. From our perspective, the intimate prior relationship between the two variables has been used, resulting in one variable dropping out of the formulation.

Other methods utilize this close relationship in a different way by treating point matching as a joint problem with both variables. With both variables in the picture, a simple alternating update process can then be implemented in the following manner: in phase one, one variable is held £xed and the other estimated and in phase two, the preceding sequence is reversed. During the alternation, each variable improved the other. The resulting algorithm is simple and

usually yields good results provided the algorithm converge quickly.

Thus all point matching algorithms can be characterized by examining the way they handle the two variables. Insofar as a method attempts to solve either the correspondence or the transformation alone, it can be regarded as a *independent* estimation approach. On the contrary, if a method tries to £nd the solution for both variables, usually via an alternating update scheme, it can be regarded as a *joint* estimation approach. We use this distinction to orient our description of previous research.

2.2.1 Independent Methods that Solve Only For the Transformation

Typically, these methods are designed for the point matching problem when only rigid transformations are involved. A search in the parameter space of rigid mappings is considered feasible due to the low dimensionality of the mapping. In 2D, a rigid/af£ne transformation has six parameters; and in the case of 3D, nine parameters are involved.

Moment-of-inertia analysis [41] is a classical technique that attempts to solve for the transformation without introducing the notion of correspondence. Originated in physics, the idea is to £nd the center of mass and the principal axis of the data and use them to align the point-sets. The center of mass provides us with the information about the global location of the data and the principal axis with the information of global orientation. While the former can be used to estimate the translation component of the transformation, the later aids in the estimation of the rotation. This method is simple and but usually only provides a rough alignment.

A more sophisticated technique is the Hugh Transform [4, 77]. The transformation parameter space is divided into small bins, where each bin represent a certain con£guration of transformation parameters. The points are allowed to vote for all the bins and the bin which get the most votes is chosen. The answer typically re¤ects the “tyranny of the majority.” The voting procedure tolerates a reasonable amount of noise and outliers.

There are numerous other methods such as tree searches [2, 39], the Hausdauff Distance[46], Geometric Hashing [53, 45] and the alignment method [88] as well.

All these methods work well only for rigid transformations. When it comes to non-rigid mappings, the huge number of the transformation parameters (often proportional to the size of the dataset) usually renders these methods ineffective.

We suspect that this is the main reason for the relative dearth of literature in non-rigid point matching despite the long history of the problem for rigid, af£ne and projective transformations [38, 11].

2.2.2 Independent Methods that Solve Only For the Correspondence

A different approach to the problem is to focus on the correspondence alone. For most problem formulations, it is usually necessary to search over all possible correspondences to guarantee optimality which is infeasible for even moderate problem sizes. Certain measures have to be taken to prune the search space. This is normally achieved by adding additional information into the correspondence estimation process. The extra information typically provides further constraints which cut down the search space. To this end, three different types of additional information can be used.

Use Higher Level Feature Structures

The £rst type of methods, typically referred to as dense feature-based methods, try to organize the feature points into higher level feature structures such as line segments, lines, curves or surfaces through object parameterization [33, 80, 60, 79, 28]. In other words, after the feature segments are extracted, they are grouped together so that matching can be performed at the higher curve or surface level. Furthermore, the curves and/or surfaces are usually smooth which helps the matching process. Grouping is done using a curve/surface £tting step. Very often, a common parameterized coordinate frame (e.g. curve arc length) is used in this step to add local context (e.g. ordering) information. With the additional information acting as a constraint, the dif£culty of searching through the high dimensional correspondence space is largely alleviated.

Most of the curve matching and surface matching methods basically belong to this type. There is a huge amount of literature in the former two £elds, we only cited a few representative ones here.

The relative advantages and disadvantages of this type of methods can be clearly seen in the curve matching case. With the extra curve ordering information available, the correspondence space is drastically reduced making the correspondence problem much easier. Standard optimization approaches such as dynamic programming can then be used to £nd the best correspondence.

However, the requirement of such extra information imposes limitations on these methods. First, an additional curve £tting step is a prerequisite which normally requires good feature extractions. Both feature extraction and curve £tting can be dif£cult when the data is noisy or when the shapes involved are complex. Second, most of these methods rely heavily on shape measures such as curvature to evaluate the similarity between curve segments. These measures can only be reliably de£ned for curves with good geometrical properties (e.g. smoothness and continuity). For this reason, these methods have dif£culty matching curves that are broken (not continuous) or jagged (not smooth). Further more, most of these methods do not handle multiple curves or partially matched curves.

In sum, the introduction of higher level feature structures eases the dif£culty for the search for correspondence. On the other hand, it also imposes limitations on the general applicability of these methods. Such limitations do not exist for the lower level point matching problem. Our analysis also suggests that a non-rigid point matching algorithm can be a powerful tool only if the dif£culty of the search through high dimensional parameter spaces can be overcome.

Use Local/Global Shape Attributes

The second type of methods work with more sparsely distributed point-sets. Shape attributes based on either local or global context are de£ned in these methods. The attributes are then used to determine the correspondence.

Following [70, 72, 20], the modal matching approach in [69] uses a mass and stiffness matrix that is built up from the Gaussian of the distances between any point feature in one set and the other. The eigenvectors of such matrices are ordered according their eigenvalues and are called mode shape vectors. The correspondence is computed by comparing each point's relative participation in the eigen-modes. The basic idea is that while a point-set of a certain *shape* is non-rigidly deforming, different points at different locations should have systematically different ways of movement. Such differences are used to distinguish the points and determine their correspondences.

The examples shown in [69] demonstrate that the method is capable of aligning point-sets which undergo non-rigid deformations. It is clear that the shape eigen-modes do provide enough information to differentiate the points at a global level. However, when examined at a more local and detailed level, their results [69] also indicate the correspondences found are not very accurate. Besides accuracy, another major limitation of these algorithms is that they can not tolerate outliers. Outliers can cause serious changes to the deformation modes which invalidates the resulting correspondences.

Another approach recently proposed in [5, 6] adopts a more probabilistic strategy. A new shape descriptor, called the "shape context", is de£ned for correspondence recovery and shape-based object recognition. For each point chosen, lines are drawn to connect it to all other points. The length as well as the orientation of each line are calculated. The distribution of the length and the orientation for all lines (they are all connected to the £rst point) is estimated through historgramming. This distribution is used as the shape context for the £rst point. Basically, the shape context captures the distribution of the relative positions between the currently chosen point and all other points. The shape context is then used as the shape attributes for the chosen point. The correspondence can then be decided by comparing each point's attributes in one set with the attributes in the other. Since attributes and not relations are compared, the search for correspondence can be conducted much more easily. Or in more technical terms, the

correspondence is obtained by solving a relatively easier bipartite matching problem rather than a more intractable (NP complete) graph matching problem [34]. After the correspondences are obtained, the point set is warped and the method repeated. Designed with the task of object recognition in mind, this method has demonstrated promising performance in matching shape patterns such as hand-written characters. The algorithm's ability to handle complex patterns and large deformations is under investigation.

Use Spatial Relationship Information

The third type of approach involves a different take. By recasting point matching as an inexact weighted graph matching problem, the *spatial relationships* between the points in each set is used to constrain the search for the correspondences.

In [22], such inter-relationships between the points is taken into account by building a graph representation from Delaunay triangulations. The search for correspondence is accomplished using inexact graph matching [73]. However, the spatial mappings are restricted to be af£ne, projective or articulated af£ne [22, 73, 89].

There has been some work done to incorporate non-rigid deformations [1, 54]. In [1], decomposable graphs are used for deformable template matching and the search is conducted using dynamic programming. In [54], a maximum clique approach is used to match relational sulcal graphs. In either case, the choice of graph is rather arbitrary since there is no direct relationship between the deformable model and the graphs that are used. Graph de£nition is also a problem because the graph inter-relationships, attributes and line type information can be very brittle and context dependent. In fact, the graphs in both cases [1, 54] had to be hand designed.

2.2.3 Joint Methods that Solve for Correspondence and Transformation:

Solving for just the correspondence or the transformation in isolation seems rather dif£cult, if not impossible. It would be much easier to estimate the non-rigid transformation once correspondences were given. However, before good correspondences can be estimated, a reasonable transformation is clearly needed. This observation points the way to a joint approach for the point matching problem — alternating estimation of the correspondence and the transformation.

These methods can be further divided into two groups according to the way they treat the correspondence variable.

Methods that Treat the Correspondence as a Binary Variable

Correspondence as conventionally construed denotes a binary variable. A pair of potential counterparts is considered to be either corresponding (hence the value 1), or not (hence the value 0). Earlier methods usually model correspondence in this way.

Iterative closest point (ICP) [7] is one of such methods. It utilizes the nearest-neighbor relationship to assign the correspondence at each step. Only binary values (0 or 1) are allowed so that the correspondences between the closest point pairs are set to 1, while the correspondences for all other pairs are set to 0. Such an estimation of correspondence is then used to re£ne the transformation, and vice visa. It is a very simple and fast algorithm which guarantees to converge to the nearest local minimum. Under the assumption that there are an adequate set of initial poses available, it can also become an global matching tool for rigid transformations.

Unfortunately, such an assumption is no longer valid in the case of non-rigid transformations especially when then deformation is large. The crude way of assigning correspondences also generates a lot of local minima and does not usually guarantee that the correspondences are one-to-one. In spite of its simplicity and speed, it is not a very robust algorithm. Its performance degenerates quickly with noise and, especially with outliers even if some robustness control is added [63, 16, 17].

Methods that Treat the Correspondence as a Continuous Variable

Recognizing this drawback of treating the correspondence as a strictly a binary variable, other algorithms have attempted to relax this constraint and have introduced the notion of "fuzzy" correspondence, namely allowing continuous values between 0 and 1 (e.g. 0.5) for the correspondence.

This kind of correspondence can have different interpretations. It can be thought of as a probability measure of the possible relationship between two points in different point-sets. With this interpretation, point matching can be modeled as a probability estimation problem. However, if we regard the correspondence as an assignment variable, point matching can also be interpreted as a classical optimization problem — the linear assignment problem. These two different interpretations lead to two different approaches.

The work in [96, 22, 42] follows the probabilistic approach. They model point matching as a probability density estimation problem using a Gaussian mixture model. The well known Expectation-Maximization (EM) algorithm is used as the alternating update procedure to search for the solution. The name "EM" come from two alternating steps involved in the algorithm — E step and M step. When applied to a point matching problem, the E step basically estimates the correspondence under the given transformation while the M step updates the transformation based on

the current estimation of the correspondence. In [96], the outliers are handled through an additional uniform distribution with the mixture model. A similar digit recognition algorithm proposed in [42] models handwritten digits as deformable B-splines. Such a B-spline as well as an af£ne transformation were solved to match a pair digits. However, both [96, 22] only solve for rigid transformations. Designed as a character recognition tool, [42] assumes that the data points themselves can be represented by B-splines and did not address the issue of matching arbitrary point patterns. One other common problem shared by all the probabilistic approaches is that they do not enforce one-to-one correspondence.

The work in [36, 63, 16] takes the optimization route. It also results in an alternating update strategy. With a £xed transformation, £nding the optimal correspondence has been treated as a linear assignment problem. The *bene£t measure* required in this assignment problem is determined by the negative of the Euclidean distance between possible matching point pairs after the transformation is applied. A fuzzy correspondence (assignment) is solved by maximizing the joint bene£t. With the estimated correspondence, the transformation can then be updated by solving a least-squares problem. Two novel techniques, namely deterministic annealing and softassign, were used to gradually re£ne the estimations, while enforcing one-to-one correspondence. Although the resulting point matching algorithms demonstrated very nice robustness properties in handling noise and outliers, they were limited to solving rigid or partially non-rigid (e.g. piecewise af£ne) transformations.

2.3 What Can We Do?

After reviewing all the related previous work, we see that the non-rigid point matching has remained largely unresolved. While most of the point matching methods available can only solve for rigid transformations, the few methods that are capable, to various extents, of handling non-rigid transformations have a number of limitations. However, the formulations have left room for further improvement and we can learn from all of them.

Improving upon the independent approaches seems rather dif£cult. With a single variable, the independent approaches inevitably require additional information to accomplish the formidable task of searching in a high dimensional parameter space. Unfortunately, this requirement imposes great limitations for the algorithms themselves. Circumventing these limitations doesn't appear to be straightforward.

The joint approach is more promising. Because of the simplicity and obvious effectiveness of the alternating strategy, the resulting algorithm has much greater potential of being a general framework which, as we earlier argued, is desirable. The introduction of fuzzy correspondence has further demonstrated better robustness properties in handling

noise and outliers [96, 17]. What is lacking is a non-rigid transformation component. The £rst task for this work is to add such a non-rigid component and derive a general algorithm for non-rigid point matching.

Further more, we also observed that the probabilistic approach and the optimization approach, though seemingly originating from quite different domains, share a great deal in common. So a second task for this work is to bridge the gap between these two different approaches and seek a way to combine their merits while attempting to avoid either one's problematic aspects.

Chapter 3

A New Non-rigid Point Matching Algorithm

In this Chapter, we develop our non-rigid point matching algorithm. Since the point matching problem can be treated as a probability density estimation problem or as a joint optimization problem, there is consequently a probabilistic approach and an optimizational approach. We will follow the optimizational approach for the algorithm development in this chapter, mainly because it is more intuitive. The alternative probabilistic approach will be discussed later in the appendix section.

This chapter is organized into £ve parts as we gradually develop the algorithm. We £rst pose point matching as a joint estimation problem of the correspondence and the transformation as motivated from the classical binary linear assignment formulation. In the second part, we point out the limitations of the binary correspondence model and discuss how we can introduce fuzzy correspondence into the formulation with the help of two optimization techniques: soft assign and deterministic annealing. We also explain how the fuzzy assignment formulation can be further improved to better model the outliers and the prior regularization. In the third section, the £nal objective function for non-rigid point matching will be derived and the algorithm will be presented. We then show how different types of transformations can be incorporated into the general framework as independent modules. We £nish this chapter by discussing the similarity and difference between our algorithm and the popular ICP method.

3.1 Point Matching as Joint Estimation of Correspondence and Transformation

Notations

Suppose we have two point-sets V and X (in $\Re^2$or in $\Re^3$) consisting of points $\{v_a, a = 1, 2, \ldots, K\}$ and $\{x_i, i = 1, 2, \ldots, N\}$ respectively. Note that the total number of points can be different so it is possible that K does not equal N. For the sake of simplicity, we will assume for the moment that the points are in 2D. In this chapter, we always apply the transformation to one point-set V so that it can be better aligned with the other point-set X. For this reason, we call point-set V the model and point-set X the target.

We represent the non-rigid transformation by a general function f with parameters α. A point v_a is mapped to a new location $u_a = f(v_a, \alpha)$. The whole transformed point-set V is then U or $\{u_a\}$. We use a *prior smoothness* measure to place appropriate constraints on the mapping to prevent it from behaving too arbitrarily[1]. To this end, we introduce an operator L and our chosen smoothness measure is $||Lf||^2$. We will explain this issue in greater detail later in this Chapter where we discuss more a speci£c non-rigid transformation — thin plate spline. Until then, the non-rigid transformation is kept as an abstract notion f. This way our algorithm will be developed as a general framework suitable for different kinds of non-rigid transformations. After such a framework is developed, different and speci£c non-rigid transformation models can be easily incorporated as independent modules.

A Joint Formulation

As mentioned before, we use an integrated non-rigid mapping and correspondence formulation. Overall, we would like to £nd a good correspondence so that the point-sets can be matched as closely as possible, while a reasonable fraction of the points are rejected as outliers. The correspondence problem can thus be cast as a linear assignment problem [61], which is augmented here to take outliers into account. A similar integrated formulation, but with no accompanying algorithm, was £rst presented in [100] for motion estimation.

We start with a typical linear assignment problem [55]. It can be formulated as the following: given a $N \times N$ bene£t matrix Q with entries q_{ai}, £nd the best $N \times N$ assignment permutation matrix Z consisting of z_{ai} that maximize

[1]Without any kind of restriction on the transformation, we can always map any set of N points onto any other set of N points in $N!$ ways. However, most of these mappings would be too arbitrary to be useful for us. We are usually only interested in mappings which do not lead to outrageous distortions to the space.

the overall bene£t

$$E(Z) = \sum_{a=1}^{N}\sum_{i=1}^{N} z_{ai} q_{ai}.$$

The name "assignment" comes from an intuitive and economic understanding of this problem: one can let the $i's$ label people and the $a's$ label objects, and set q_{ai} to be the bene£t of assigning the a'th object to the i'th person. The problem then consists of £nding a one-to-one correspondence between people and objects such that a maximum overall bene£t can be achieved.

Put in the context of the point matching problem, the assignment variable Z or $\{z_{ai}\}$ clearly plays the same role as the correspondence. We named the matrix Z or $\{z_{ai}\}$ the *correspondence matrix* [36, 65]. It is a augmented version (though we still use the same notation, the dimensionality has been increased to be $(N+1)\times(K+1)$) of the assignment matrix in standard linear assignment problem.

Since our task now is to transform the model set V to get the best £t of the model to the target data, we can design the bene£t matrix to re¤ect such a £tness measure (similarity measure). A simple way is to use the negative of the Euclidean distance between the possible matching point pairs after one point-set is transformed. Since a smaller distance mean a better £t, and hence a larger bene£t, the original maximization problem is converted to a minimization problem.

More formally, given two point-sets V and X consisting of points $\{v_a, a = 1, 2, \ldots, K\}$ and $\{x_i, i = 1, 2, \ldots, N\}$, point matching can be regarded as the minimization of the following *binary* linear assignment—least squares energy function:

$$E_1(Z,\alpha) = \sum_{i=1}^{N}\sum_{a=1}^{K} z_{ai}||x_i - f(v_a,\alpha)||^2 + \lambda||Lf||^2 - \zeta\sum_{i=1}^{N}\sum_{a=1}^{K} z_{ai} \tag{3.1}$$

where the parameters λ and ζ are just constants, and the variable Z or $\{z_{ai}\}$, with only possible binary values 0 or 1, is subject to the following constraints,

$$\sum_{i=1}^{N+1} z_{ai} = 1, \quad \text{for} \quad a \in \{1, 2, \ldots, K\}$$

$$\sum_{a=1}^{K+1} z_{ai} = 1, \quad \text{for} \quad i \in \{1, 2, \ldots, N\}$$

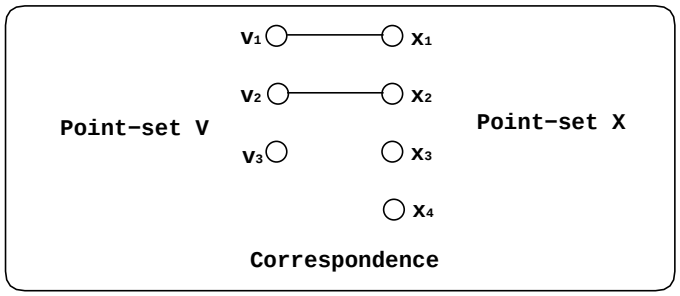

Figure 3.1: The correspondence in a point matching example. The Two point-sets (V of $\{v_1, v_2, v_3\}$ and X of $\{x_1, x_2, x_3, x_4\}$) and their correspondence are shown. Points v_1 and v_2 correspond to $x_1$and x_2 respectively, and the rest of the points are outliers. Each node represent one point symbolically (not its actual location).

The energy function can be easily dissected. The £rst term is essentially a similarity measure to evaluate how good a £t is with a set of speci£c correspondence and transformation. The second term is the prior constraint on the transformation. The third term is the robustness control term preventing rejection of too many points as outliers[2]. The parameters λ and ζ are the weight parameters to balance these terms.

As we mentioned above, the matrix Z or $\{z_{ai}\}$ is called the *correspondence matrix* [36, 65]. It consists of two parts. The inner $K \times N$ part of Z de£nes the correspondence. If a point v_a corresponds to a point x_i, $z_{ai} = 1$. If $z_{ai} = 0$, then the points are unmatched and their distance does not contribute to the total energy. The row and column summation constraints guarantee that the correspondence is one-to-one. The outer extra $(N+1)^{th}$ row and $(K+1)^{th}$ column of Z are introduced to handle the outliers. When a point is rejected as an outlier, the related inner entries all become zero. Then the extra outlier entries will start taking non-zero values to satisfy the constraints. An example of a point matching case and its correspondence matrix are shown in Figure 3.1 and 3.2.

Problems with the Binary Correspondence

Posed in this manner, the point matching objective function in (3.1) consists of two interlocking optimization problems: a linear assignment discrete problem on the correspondence [52] and a least-squares continuous problem on the transformation. Both problems have unique solutions when only one of the variables is unknown. When both the correspondence and the transformation are unknown, this becomes a much more dif£cult problem. That is mainly the

[2]The need for the robustness term can be best understood from the extreme case when all of the points are rejected as outliers. Then all inner entries of m_{ai} become zero. It is obviously a trivial answer. Basically, the addition of this robustness term encourages the match matrix entries m_{ai} to take non-zero values.

z_{ai}	x_1	x_2	x_3	x_4	outlier
v_1	1	0	0	0	0
v_2	0	1	0	0	0
v_3	0	0	0	0	1
outlier	0	0	1	1	

Figure 3.2: The resulting binary correspondence matrix of the example shown in Figure 3.1. Note that the existence of an extra outlier row and outlier column makes it possible for the row and column constraints to always be satis£ed even with outliers.

reason why we resort to the the alternating strategy for the optimization.

However, since such an alternating approach is basically still a grouped descent algorithm, it is still going to be plagued by countless local minima caused by high dimensional correspondence and no-rigid transformation parameters, noises and possible outliers. To make the optimization process robust to these local minima, it turns out that a better model for the correspondence is the key [101, 102, 36, 17].

The limitation of correspondence only being able to take binary values 0 or 1 can cause a lot of dif£culties for the optimization. Though the correspondence is binary, the transformation variable involved in the point matching problem is obviously continuous. With two variables of quite different natures, the energy function is hard to analyze. While the transformation variable can be continuously and gradually improved in the optimization, the binary correspondence variable does not allow intermediate states between the binary values, forcing the update on the correspondence variable to always jump abruptly, and hence to be more vulnerable to be trapped in local minima. Put in a more intuitive way, a binary valued correspondence model makes “hard” judgment all the time: a pair of points either match completely ($z_{ai} = 1$) or they do not match at all ($z_{ai} = 0$). Making this kind of conclusive judgment at each step is not meaningful, especially when the transformation is far away from the optimal solution.

Consequently, we have adopted an alternating algorithm approach where the correspondences are not allowed to approach binary values until the transformation begins to converge to a reasonable solution. This is done with the help of two techniques—softassign and deterministic annealing, which have been previously used to solve such joint optimization problems [36, 65, 16].

m_{ai}	x_1	x_2	x_3	x_4	outlier
v_1	0.90	0.03	0.03	0.03	0.01
v_2	0.03	0.90	0.03	0.03	0.01
v_3	0.03	0.03	0.03	0.03	0.88
outlier	0.04	0.04	0.91	0.91	

Figure 3.3: A fuzzy correspondence matrix of the example shown afore in Figure 3.1. We can tell that points v_1 and v_2 most likely correspond to $x_1$and x_2 respectively, and the rest of the points are probably outliers. Note that the row and column constraints are still satis£ed.

3.2 Introducing Fuzzy Correspondence

Softassign

The basic idea of the softassign [36, 65, 16, 17] is to relax the binary correspondence variable Z to be a continuous valued matrix M in the interval of $[0, 1]$, while still enforcing the row and column constraints.

The continuous nature of the correspondence matrix M basically allows fuzzy, partial matches between the point-sets. The correspondence becomes more like a probability measure. Now, one point does not necessarily just correspond to only one other point; it could have multiple possible matching partners, while maybe preferring the ones which are closer to itself a little bit more. This allow the point to "see" further away to £nd a potentially ideal match. From an optimization point of view, this fuzziness makes the resulting energy function better behaved [102] because the correspondences are able to improve gradually and continuously during the optimization without jumping around in the space of binary permutation matrices (and outliers). In more formal terms, making the correspondence fuzzy smoothes the energy function and gets rid of poor local minima [64].

Though allowed to be fuzzy, the correspondence matrix still has to satisfy the row and column constraints. This can be enforced via the iterative row and column Sinkhorn balancing normalization [75] of the matrix M. An example of fuzzy correspondence is shown in Figure 3.3.

There are different degrees of fuzziness. The extremely fuzzy case happens when all m_{ai} are equal, i.e., any point thinks any point else is probably its partner without any preference at all. At the other end of the spectrum is the least fuzzy case when a point only choose one partner, which is most likely the closest, and disregard the rest. This is the same as the binary correspondence. So in fact the binary correspondence can be regarded as one extreme case of

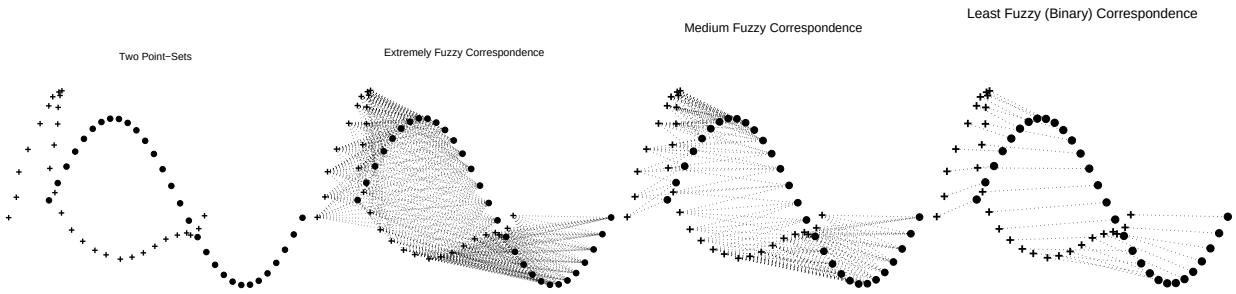

Figure 3.4: Different degrees of fuzzy correspondence. From left to right, i) two point-sets are shown; ii,iii,iv) The degrees of the fuzziness gradually decrease from the extremely fuzzy to the least fuzzy (binary) .

the fuzzy correspondence. Between these two extremes, there can be many intermediate degrees of fuzziness when a point would only choose a portion of possible points (neither all of them, nor a single one) as its partners. The idea of different degrees of fuzziness is illustrated in Figure 3.4.

The concept of different degrees of fuzziness helped us to come up with a novel search strategy. In this search mechanism, each point is always actively searching for its potential partners while the searching is regulated by some "*search range*".

In the extremely fuzzy case, one point is being matched to almost all the points in the other set, even including those that are quite far away from itself. At this stage, any single point search range can be interpreted as being so large that it puts all other possible points within its searching scope, causing the resulting correspondence to be very fuzzy.

In the least fuzzy case where a point only choose the closest partner. It is as if the "search range" has shrunk to so small a value that only the closest candidate can get in.

With an intermediate degree of fuzziness, the search range is neither large enough to accommodate all candidates, nor so small that the point can consider no more than the closest candidate. A scale-space strategy can be easily conceived at this point by manipulating such a search range, thus effectively controlling the degree of fuzziness. A point can start with searching globally over long ranges for its possible partners, then slowly reduce the range to re£ne the search to more and more local scales. The idea is illustrated in Figure 3.5.

Deterministic Annealing

Deterministic annealing (DA) [32, 102, 67, 36, 65, 17] is the perfect optimization technique that can provide us with the needed £ne controls over the fuzziness. It is done by adding an entropy term in the form of $H = T \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \log m_{ai}$

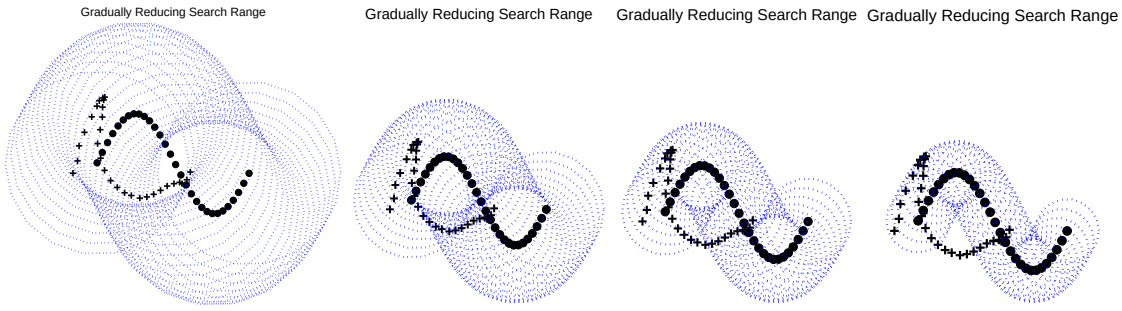

Figure 3.5: A global-to-local search strategy. By gradually reducing the search range, a point can £rst search global for all possible partners and slowly re£ne the matching. Two point-sets are shown in each panel. Circles are placed around the point-set, to which the transformation is being applied (V), to indicate the current scale of search range. From left to right, the search range is gradually decreasing.

to the original assignment energy function (3.1). The newly introduced parameter T is called the temperature parameter.

The name "temperature" originates from the fact that as you gradually reduce T, the energy function is minimized by a process similar to physical annealing. The annealing in DA refers to the procedure of tracking the minimum of the energy function (with the entropy term) while gradually lowering the temperature. In other words, the minima obtained at each temperature are used as initial conditions for the next stage as the temperature is lowered. With the entropy term, the energy function tends to be more convex, enabling the algorithm to overcome many local minima. The in¤uence of the newly added entropy term will gradually decrease as T approaches zero. So it is guaranteed that the minimum tracked with DA will also always be a minimum of the original energy function (but not necessarily the global minimum). Once the annealing schedule for the temperature is speci£ed, the whole process is determined, hence the name deterministic annealing.

We will discuss DA's ability to overcome local minima more carefully in the appendix section. We will also provide a lot more references there. Here we focus on the intuition.

Some intuitive notion of the workings of deterministic annealing can be obtained from observing the effect of the entropy term in the annealing evolution. Since all m_{ai} are smaller than 1, the entropy term is minimized when all m_{ai} are equal, i.e, the correspondence is most fuzzy. At higher temperatures, the entropy term dominates the energy function. The attempt to minimize the energy function forces the correspondence to be more fuzzy and hence becomes a factor in "convexifying" the objective function [102]. As T is lowered, the in¤uence of the entropy decreases and less fuzzy con£gurations of M are allowed. In short, the entropy term always encourages a certain level of fuzziness.

The level is determined by the temperature parameter T. The larger the T, the higher the level of fuzziness. The temperature parameter T is acting as the "search range" discussed just above. The deterministic technique basically allows us to control the "search range", hence control the degree of fuzziness.

When T goes to zero, we will get back to our original energy function. At this point, the correspondence will also approach binary. This is a consequence of the following extension to the well-known Birkhoff-von Neumann theorem [8]: the set of doubly substochastic matrices[3] is the convex hull of the set of permutation matrices and outliers. So it can be ensured that we will always achieve a one-to-one correspondence.

A Fuzzy Assignment-Least Squares Objective Function

We replace the binary correspondence variable z_{ai} with our new fuzzy correspondence m_{ai}, and add in the new entropy term $T\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}\log m_{ai}$. After making these modi£cations, the original binary assignment—least squares problem is converted to the problem of minimizing the following *fuzzy* assignment—least squares energy function [36, 65, 16]:

$$\begin{aligned} E_2(M,\alpha) &= \sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}||x_i - f(v_a,\alpha)||^2 + \lambda||Lf||^2 - \zeta\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai} \\ &\quad + T\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}\log m_{ai}, \end{aligned} \tag{3.2}$$

where again the parameters λ and ζ are just weight constants, and the variable M or $\{m_{ai}\}$, with continuous values between 0 or 1, is subject to the following constraints

$$\sum_{i=1}^{N+1} m_{ai} = 1, \quad \text{for} \quad a \in \{1,2,\ldots,K\}$$
$$\sum_{a=1}^{K+1} m_{ai} = 1, \quad \text{for} \quad i \in \{1,2,\ldots,N\}$$

The energy function in (3.2) can be minimized by an alternating optimization algorithm that successively updates the correspondence parameter M and the transformation parameter α while gradually reducing the temperature T. Such

[3] A doubly stochastic matrix is one whose entries are between $[0,1]$, and each row and each column sum up to 1. A doubly substochastic matrix is similar except that the sums only need to smaller than 1.

an algorithm has proven to be quite successful in the case of rigid point matching [36, 65].

However, there are two issues that motivated us to examine this approach more carefully before we apply it for non-rigid point matching:

i) Even though the robustness parameter ζ that weighs the robustness control term in (3.2) can be interpreted as a prior estimate of the percentage of outliers in both point-sets, there is no clear way to estimate this parameter.

ii) Setting the parameter λ for the prior smoothness term can be dif£cult. Large values of λ greatly limit the range of non-rigidity of the transformation. On the other hand, when the value of λ becomes too small, the transformation can be too ¤exible, making the algorithm unstable. Our experiments show that gradually reducing λ via an annealing schedule (e.g. setting $\lambda = \lambda^{init}T$ where λ^{init} is a constant) leads to much better performance. Annealing on the prior regularization term *seems to be* a rather happy coincidence.

Modi£cations for Outliers and Regularization

Estimation of the robustness parameter is dif£cult essentially because we do not have a direct model for outliers in our assignment formulation, either in (3.1) or in (3.2). As the name of outlier implies, an outlier is a point that does not have any corresponding point in the other set. It only becomes an outlier when it is rejected by all its possible partners. In this sense, the assignment formulation is mainly trying to model this rejection to differentiate outliers from the other points. This is not a direct model, and hence makes the estimation of the related parameter beforehand dif£cult. The alternative is to choose a more direct model for the outliers. To keep the formulation simple, it should also be compatible with the model for other normal points.

We introduce an imaginary *outlier cluster* for this purpose. Each point-set has one such outlier cluster so that now all the outlier *points* in the other set correspond to this outlier *cluster*. We call it a “cluster” (instead of a point) because it allows multiple points to correspond to it, and the total sum of such correspondences can be greater than 1 since there can be more than one outliers in a point-set. With the newly introduced outlier cluster, the outlier points now have something to “match” to, just as the other normal points. A simple illustrative example is given in Figure 3.6.

The outlier cluster’s main duty is to account for any possible outlier points in the other point-set. To make the outlier cluster model compatible with the rest of the points, we need to specify its parameters such as the location accordingly, while still keeping its special purpose in mind. Remember that we talked about the “search range” phenomenon generated by the fuzzy correspondence and how deterministic annealing can control the degree of the fuzzyness by gradually shrinking that “search range”. At high temperature, the search range is large and a single

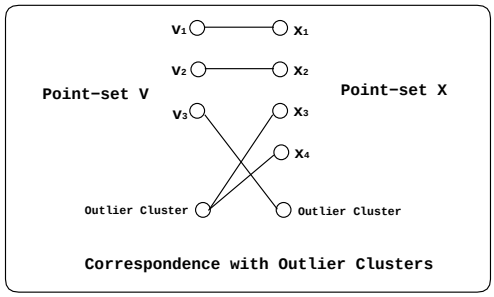

Figure 3.6: Outlier cluster model. Two point-sets (V of $\{v_1, v_2, v_3\}$ and X of $\{x_1, x_2, x_3, x_4\}$) and their correspondence are shown. Points v_1 and v_2 correspond to $x_1$and x_2 respectively, and the rest of the points are outliers. An outlier cluster is introduced into each point-set to account for the outlier points in the other set.

point's partners can be all others. Since we do not have any information about the outliers beforehand, the simplest way is to put the outlier cluster center at the center of its point-set and associate a large search range (speci£ed by temperature T) with it to accommodate any possible outlier points. Each outlier cluster can be transformed as the rest of the points, and correspondence can also be estimated between the cluster and points in the other set. However, while other points follow the annealing process (T decrease), the temperature for the outlie cluster is kept high (a large value T_0).

Informally speaking, the outlier cluster acts like a garbage collector: any point rejected in the matching process is then matched with the outlier cluster. We give a name for the outlier cluster model — "*direct outlier model*". An demonstrative example is shown by Figure 3.7 and 3.8.

We denote the outlier clusters as v_{K+1} and x_{N+1}, and their unique temperature as T_0. We add the outlier model into our formulation and make the following changes to the fuzzy energy function.

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}||x_i - f(v_a,\alpha)||^2 \quad &\rightarrow \quad \sum_{i=1}^{N+1}\sum_{a=1}^{K+1} m_{ai}||x_i - f(v_a,\alpha)||^2 \\
T\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}\log m_{ai} \quad &\rightarrow \quad T\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}\log m_{ai} + T_0\sum_{a=1}^{K} m_{a,N+1}\log m_{a,N+1} \\
&\quad\quad +T_0\sum_{i=1}^{N} m_{K+1,i}\log m_{K+1,i}
\end{aligned}$$

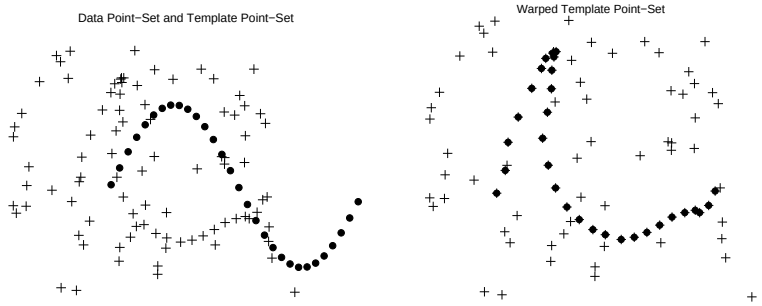

Figure 3.7: Point matching with outliers. Two point-sets (the template point-set V as dark discs, and the data point-set X as crosses) are shown on the left. For demonstration, only the data point-set X has outliers. We need to deform the template point-set so that it is matched with part of the data point-set. The rest of the data point-set should be rejected as outliers. The outlier rejection can be accomplished by the introduction of the outlier cluster (as shown in Figure 3.8).

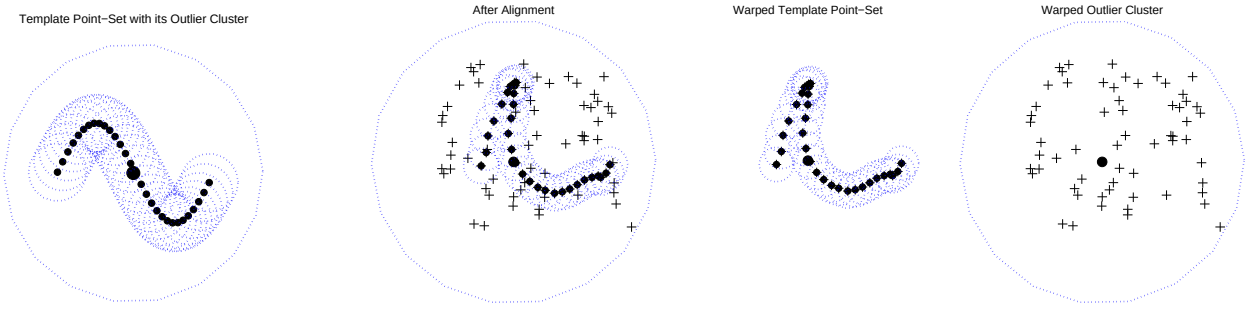

Figure 3.8: Using the outlier cluster to reject outliers. From left to right, i) The augmented template point-set V. Circles (small ones) placed around the template point-set (small discs) indicate the current search range. An outlier cluster is added. The outlier cluster center is shown as a bigger dark disc and its search range is the biggest circle; ii) the optimal matching of the template point-set V to the data point-set X; iii,iv) at the optimal position, the template point-set is matched to the true object within the data point-set. The rest of the data points are matched to the outlier cluster cluster. Note that the larger search range of the outlier cluster allows it to account for all the possible outlier data points.

where v_{K+1} and x_{N+1} are determined by,

$$v_{K+1} = \sum_{a=1}^{K} \frac{v_a}{K}, \text{ and } x_{N+1} = \sum_{i=1}^{N} \frac{x_i}{N}$$

Since now we are directly modeling the outliers, the original robustness term (the indirect model provided by the term $-\zeta \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai}$) is no longer needed and can be dropped out.

The explanation that we have given here for the outlier model is largely intuitive. However,it is not just an arbitrary choice. As we mentioned before, the point matching problem can also be treated via fundamental probabilistic principles. In the probabilistic approach, the outlier model can be naturally derived. That was actually the original inspiration for us to add the outlier model into our optimizational approach. The probabilistic approach also provides the answer for the second question regarding the setting for the regularization as well. It can be derived that the regularization weight parameter λ should actually be in the form of λT. The details of the probabilistic approach is included in the appendix section. Again, here we focus on the intuitive explanation.

The treatment of the regularization prior essentially suggest that the prior term should be "annealed" too. Accordingly, the following modi£cation should be added to the original energy function (3.2):

$$\lambda ||Lf||^2 \ \rightarrow \ \lambda T ||Lf||^2$$

We can gain some insight into the workings of this new version of regularization by examining its effect in the annealing evolution. The transformation is always in¤uenced by the data similarity term $\sum_{ai} m_{ai} ||x_i - f(v_a, \alpha)||^2$ as well as this prior regularization term $\lambda T ||Lf||^2$. The data similarity term causes the transformation to provide the closest alignment, while the prior regularization term trades the alignment with better behavior of the transformation (e.g. more smooth). When the temperature is high, the prior term clearly will dominate and the transformation will be heavily regularized. As the temperature is lowered, the in¤uence of the prior wanes and the transformation is in¤uenced more by the data. At high temperatures, since the algorithm is still working in its early stage and the estimations of the transformation and the correspondence are far from optimal, the information from the data is most likely not very reliable. By emphasizing more the in¤uence of the prior, the algorithm is less likely to be adversely in¤uenced by bad initializations. As the temperature is lowered, the balance is gradually shifted from the prior to the data. With less regularization, the transformation has more freedom so that the alignment is improved faster.

Gradually allowing more freedom for the transformation is an intuitively appealing technique. In fact, this

technique has long been widely, but maybe implicitly, practiced in almost all registration methods. The more complex and more localized alignments (with more freedom) are always initialized by some easier and more global alignments (with less freedom). In order to make sure that the transformation is gradually re£ned, the registration processes are deliberately broken into different stages (global to local) in many of these previous methods, while, as we just have explained, the same purpose can also simply be achieved by annealing the prior regularization parameter. We call the new prior term "*dynamic regularization prior*" since it automatically relaxes the prior constraint in the matching process instead of always holding it at a constant level.

3.3 Robust Point Matching Objective Function and Algorithm

The Robust Point Matching Objective Function

Armed with our direct outlier model and self-relaxing prior model, we modify the fuzzy assignment-least square energy function in (3.2) accordingly. After making these modi£cations, we get the £nal point matching energy function:

$$\begin{aligned} E(M,\alpha) &= \sum_{i=1}^{N+1}\sum_{a=1}^{K+1} m_{ai}||x_i - f(v_a,\alpha)||^2 + \lambda T||Lf||^2 \\ &\quad + T\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}\log m_{ai} \\ &\quad + T_0\sum_{a=1}^{K} m_{a,N+1}\log m_{a,N+1} + T_0\sum_{i=1}^{N} m_{K+1,i}\log m_{K+1,i} \end{aligned} \tag{3.3}$$

where again the parameter λ is a weight constant, and the variable $m_{ai} \in [0,1]$ is subject to the following constraints,

$$\sum_{i=1}^{N+1} m_{ai} = 1, \quad \text{for} \quad a \in \{1,2,\ldots,K\}$$

$$\sum_{a=1}^{K+1} m_{ai} = 1, \quad \text{for} \quad i \in \{1,2,\ldots,N\}$$

The outliers are now directly included through our outlier model, which is provided by the associated parameters x_{N+1}, v_{K+1} and T_0 (note that they are all constants). The regularization prior now has the the temperature parameter T in it.

This new energy function incorporates the notion of fuzzy correspondence, the deterministic annealing technique, a direct outlier model and a self-relaxing regularization prior all together. Embedded in a deterministic scheme,

it will enable us to search globally to locally for the correspondence and gradually re£ne the transformation while rejecting outliers in a robust probabilistic manner. Hence we call the resulting algorithm-"Robust Point Matching Algorithm" (RPM).

The Robust Point Matching Algorithm

The resulting robust point matching algorithm (RPM) is essentially an alternating update process embedded within an annealing scheme.

The alternating update strategy has been used both in probabilistic point matching methods [96, 22, 42] and the optimization point matching methods [36, 63, 16]. In the probabilistic context, this strategy is basically slightly modi£ed version of the very popular EM algorithm, which has long been established as a powerful probability density estimation tool [59, 40]. In the optimizational context, the alternating update strategy can be regarded as a simple grouped descent algorithm. The algorithm is easily to implement. We brie¤y describe the algorithm.

I. The alternating update

The alternating update basically involves differentiating the objective function with respect to the two variables m_{ai} and α, and £nding the answers to set the differentiations to zero.

Step 1, update the correspondence m_{ai} ***given the current transformation parameters*** α.

First calculate,

$$\begin{aligned} q_{ai} &= e^{-\frac{||x_i - f(v_a,\alpha)||^2}{T}}, \quad \text{for the points } a = 1, 2, \dots K \text{ and } i = 1, 2, \dots N, \\ q_{K+1,i} &= e^{-\frac{||x_i - f(v_{K+1},\alpha))||^2}{T_0}}, \text{ for the outlier entries } a = K+1 \text{ and } i = 1, 2, \dots N, \\ q_{a,N+1} &= e^{-\frac{||x_{N+1} - f(v_a,\alpha)||^2}{T_0}}, \text{ for the outlier entries } i = N+1 \text{ and } a = 1, 2, \dots K. \end{aligned} \tag{3.4}$$

Then run the Sinkhorn algorithm (iterated row and column normalization) until convergence is reached,

$$\text{Column Normalization: } m_{ai} = \frac{q_{ai}}{\sum_{b=1}^{K+1} q_{bi}}, \tag{3.5}$$

$$\text{Row Normalization: } m_{ai} = \frac{q_{ai}}{\sum_{j=1}^{N+1} q_{aj}}. \tag{3.6}$$

For the purposes of symmetry breaking, a very small amount of Gaussian noise (mean zero and small standard deviation σ) can be added to m_{ai} after the double normalization.

Step 2: Update the Transformation parameters α ***given the current correspondence*** m_{ai}.

After dropping the terms independent of α, we need to solve the following least-squares problem,

$$\min_{\alpha} E(\alpha) = \min_{\alpha} \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} ||x_i - f(v_a, \alpha)||^2 + \lambda T ||Lf||^2. \tag{3.7}$$

Including the outlier points in the least squares formulation is very cumbersome. We implemented a slightly simpler form as shown below

$$\min_{\alpha} E(\alpha) = \min_{\alpha} \sum_{a=1}^{K} ||y_a - f(v_a, \alpha)||^2 + \lambda T ||Lf||^2 \tag{3.8}$$

where

$$y_a = \frac{\sum_{i=1}^{N} m_{ai} x_i}{\sum_{i=1}^{N} m_{ai}} \tag{3.9}$$

Note that we are always trying to transform the point-set V to match onto the other target point-set X. Once a correspondence is given, the variable y_a can be regarded as our newly estimated target positions. Now between the point-sets of V and Y, the correspondence is known: each v_a corresponds to the point in Y with the same index a, i.e., y_a. Since the correspondence is known, solving for the transformation is basically a normal interpolation problem. In our case, since the Euclidean distance is used, it becomes a least-squares problem.

This simpli£cation in (3.8) is exactly equivalent to the original form in (3.7) if the problem at hand is a template matching problem, i.e., we try to £nd a subset of the data point-set Y and align it with a given template point-set X. In other words, every point in the template set X should have a corresponding point in the data set Y, but not necessarily so when reversed. For the more general cases where both sides might have outliers, the simpli£cation is not equivalent to its original form. Extra bookkeeping is needed to check for outliers (if $\sum_{i=1}^{N} m_{ai}$ is too small) and eliminate them from the least-squares problem.

Either way, this second step will eventually come down to a least-squares problem for the transformation parameters α. The solution depends on the speci£c form of the non-rigid transformation. We will discuss the incor-

poration and the solution for two splines — Gaussian Radial Basis Function Spline and Thin-Plate Spline later in this thesis.

II. Annealing

An annealing scheme (for the temperature parameter T) controls the dual update process. Starting at $T_{init} = T_0$, the temperature parameter T is gradually reduced according to a linear annealing schedule, $T^{new} = T^{old} \cdot r$ (r is called the annealing rate). The alternating updates are repeated until convergence at each temperature. The parameter T is then lowered and the process is repeated until some £nal temperature T_{final} is reached.

Thus to setup an annealing schedule, there are three parameters we need to specify: the starting temperature T_0, the £nal temperature T_{final} and the annealing rate r. They can be set up according to some intuitions. The general rule is that T_0 should be large enough for the initial long range search, T_{final} should be small enough for the £nal detailed match and r should be close to (but smaller than) 1 to make the annealing process slow enough.

The parameter T_0 is set to be the largest squared distance of all point pairs. In this way, it provides large enough search range for all possibly matching point pairs.

We can always set T_{final} to be a very small value close to zero. However, there are reasons that it is neither necessary nor optimal. Once the temperature is low enough, the fuzzy correspondence will approach an unique binary limit. Normally such a binary limit can be determined (and the algorithm will have converged) much long before the temperature reaches zero. So dropping the temperature to zero is unnecessary in most cases. It is not the optimal answer in many cases either. Since the data is often quite noisy, matching them exactly to get binary one-to-one correspondences causes over£tting to the noise. Ideally, the parameter T_{final} should be set according to the amount of noise within the data set. In this work, we make a simpli£ed treatment to set T_{final} to be equal to the average of the squared distance between the nearest neighbors within the set of points which are being deformed. The interpretation is that at T_{final}, the search scopes for all the points will then barely overlap with one another. A fuzzy (though close to binary) correspondence will be found at this £nal temperature. If the user does need a binary one-to-one correspondence, such a binary correspondence can always to be calculated simply by examining the largest correspondence values in each row and each column.

We usually set r to be 0.93 (normally between $[0.9, 0.99\ ]$) so that the annealing process is slow enough for the algorithm to be robust, and yet not too slow. For the outlier cluster, the temperature is not annealed and always kept at T_0.

III. The RPM algorithm pseudo-code

We normally starts the algorithm's alternating update process by setting the transformation parameters α to be zeros (so that the transformation is an identity transformation and points stay at their original place). Then we run the correspondence update and the transformation update while gradually lower the temperature.

The General RPM Algorithm Pseudo-code:

Initialize parameters $T = T_0$ and λ.

Initialize parameters α (or M).

Begin A: Deterministic Annealing.

  Begin B: Alternating Update.

  Step I: Update correspondence parameter M based on current α.

  Step II: Update transformation parameter α based on current M.

  End B

Decrease $T \leftarrow T \cdot r$ until T_{final} is reached.

End A

3.4 Incorporation of Different Transformations

Different models of transformation have their different properties and, hence are suitable for different applications. An algorithm's ability to accommodate different transformation models can make it a general tool for many problems. Our algorithm is designed with this in mind. At this point, the RPM algorithm is a general framework:, i.e., any speci£c non-rigid transformation can be put in to replace the general notion of f. The only module that needs to be modi£ed is the solution in the second update step for the least-squares problem. As an example, here we discuss the incorporation of one speci£c non-rigid transformation parameterization — thin plate spline (TPS) [10, 94] .

TPS also provides us with an good example to explain the reason why we need to include a prior smoothing term for a non-rigid transformation model. As most other splines, TPS's main task is to generate a appropriate spatial mapping given two sets of landmark points. The constraint that these two sets of landmark points are to be mapped exactly onto each other is not strong enough to specify a unique non-rigid transformation. If we think of a non-rigid spatial mapping/transformation as a spatial displacement £eld, such a requirement only tells us the displacements at certain particular locations, where those landmarks happen to be at. Apart from those special locations, we need to £ll in what the displacements are for the rest of the space. Since the mapping is allowed to be non-rigid, there are an in£nite number of different choices. To narrow down the choices, we need to add extra information about how the non-rigid mappings should behave. More often than not, we would prefer the mapping to be somewhat "smooth", i.e.,

it should not distort the space too much, especially if it is unnecessary. This kind of preference can be enforced via a prior penalty term that discourages mappings that are too arbitrary. Since the penalty is often measure by some kind of "smoothness" of the mapping, we call it the "smoothness prior/constraint/measure". We can control the behavior of the mapping by choosing a speci£c smoothness measure, which basically re¤ects our prior knowledge on the transformation.

One of the simplest form of the the smoothness measures is the space integral of the square of the second order derivatives of the mapping function. More formally, we are looking for a mapping function $f(v_a)$[4] between corresponding point-sets $\{y_a\}$ and $\{v_a\}$ that minimizes the following energy function:

$$E_{TPS}(f) = \sum_{a=1}^{K} ||y_a - f(v_a)||^2 + \lambda \int\int [(\frac{\partial^2 f}{\partial x^2})^2 + 2(\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2] dx dy \tag{3.10}$$

where λ is a weight constant. With a £xed weight parameter λ, there exists a unique minimizer f. We call it the thin plate spline (TPS).

TPS can be parameterized by two matrices d and c, ($\alpha = \{d, c\}$). When applied to a point v_a, the TPS function warps it to a new location $f(v_a, \alpha)$.

$$f(v_a, \alpha) = f(v_a, d, c) = v_a \cdot d + \phi(v_a) \cdot c \tag{3.11}$$

where d is a $(D+1) \times (D+1)$ matrix representing the af£ne transformation and, c is a $K \times (D+1)$ warping coef£cient matrix representing the non-af£ne deformation. We use *homogeneous* coordinates for the point-set where a point v_a is represented as a vector $(1, v_{ax}, v_{ay})$. This way we can have an single af£ne transformation matrix d with the translation components integrated into it. The vector $\phi(v_a)$ is a $1 \times K$ vector for each point v_a, where each entry $\phi_b(v_a) = ||v_b - v_a||^2 \log ||v_b - v_a||$. Loosely speaking, the TPS kernel contains the information about the point-set's internal structural relationships. When it is combined with the warping coef£cients w, a non-rigid warping is generated.

The detailed derivation of the solution of TPS (3.11) from the energy function (3.10) is included in the appendix. Here we would like to just point out that it is a closed form analytic solution since the energy function (3.10) is of the least-squares form. All we need to do is to put the solution for TPS in the RPM algorithm (alternating

[4]In the case of 2D, the mapping function f is composed of two components (f_x, f_y) where f_x and f_y are x component and y components of the mapping. The notations like $\frac{\partial^2 f}{\partial x^2}$ are in fact de£ned for both f_x and f_y. Please also note that here x, y are coordinates, not points.

update step II) and we will have a RPM algorithm that is capable of matching two point-sets with TPS. We will refer to such a speci£c implementation of our RPM algorithm with TPS as TPS-RPM from now on.

The TPS-RPM Algorithm Pseudo-code:

Initialize parameters $T = T_0$ and λ.

Initialize TPS parameters (c, d) (e.g., $c = 0$ and $d =$ Identity matrix).

Begin A: Deterministic Annealing.

Begin B: Alternating Update.

Step I: Update correspondence parameter M based on current (d, c).

Step II: Update TPS parameter (d, c) based on current M.

End B

Decrease $T \leftarrow T \cdot r$ until T_{final} is reached.

End A

A nice property of the TPS is that it can always be decomposed into a global af£ne (speci£ed by d) and a local non-af£ne component (speci£ed by c). Consequently, the TPS smoothness term in (3.10) is solely dependent on the non-af£ne component. This is a desirable property, especially when compared to other splines, since the global pose parameters included in the af£ne transformation are not penalized.

Other splines, or other non-rigid mapping parameterizations, have their own different properties. The choice of a certain non-rigid transformation model depends on the problem at hand. For example, if the registration problem is trying to model real elastic materials, properties such as diffeo-morphism are often deemed important. Hence a non-rigid transformation that guarantees diffeo-morphism [49, 14] might be more appropriate for those problems (than TPS). Our general RPM algorithm, again, still applies. With a little modi£cation to put in the new transformation model, we will get a matching algorithm that is capable of solving the chosen non-rigid transformation. Such an algorithm will also always enjoys all the bene£ts inherited from the general framework such as robustness to local minimum and tolerance of noise and outliers.

3.5 Relationship to ICP

The iterated closest point (ICP) algorithm [7] is actually quite similar to our algorithm. As we discussed in our review section, the main difference is that ICP depends on the nearest-neighbor heuristic and always returns a binary correspondence, which is not guaranteed to be one-to-one. However, it is close to the limiting case of our algorithm, obtained when annealing is replaced with quenching (by starting T close to zero). In that case, the fuzzy correspon-

dence will always be close to binary. Since T can be regarded as a search range parameter, it is not hard to see that the over-simpli£ed treatment of correspondence in ICP makes it much more vulnerable to local minima.

A second difference is that ICP relies on other heuristics to handle outliers. For example, outliers are rejected through a dynamic thresholding mechanism [28]. Distances between nearest point pairs are £rst £t to a Gaussian distribution. Points with distance measure values larger than the mean plus L (usually set to 3) times the standard deviation are then rejected as outliers. We implemented this improved version of ICP as a benchmark comparison for our algorithm. The pseudo-code for ICP is included below.

The ICP Algorithm Pseudo-code:

Initialize TPS parameters α (e.g., $\alpha = 0$, Identity matrix).

Begin A: Iterative Matching.

Begin B: Alternating Update.

Step I: Update correspondence parameter M based on current transformation parameter α based on the closest neighbor criteria (with outlier rejection).

Step II: Update transformation parameter α based on current correspondence parameter M.

End B

Until convergence or maximum number of iterations is reached.

End A

Chapter 4

Examples and Experimental Results

4.1 Introduction

In this section, we test our RPM algorithm on both synthetic and real non-rigid point matching examples. These examples will serve two purposes. The first is to provide a careful evaluation of the algorithm and a comparison to previous algorithms such as ICP. Secondly, by observing the algorithm's matching process during the annealing, we can also get a better understanding of the algorithm's mechanism.

We begin with a few demonstrative simple examples. Through these simple but representative examples, the workings of the ideas, such as fuzzy correspondence and deterministic annealing, that were discussed mostly in an intuitive but abstract manner in the previous chapter will be vividly demonstrated. We will also show how the presence of a few outliers can adversely affect ICP whereas RPM is relatively unaffected.

We then set out to evaluate the algorithm's performance under various circumstances using synthetic data. The ground truth (for both the correspondence and the transformation) is known for the synthetic data so that we can carry out quantitative evaluations of the algorithm. We conducted 3200 synthetic experiments covering two templates, separate outlier, noise, and deformation studies and comparisons with ICP. We present some typical examples from these synthetic experiments and, the error statistics (mean and variance) as a measure of evaluation of RPM and ICP.

After these synthetic experiments, we conduct more experiments on real examples where the ground truth is unknown. Three examples are included here. To investigate RPM's ability to determine meaningful correspondences under large deformations, we studied the matching of a sequence of progressively deformed caterpillar like hand drawings. We will also look at two potential applications for non-rigid point matching. One is the 2D face matching.

The other is the 3D sulcal point matching in the brain registration £eld.

We choose the TPS-RPM for the experiments in this chapter. The implementation of thin plate spline (TPS) has been discussed in the previous chapter. The TPS-RPM algorithm follows the same setup as the general RPM algorithm. The main issues involved such as the dual update and the annealing scheme has also been discussed before. The rest of the algorithm is set up in the following manner. The parameter λ is set to 1. We found that this value works very well for the cases where the points are lying within a unit square (their coordinates are all between $[0, 1]$). To make it generally applicable for all the examples, we just need to re-scale all the points so that they are within the unit square. The alternating update of the correspondence M and the transformation (d, c) is repeated 5 times after which they are usually close to convergence. We then decrease the temperature T and repeat the alternating update process. The implementations of the thin plate spline in 2D and in 3D are almost identical except that in 3D, a different kernel of the form $\phi_b(v_a) = ||v_b - v_a||$ is used [94].

4.2 A Simple 2D Example

We start with a simple case in 2D non-rigid point matching. Two point-sets are shown in Figure 4.1. Neither the correspondence nor the transformation is known. We are attempting to warp one point set V onto the other point-set X with the help of our algorithm. The matching process, which is shown in Figure 4.2, demonstrates a very interesting scale-space like behavior.

In the beginning, at a very high temperature T, the correspondence M is almost uniform. Then the estimated corresponding point-set $\{y_a = \sum_{i=1}^{N} m_{ai} x_i / \sum_{i=1}^{N} m_{ai}\}$ is essentially very close to the center of mass of X. This helps us recover most of the translation needed to align the two point-sets. This can be seen from the £rst row in Figure 4.2.

With slightly lowered temperatures as shown in the second row of Figure 4.2, the algorithm starts behaving very much like a principal axis method [41]. Points in V are rotated and aligned with the major axis along which the points in X are most densely distributed.

Lowering the temperature even more, we observe that the algorithm £nds more localized correspondences which makes it possible to capture the more detailed structures within the target point-set. Progressively more re£ned matching is shown in the third, the forth and the £fth rows. The last row shows that the algorithm almost converges to a nearly binary correspondence at a very low temperature and the correct non-rigid transformation is fully recovered.

The algorithm is clearly attempting to solve the matching problem using an emergent, coarse to £ne strategy.

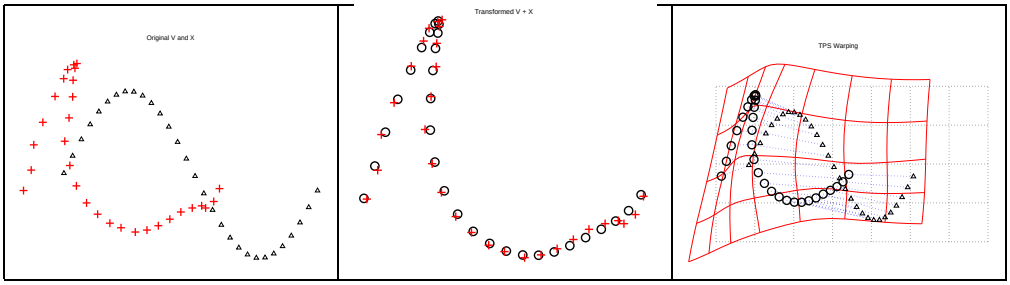

Figure 4.1: A simple 2D example**. Left:** Initial position of two point-sets V (triangles) and X (crosses). **Middle:** Final position. U (transformed V) and X (crosses). **Right:** Deformation of the space is shown by comparing the original regular grid (lighter dotted lines) and its transformed version (darker lines).

Global structures such as the center of mass and principal axis are £rst matched at high temperature, followed by the non-rigid matching of local structures at lower temperatures. It is very interesting that during annealing, this whole process occurs seamlessly and implicitly.

The matching process also shows very close resemblance to a data clustering process. If we regard the point-set V (the point-set being deformed) as a set of template cluster centers and the temperature T as the size of these clusters, what we are doing is basically trying to deform the template cluster model to £t the data (the other point-set X). The £tting is gradually being improved during the process by carefully controlling the cluster size via annealing. Rough £ttings are achieved £rst when the cluster size is large. They are then slowly re£ned to get more accurate £ttings as we reduce the cluster size. This interpretation is not accidental. It is actually an inevitable result if we take the different probabilistic approach for the non-rigid point matching. This intuition will be useful later for us to tune the RPM algorithm for more complicated problems in brain registration where thousands to point features need to be matched.

4.3 A Simple Comparison Between RPM and ICP

ICP's treatment of the correspondence as a binary variable determined by the nearest neighbor heuristic leads to signi£cantly poorer performance when compared to RPM. The difference is most evident when outliers are involved. We have come across numerous examples where a small percentage of outliers would cause ICP to fail.

To demonstrate this point, we tested both ICP and RPM on the same example as above with a single, extra outlier. Since we used an annealing schedule for the regularization parameters λ in RPM, we also used it in ICP. The results are shown in Figure 4.3.

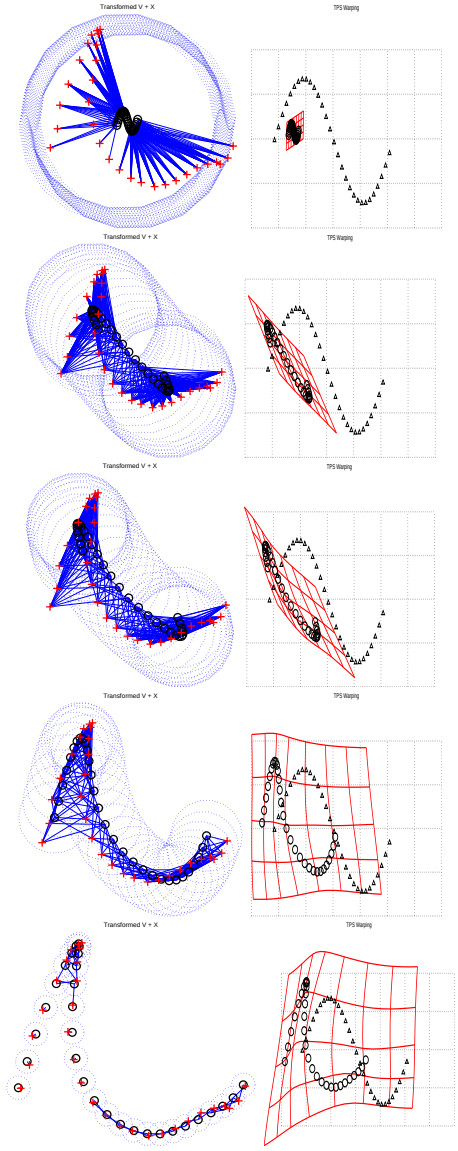

Figure 4.2: The RPM matching process of the simple 2D example. Each row shows the state of the algorithm at a certain temperature. **Left Column:** Current correspondence between U (transformed V, circles) and X (crosses). The most signi£cant correspondences ($m_{ai} > \frac{1}{K}$) are shown as solid links. A dotted circle of radius $\sqrt{T}$ is drawn around each point in U to show the annealing process. **Right Column:** Deformation of the space. Again dotted regular grid with the solid deformed grid. Original V (triangles) and U (transformed V , circles).

Even though there is only one outlier, ICP gets trapped and could not recover. The answer is obviously sub-optimal. We see that RPM also gets affected by the outlier in the middle of the matching process. However, the correspondence is kept fuzzy so that the right correspondence are not completely rejected right from the beginning. This gives the algorithm a chance to eventually break free from the outlier and converge to a better answer.

4.4 Experiments Based on Synthetic Data

Setup of the Synthetic Experiments

To test RPM's performance, we ran 3200 experiments on synthetic data with different degrees of warping, different amounts of noise and different amounts of outliers and conducted comparisons with ICP. The synthetic experiments are set up in the following manner.

We £rst choose a template point-set. Two different templates used here are shown in Figure 4.4. The £rst template comes from the outer contour of a tropical £sh. It mainly consists of a closed contour with some very sharp corners. To de£ne a continuous curve, those sharp corner may pose some challenges because geometrical measures such the curvature are discontinuous at those locations. However, since we represent the image as discrete points, it doesn't affect us at all. The second comes from a Chinese character (blessing). It is a more complex pattern with multiple curve-like strokes that may or may not be connected to each other. Again, modeling such a complex pattern with high level feature representations, such as curve, might be quite messy while the simple point representation can avoid these problems all at once.

After a *template* point-set is chosen, we apply a randomly generated non-rigid transformation to warp it. Then we add noise or outliers to the warped point-set to get a new *target* point-set. To avoid unfair biases, we use a different non-rigid mapping, namely Gaussian radial basis functions (GRBF) [100, 94] for the random, non-rigid transformation instead of TPS. The details of GRBF are included in the appendix section.

The coef£cients of the GRBF were sampled from a Gaussian distribution with a zero mean and a standard deviation s_1. Increasing the value of s_1 generates more widely distributed GRBF coef£cients and hence leads to generally larger deformation. A certain percentage of outliers and random noise are added to the warped template to generate the target point-set. We then used both ICP and RPM to £nd the best TPS mappings that warp the template set onto the target set. The errors are computed as the mean squared distance between the warped template using the TPS found by the algorithms and the warped template using the ground truth Gaussian GRBF. This setup is illustrated by one example in Figure 4.5.

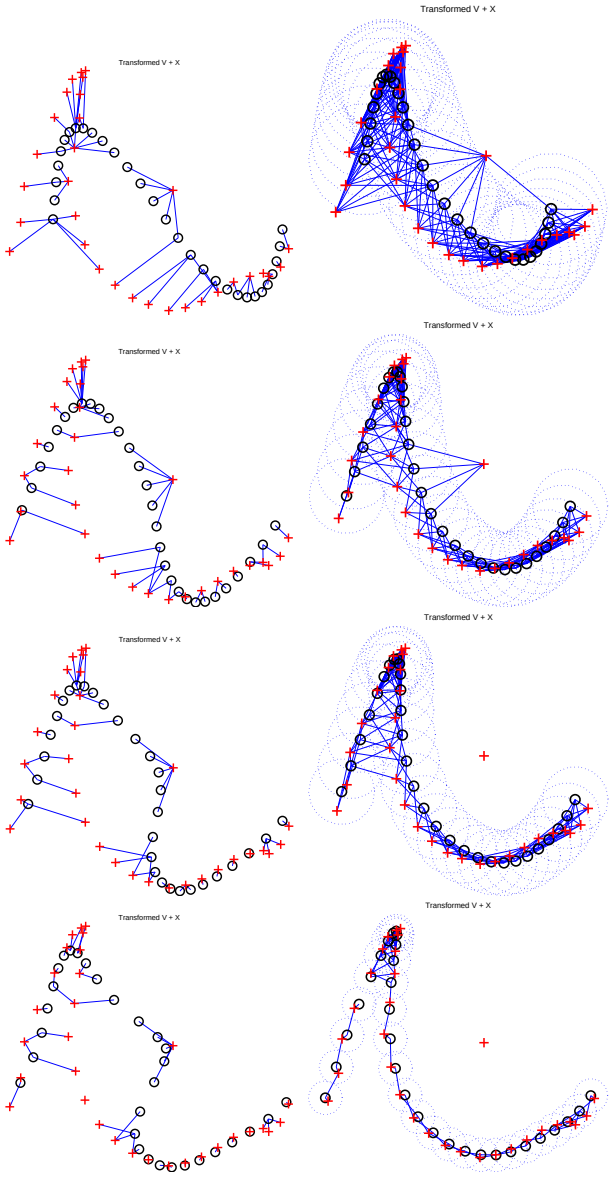

Figure 4.3: An example with one outlier. **Left Column:** Sequential intermediate states of ICP. **Right Column:** Sequential intermediate states of RPM.

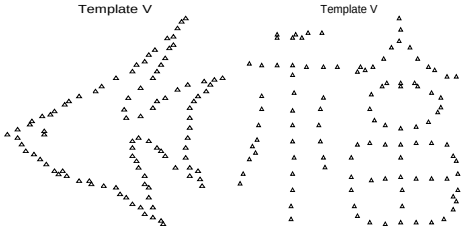

Figure 4.4: Template point-sets for the synthetic experiments.

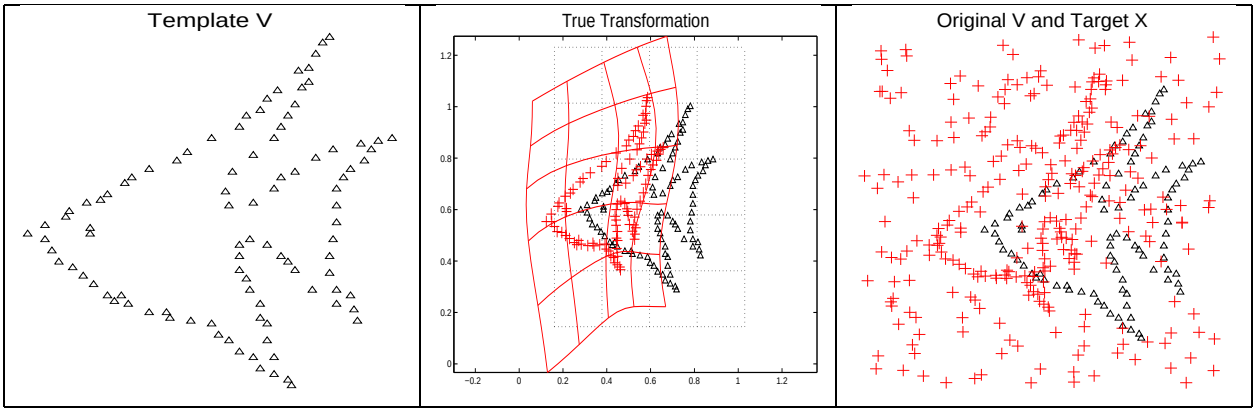

Figure 4.5: Synthetic data generation. **Left:** The template point-set V (triangles). **Middle:** a random GRBF transformation is applied to warp the template point-set. The warped template point-set $V'$are shown as crosses. The warped template point-set is serving as the ground truth. **Right:** Noise and outliers are then added to the warped template data to get the target point-set X. The matching is then done between the template point-set V and the target point-set X. Note that although we know the ground truth correspondence between V and X, such information is not being used in the matching. We intend to recover both the correspondence as well as the non-rigid transformation through our point matching algorithm.

We conducted three series of experiments. In the £rst series of experiments, the template was warped through progressively larger degrees of non-rigid warping. The warped templates were used as the target data without adding noise or outliers. The purpose is to test the algorithms' performance when solving different degrees of deformations. In the second series, different amounts of Gaussian noise (standard deviation s_2 from 0 to 0.05) were added to the warped template to get the target data. A medium degree of warping was used to warp the template. The purpose is to test the algorithms' tolerance of noise. In the third series, different amounts of random outliers (outlier to original data ratio s_3 ranging from 0 to 2) were added to the warped template. Again, a medium degree of warping was used. Some examples of each of these three series are shown in Figure 4.6.

A total of 100 random experiments were repeated for each setting within each series. All three series of experiments were run on both templates.

Results of the Synthetic Experiments

After we explained how the synthetic experiments are set up, we now examine the results by looking at some examples and the error statistics.

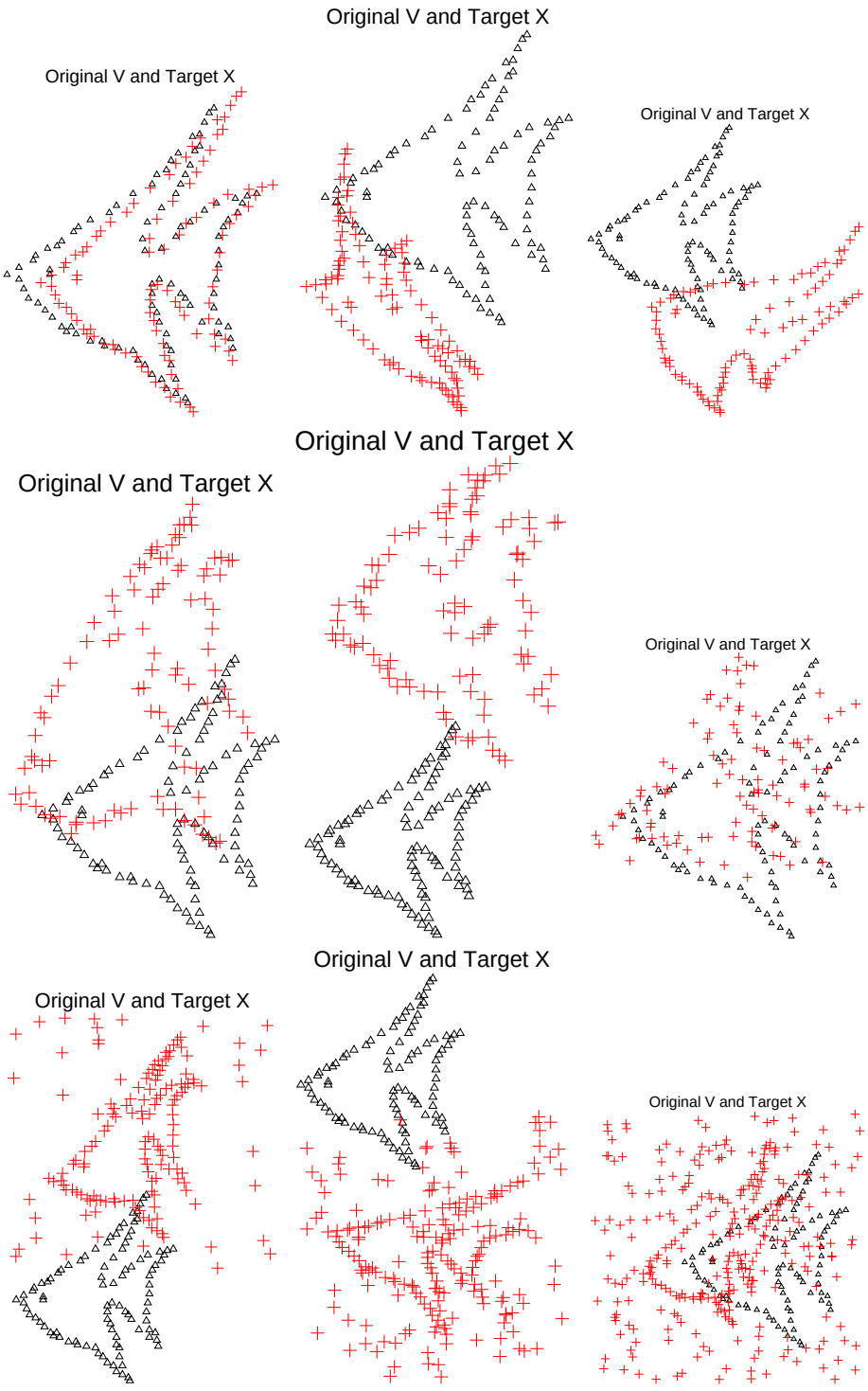

Figure 4.6: Three series of experiments. Each row shows 3 examples from one series. **Top Row**: examples with different degrees of deformation. **Middle Row**: examples with different degrees of noise. **Bottom Row**: examples with different degrees of outliers.

We include some experiments in each series for both templates here as examples. Such examples of the deformation series, the noise series and the outliers series for the £sh template are shown in Figures 4.7, 4.8 and 4.9. Similar examples for the Chinese character template are shown in Figures 4.10, 4.11 and 4.12. For each example, the results from both ICP and RPM are included for comparison. The actual error values (as we discussed above) are also shown. Normally, an error beyond 0.05 indicate a poor matching.

Both RPM and ICP work well for the simpler cases when the degree of deformation is low and the amount of noise and/or outliers is also small. However, in the more dif£cult cases, there is a big performance difference between RPM and ICP. We brie¤y discuss such difference for each series of experiments below.

Increasing the degree of deformations leads to larger shape difference between the template and the target data. It tests an algorithm's ability to track correspondences and large deformations over long ranges. From the examples shown in Figures 4.7 and 4.10, we see that ICP can get easily distracted and trapped by bad matches while the correspondences recovered by RPM are much better. The advantage of RPM's global to local search strategy is clearly demonstrated.

The presence of noise makes the locations of the points less reliable. Visually, the patterns look "blurred". The freedom of fuzzy, partial correspondences inside RPM perfectly compensate for this kind of blurring effect. From the examples in Figures 4.8 and 4.11, we see that the results from RPM always approximate the true underneath shape quite nicely. ICP, on the other hand, starts to fail completely once the noise level become relatively high.

In addition to large deformations and heavy noise, the possible existence of outliers further greatly complicate the point matching problem. In order to achieve a good match, the algorithm has to evaluate the evidence and reject a fraction of the points as outliers. There is a delicate balance in this rejection process. Rejection of too few points, too many points, or the wrong points are all bound to cause poor matches. With the extreme ¤exibility of non-rigid transformations, this balance can be easily upset causing the algorithm to be trapped in a wrong match. From the examples shown in Figures 4.9 and 4.12, we see that ICP is easily confused by outliers. The seemingly reasonable outlier rejection mechanism inside ICP obviously is not suf£cient. Again, RPM shows its robustness to large amount of outliers. Even for the cases where the majority of the target data points are outliers, all the results of RPM are almost optimal.

The £nal statistics (error means and standard deviations for each setting) are shown in Figure 4.13. It gives us an overall picture of both algorithm's performance. ICP's performance deteriorates much faster when the examples become harder due to any of these three factors—degree of deformation, amount of noise or amount of outliers. The difference is most evident in the case of outliers. ICP starts failing in most of the experiments and gives large errors

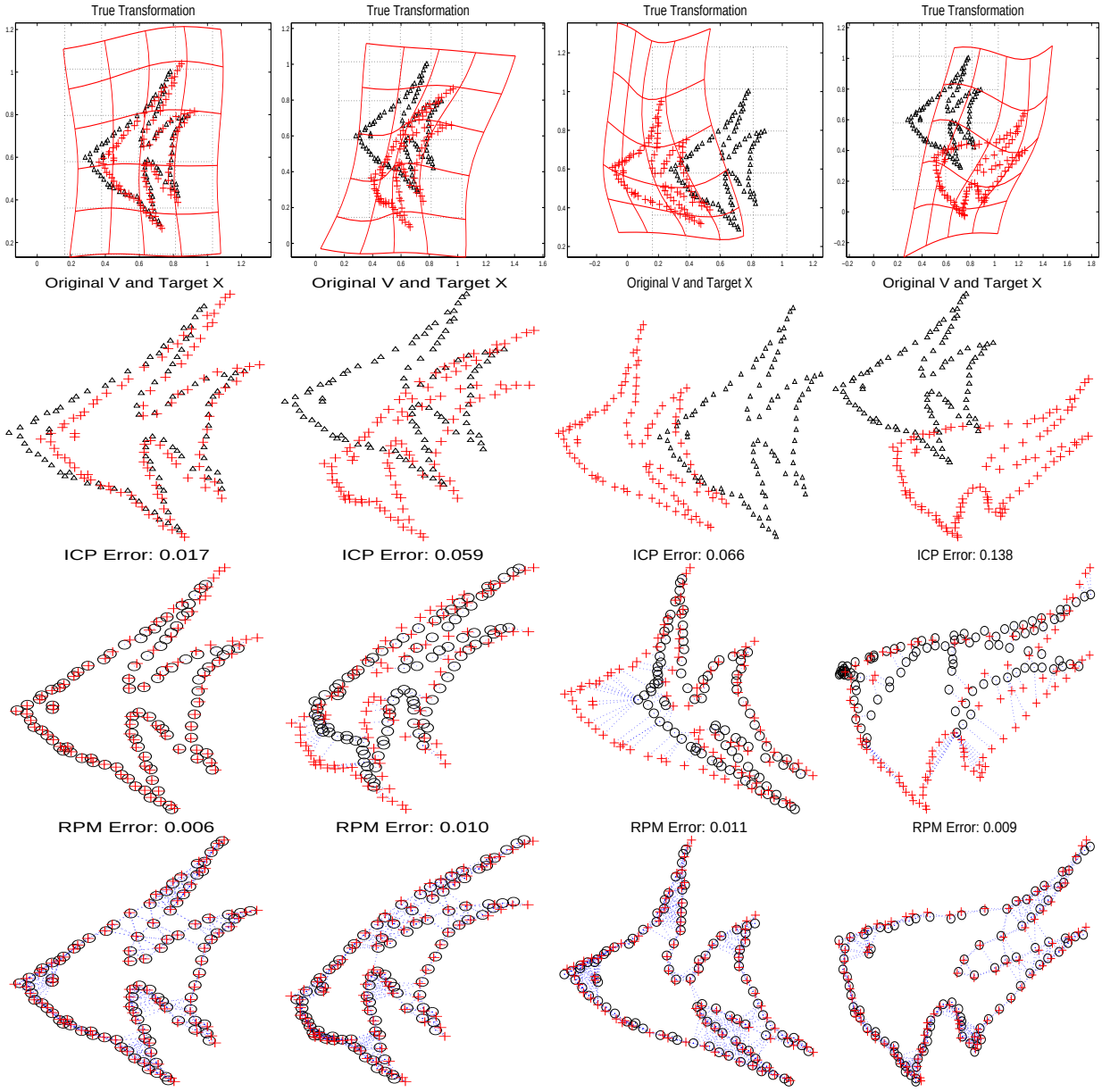

Figure 4.7: Experiments on deformation. Each column represents one example. Four examples are shown. From left to right, increasing degrees of deformation. **First Row:** Warped template. **Second Row:** Template and target (same as the warped template here). **Third Row:** ICP results. **Forth Row:** RPM results.

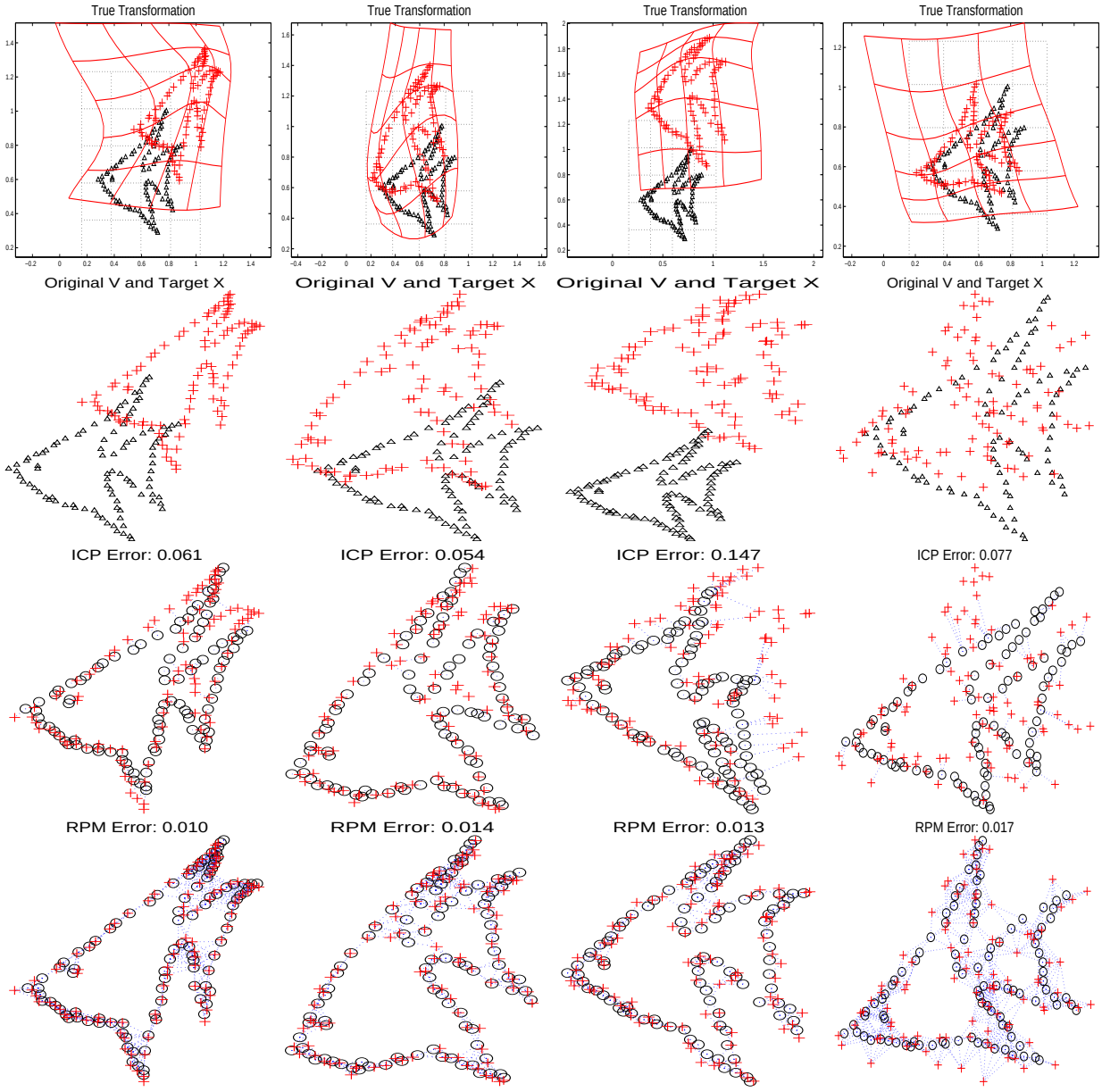

Figure 4.8: Experiments on noise. From left to right, increasing amounts of noise. **Top Row:** Warped template. **Second Row:** Template and target (warped template with noise added). **Third Row:** ICP results. **Bottom Row:** RPM results. The actual error values are also shown. Normally, errors beyond 0.05 indicate poor matching.

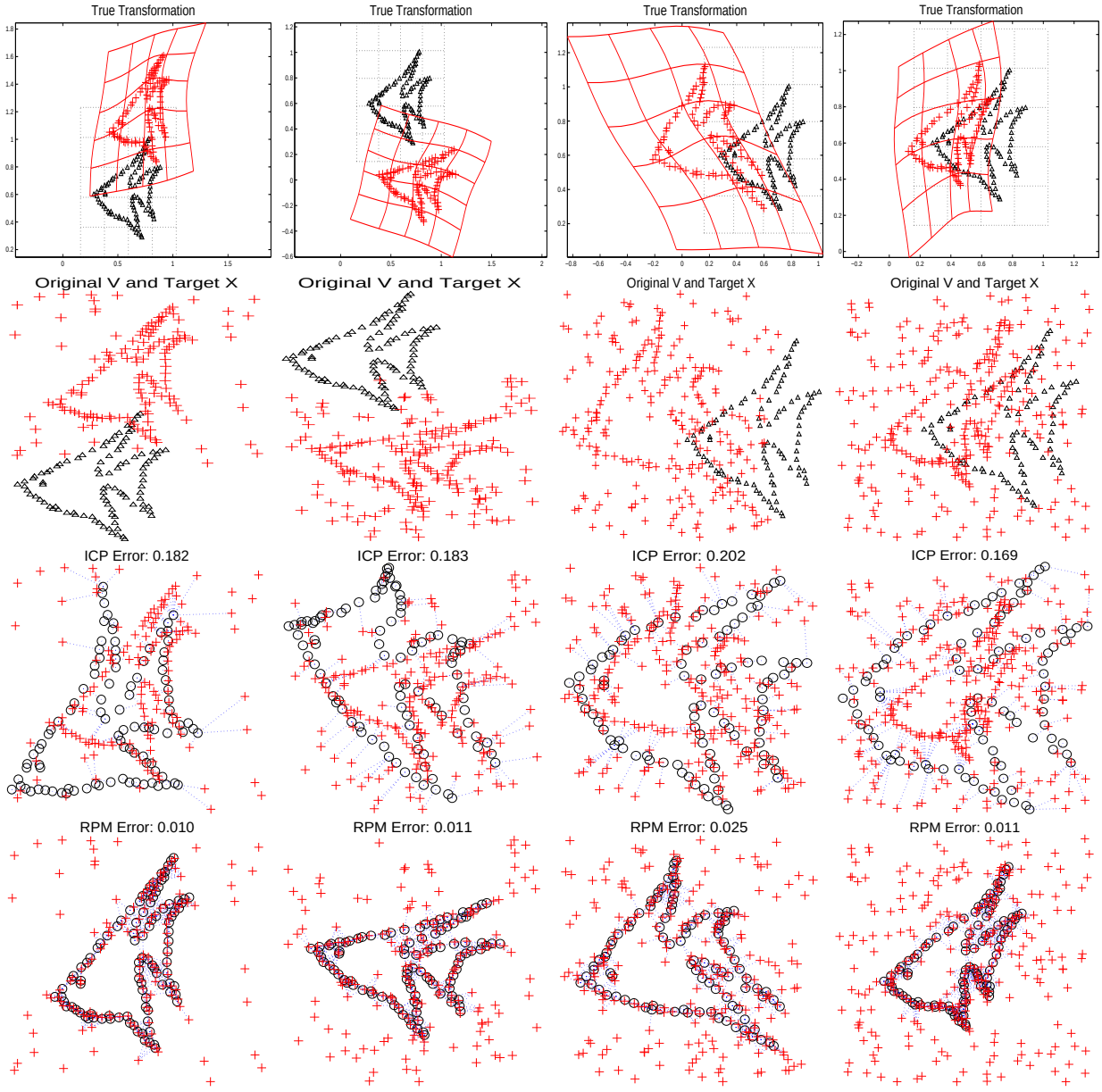

Figure 4.9: Experiments on outliers. From left to right, increasing amounts of outliers. **Top Row:** Warped template. **Second Row:** Template and target (warped template with outliers added). **Third Row:** ICP results. **Bottom Row:** RPM results. The actual error values are also shown. Normally, errors beyond 0.05 indicate poor matching.

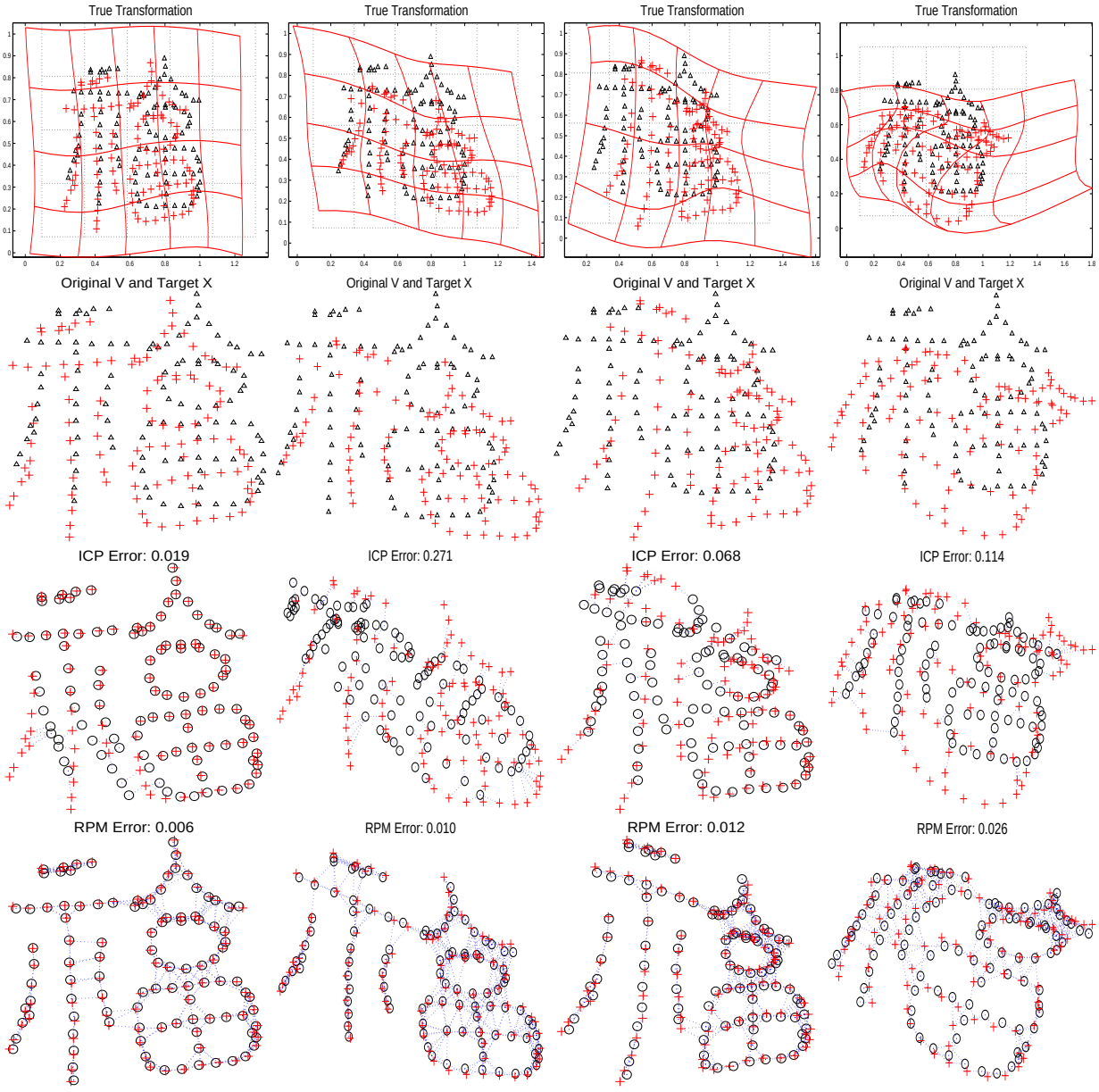

Figure 4.10: Experiments on deformation. From left to right, increasing degrees of deformation. **Top Row:** Warped template. **Second Row:** Template and target (same as the warped template). **Third Row:** ICP results. **Bottom Row:** RPM results. The actual error values are also shown. Normally, errors beyond 0.05 indicate poor matching.

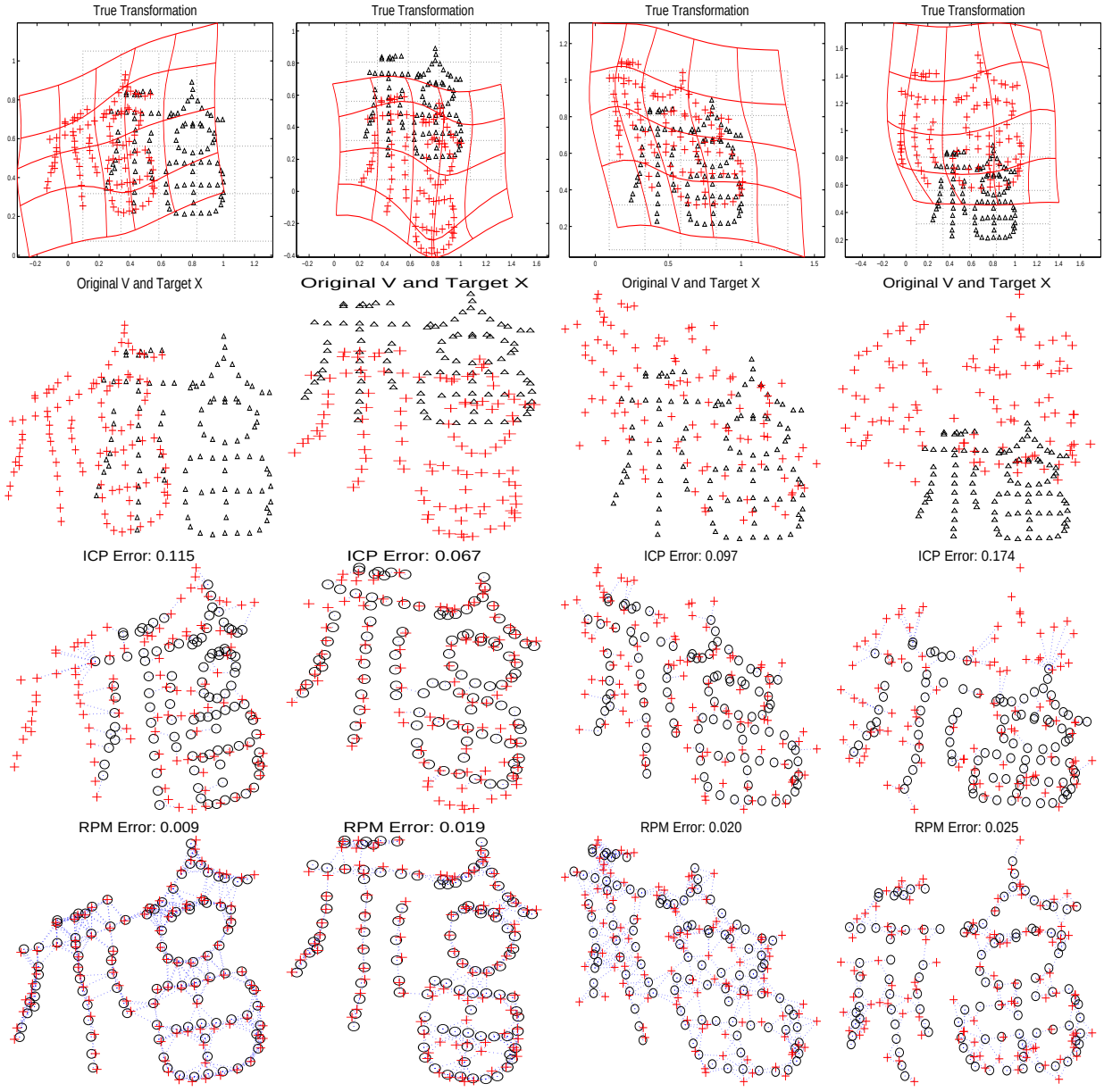

Figure 4.11: Experiments on noise. From left to right, increasing amounts of noise. **Top Row:** Warped template. **Second Row:** Template and target (warped template with noise added). **Third Row:** ICP results. **Bottom Row:** RPM results. The actual error values are also shown. Normally, errors beyond 0.05 indicate poor matching.

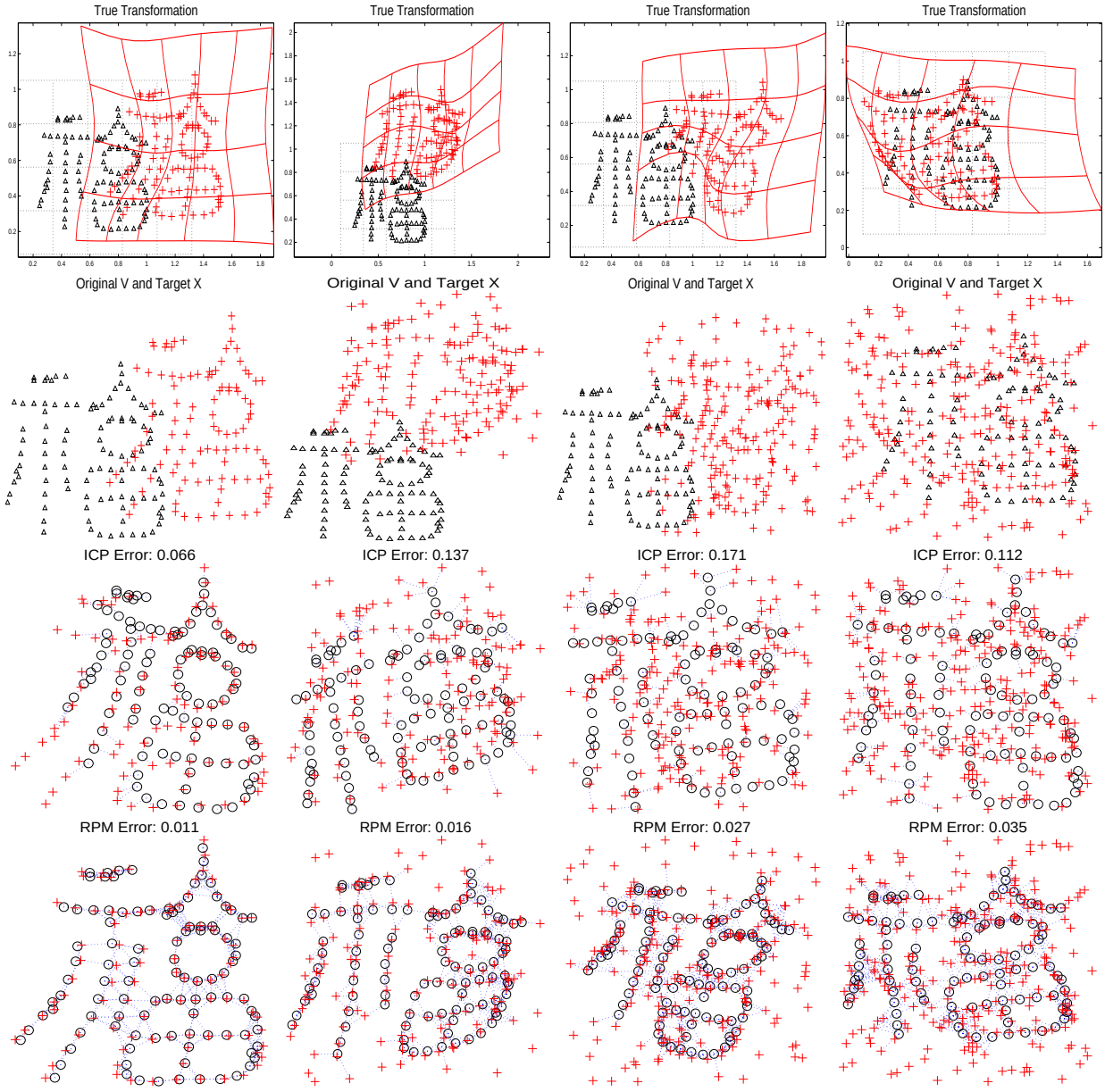

Figure 4.12: Experiments on outliers. From left to right, increasing amounts of outliers. **Top Row:** Warped template. **Second Row:** Template and target (warped template with outliers added). **Third Row:** ICP results. **Bottom Row:** RPM results. The actual error values are also shown. Normally, errors beyond 0.05 indicate poor matching.

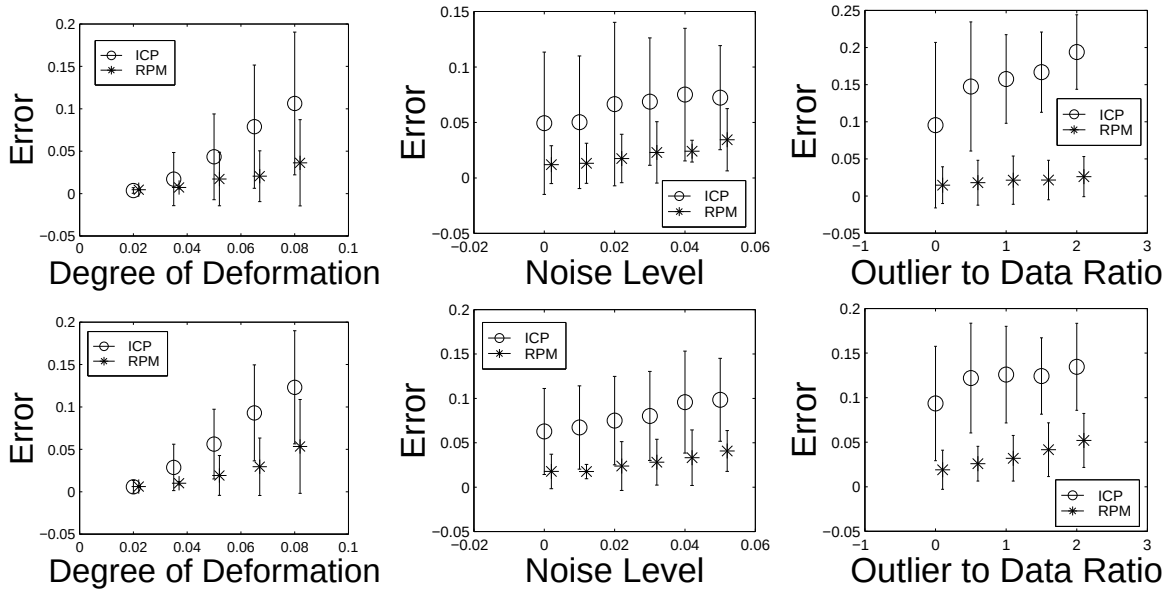

Figure 4.13: Statistics of the synthetic experiments. **Top Row:** Results using the £sh template. **Bottom Row:** Results using the Chinese character template. Note that RPM's errors increase much more slowly than ICP's in all cases.

once outliers are added. On the other hand, RPM seems to be relatively unaffected.

In fact, there are cases when the amount of outlier is so large, it is not easy even for us to tell where the true data points are. However, the RPM algorithm seems not be bothered at all. This phenomenon is very interesting, and actually a little surprising, especially when we know that for the more dif£cult cases more than half of the points in the barge pattern are outliers. When we pursue this issue further, we found that there is an interesting connection between non-rigid point matching and another fundamental problem in computer vision — graph matching. We will discuss this connection later in the next Chapter.

4.5 Experiments Based on Real Data

After these synthetic experiments, we conduct more experiments on real examples where the ground truth is unknown. Three examples are included here.

To investigate RPM's ability to determine meaningful correspondences under large deformations, we studied that matching of a sequence of progressively deformed caterpillar like hand drawings. We will also look at three potential applications for non-rigid point matching. The £rst one is keyframe animation. The second one is the 2D face matching. The third one is the 3D sulcal point matching in the brain registration £eld.

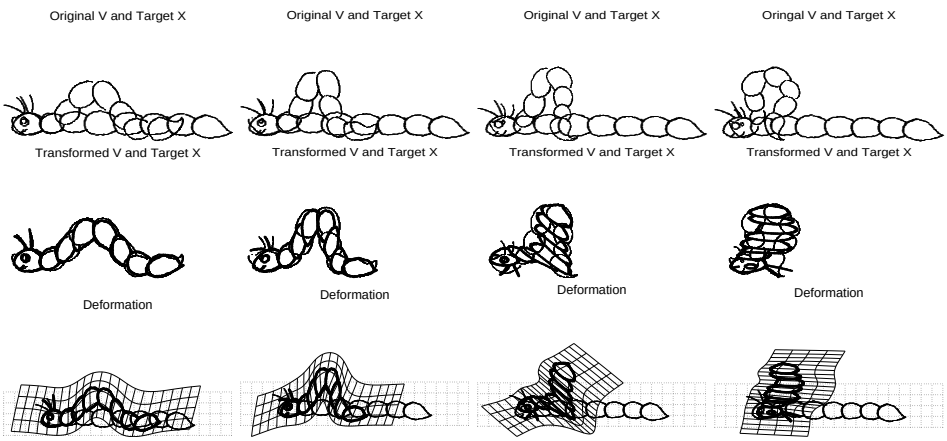

Figure 4.14: Caterpillar example for large deformation study. From left to right, matching frame 1 to frame 5, 7, 11 and 12. **Top Row:** Original location of the two frames. **Middle Row:** Matched result. **Bottom Row:** Deformation found.

Matching a Caterpillar Sequence

From the synthetic experiments, we see that RPM can recover rather large deformations. The larger the deformation, the bigger the difference between the two point-sets and hence the more dif£cult the matching. A natural question is whether there is a limit how large a deformation can RPM successfully recover. To answer this question, we conducted another series of experiments.

We hand-drew a sequence of 12 caterpillar images. The caterpillar is £rst supine. Then it gradually bends its tail towards its head. Further into the sequence, the bending increases. Points are extracted from each image in the sequence through thresholding. We then tried to match the £rst caterpillar point-set with the later deformed ones. The purpose is to test RPM's ability to track various degrees of deformation.

The results of matching the £rst frame to the £fth, the seventh, the eleventh and the twelfth frame are shown in Figure 4.14. We observed that up to frame eight, RPM is able to recover most of the desired transformation. Beyond that, it still tries to warp the points to get the best £t and ends up generating some "strange" unphysical warping with the plane being totally ¤ipped.

Clearly RPM can not track arbitrarily large transformations. However, it is quite possible to improve the RPM algorithm further to handle these extreme cases. For example, certain labelling information can be added to the points. In the caterpillar matching case, labels can be used to differentiate a point in the head section from a point in the tail part so that mismatching the head to the tail can be avoided. Another important improvement can come from

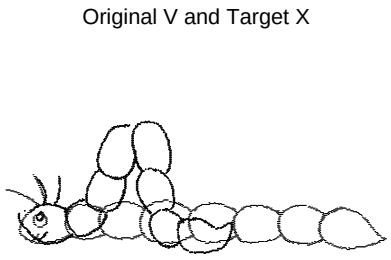

Figure 4.15: Caterpillar example for demonstration of keyframe animation application. Two point sets representing a crawling caterpillar are used here as the key frames.

a better non-rigid transformation model. In this case, a good model for the movement of a caterpillar should obvious take into account of the fact that a caterpillar can only crawl. It should probably only model the deformation of only caterpillar itself instead of model the deformation of the whole space. TPS satis£es neither of these two requirements. It is obviously a too rough a model to most realistically represent the movement of a caterpillar.

Our RPM framework certainly has room to incorporate these improvements. The labeling information can be easily added into the correspondence estimation process to make sure that correspondences are only valid between points sharing the same labels. As we discussed before, the incorporation of a new transformation model is also straight forward.

Keyframe Animation

Given two *key frames*, a fundamental problem in computer aided animation is to automatically generate new *intermediate frames* (*inbetweens*) [76]. In order to accomplish this, corresponding structures in the two key frames have to be found and matched. The caterpillar images provide us with a good example (as shown in Figure 4.15).

Once two keyframes are given, RPM can be used to automatically match them, returning both the correspondence and the transformation information. With this information, the intermediate frames can be easily and accurately generated through interpolation. Here we show a sequence of automatically generated intermediate frames using TPS-RPM (as shown in Figure 4.16). The interpolation is a simple linear interpolation of the estimated TPS parameters d and c. The crawling motion of the carterpillar is quite nicely represented by the sequence.

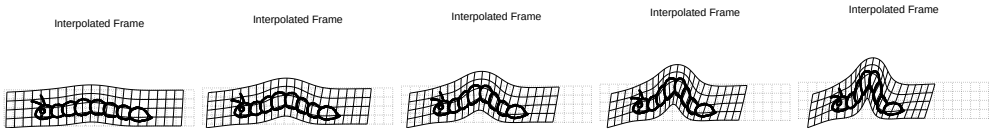

Figure 4.16: Keyframe animation. The automatically generated intermediate frames between the key frames. They seem to be quite realistic.

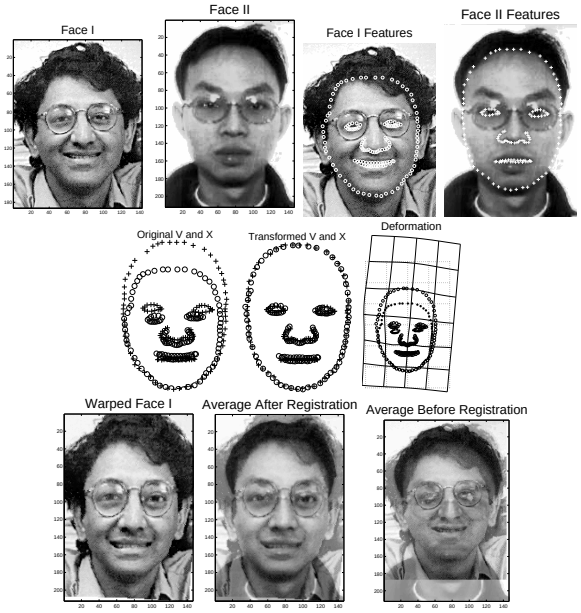

Figure 4.17: Face Matching. **Top Row:** Two face images with their feature points. **Middle Row:** Original position, £nal overlay position and deformation grids. **Bottom Row:** Left, warped face of person I. Middle, average of warped face I and original face II. Right, average of original face I and II.

Face Matching

In the second application, we attempted to match and warp faces [12, 25]. Please note that we are not proposing a complete new approach to solve the face image alignment problem. There has been an enormous amount of work done in that £eld and we certainly do not intend to reinvent the tools. We chose to match faces simply to test and demonstrate the algorithm.

We manually marked a few distinctive points on two faces to get a rough outline of the face structures (face contour, eyes, nose and mouth, see Figure 4.17). RPM is used to solve the point matching problem. The TPS deformation found was then used to warp one of the face images onto the other. It is clear that the outlined structures (and indeed the faces as well) are better aligned through this process.

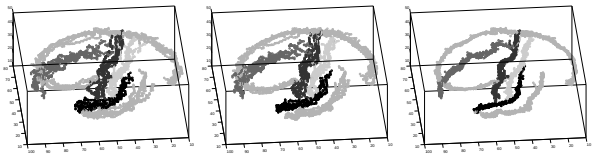

Figure 4.18: 3D Sulcal Point Matching. **Left:** Original position of the 5 sulcal point-sets overlay. Only 6 sulci on one half of the brain are shown. **Middle:** The overlay after af£ne RPM registration. **Right:** The overlay after TPS-RPM registration. Note the improvement of TPS-based registration over af£ne registration.

Brain Sulcal Point Matching

The registration of anatomical structures in brain MRI is a challenging problem for brain mapping in medical imaging. One way to accomplish this is through the matching of sulcal feature points [16]. It is a dif£cult task because of the highly convoluted nature of the human brain cortex. Non-rigid registration is necessary to bring different subjects' brain MRI into a comparable common coordinate frame.

In fact, brain registration is one of the main applications we studied in this work. We will discuss the brain registration problem in much greater detail in Chapter 7. The example shown here is only intended to be a illustrative case for the RPM for 3D point matching. We use RPM to solve for the TPS, and (for comparison purposes), for the af£ne on 5 MRI sulcal point-sets (acquired from 3D MRI volumes [16]). Overlays of all sulcal point-sets before and after the matching are shown in Figure 4.18. Denser, tighter packing of the sulci indicates better registration.

Chapter 5

Relationship of Non-rigid Point Matching to Graph Matching

5.1 Relationship to Graph Matching

From the synthetic experiments, we found it especially interesting that RPM does not seem to be adversely affected even in the presence of a large number of outliers. Somehow, the points in the template seem to be acting holistically, i.e, a point is actually relying on the shape/layout of its neighboring points while seeking a good match. At £rst look, our algorithm is seemingly only trying to match the points simply based on their coordinates. There is no other information, such as the point inter-relationship, to describe the points in our formulation. Without any such information to describe the relationship between the points, they should be acting more like independent entities. There seem to be no reason that the points should behave the way they did. This rather puzzling phenomenon prompted us to look into this issue deeper. The investigation reveals an interesting connection between point matching and graph matching.

Graph representation has long been used as a good way to formulate a structural description of an object [73]. Inside a graph representation, there are usually two components. The £rst is a set of nodes. In our case, each node is a point and the point coordinates describe the node's identity (attribute). The second is a set of links (edges) between the nodes. There are many different types of links proposed, but all of them essentially serve the same purpose—representing the inter-relationship between the nodes (points). When graphs are being matched, both the nodes as well as the links will be matched. Informally speaking, a pair of nodes is considered matchable only when both their

"personal identities" and their "social connections" are similar to each other. Because of the representation power of graphs, the construction of the graph representation and the resulting graph matching problem have been regarded as two of the most fundamental research areas in the computer vision.

Compared to graph matching, our point matching seems to be quite different since only the points (nodes) are being matched. However, the points are not totally independent. During the matching, the nearby points do in¤uence each other via the smoothness constraint, i.e., each point's movement is actually constrained by its neighbors. This effectively provides an implicit way to model the relationship between the points.

It can be shown more formally that there is always a connection between our non-rigid point matching approach and graph matching. For example, we can show that the TPS-RPM formulation is equivalent to a weight graph matching. Since the TPS transformation can always be uniquely determined once the correspondence is given, we can plug the answer for TPS into the original energy function and eliminate TPS from the unknown variables. The effective energy will then only depend on the correspondence M. The detailed derivation for TPS is discussed in Chapter 3 and in the Appendix section.

For the sake of simplicity, here we only include the relevant £nal results. By substituting the TPS solution (9.34), (9.35) and (9.36) into the least-squares non-rigid transformation cost function (3.8) and by using the relationship in (3.9), we get the following *effective* energy function which depends only on the correspondence M:

$$E = \lambda \,\text{trace}\,[M^T G_V M G_Y] \tag{5.1}$$

where,

$$\begin{aligned} G_V &= Q_2(Q_2^T \Phi Q_2 + \lambda I_{(K-D-1)})^{-1} Q_2^T, \\ G_Y &= YY^T. \end{aligned}$$

If we think of G_V and G_Y as two graph matrices, this is also exactly a standard quadratic assignment graph matching energy function [100, 34]. While we are minimizing this energy function for point matching, we are effectively solving a graph matching problem as well. In this sense, TPS-RPM is a graph matching algorithm as well.

This intimate relationship between TPS in point matching and weighted graph matching is not accidental. Whenever a geometrical mapping used in point matching is constrained by a smoothness measure, the points will, we believe, interfere with each other, providing implicit model for the inter-relationship. The assignment formulation,

which is used in our algorithm, has also been shown to be closely related to graph matching [100, 34].

Once this connection is established, we go on to study the properties of the new graphs derived here.

5.2 TPS Graphs

Since the graph G_V comes from the template point-set V that is being transformed by a TPS, we call it the TPS graph. The other graph G_Y is from the target point-set Y and we call it the target graph.

The target graph G_Y is rather simple and doesn't seem to has a lot of information. The TPS graph G_V is much more interesting. Since it depends on the parameter λ, we denote it as $G_V(\lambda)$. During our point matching process, estimating the correspondence matrix M at each step is equivalent to solving the weighted graph matching problem between G_Y and $G_V(\lambda)$. The template graph $G_V(\lambda)$ keeps changing as the parameter λ decreases, whereas the data graph G_Y is held £xed in the whole process.

We mentioned previously that the TPS kernel contains the information about the point-set's internal structural relationships. Now the graph matrix $G_V(\lambda)$ allows us to explicitly visualize these inter-relationships in the context of graph matching. Some examples of the TPS graph are shown in Figures 5.1 and 5.2. The graphs of the target graph are also include for comparison.

From the examples, we see that the TPS graph model different aspect of the points' shape as the parameter λ changes. When the value of λ is high, the signi£cant links are distributed at the outer boundary as well as at the major diagonal region. As the value of λ becomes small, such global links are replaces by links that are more and more localized. The TPS graphs at low λ look more like conventional graphs.

This connection to graph matching might be another reason to explain the RPM algorithm's global-to-local match property. The results here are preliminary. Further study is required to bridge the gap between these two seemingly disparate areas.

5.3 Relationship of Non-Rigid Point Matching to Other Fields

There is also an interesting connection between non-rigid point matching and a class of problems in the £eld of neural networks. One of them is the Kohonen *self-organization map* (SOM) [51]. The SOM accomplishes dimensionality reduction by approximating a large number of data (often high dimensional) with a small number of unit kernel functions (for example, the Gaussian radial basis function). When combined with an additional smoothing prior, the SOM algorithm [90] shares a lot in common with our algorithm. As a probabilistic re-formulation of the SOM, the

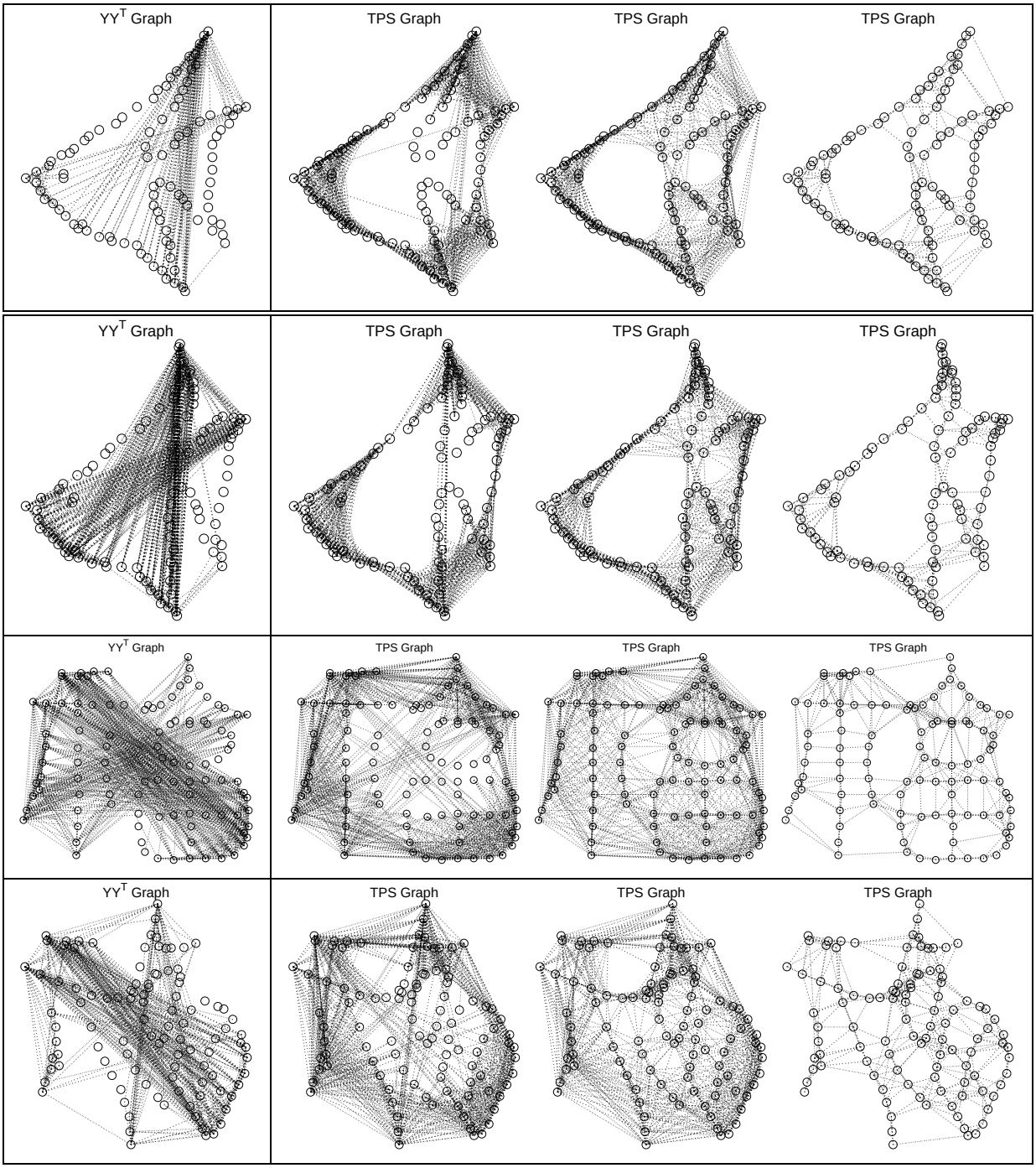

Figure 5.1: Two graph representations. **First Row:** the graphs of the £sh pattern. Leftmost: XX^T graph. The entries with relatively large absolute values are shown as links. From second left to the rightmost: TPS graph with gradually reduced values of λ . **Second Row:** the graphs of a warped £sh pattern. **Third Row:** the graphs of a Chinese character. **Forth Row:** the graphs of a warped Chinese character.

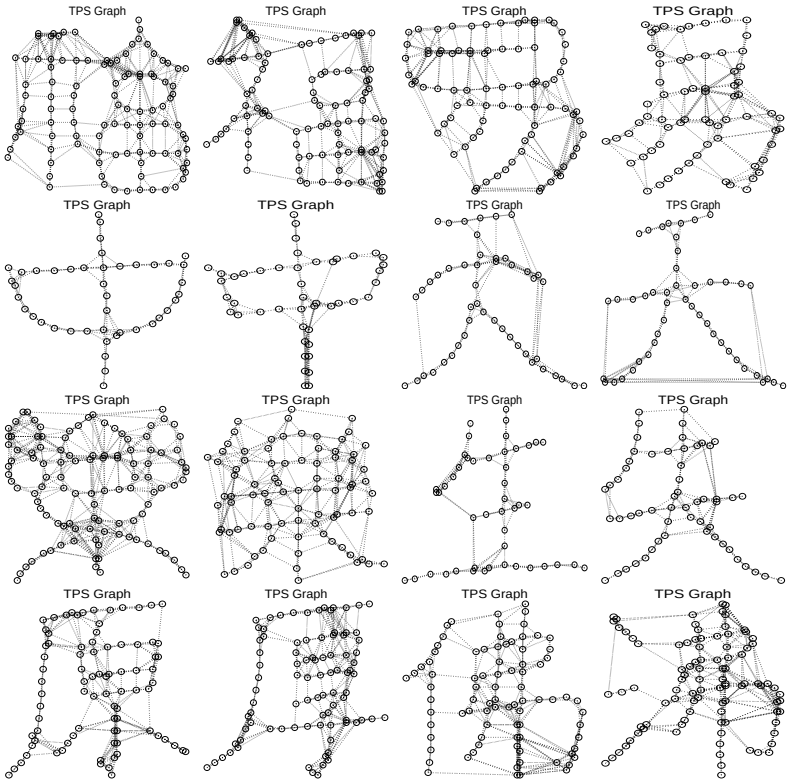

Figure 5.2: More TPS graphs. All the graphs shown here are TPS graphs at a small value of λ.

Generative Topographic Mapping (GTM) model [9] is also closely related to our point matching framework, especially when viewed from a probabilistic angle.

Chapter 6

Extensions of the Non-Rigid Point Matching Algorithm

6.1 Theoretical and Practical Reasons for the Extensions

So far, we have explained the development of our robust non-rigid point matching algorithm and evaluated its performance through various experiments. We also discussed the interesting connection between point matching and graph matching.

Essentially, the formulation of the non-rigid point matching problem and algorithm is the foundation of this thesis. In this chapter, we seek to further extend the RPM framework. There are theoretical as well as practical reasons for these modi£cations. They can be summarized as follows:

1. Develop more symmetric formulations.
2. Reduce the computational cost.
3. Augment the functionality of the point matching algorithm.

We £rst discuss the symmetric formulation. The basic RPM algorithm provides a framework to warp one point-set onto another as closely as possible, while returning a good correspondence estimate. The framework is quite general. It accommodates different types of non-rigid transformations. It tolerates a reasonable amount of noise and can automatically reject a fraction of the points as outliers. However, when examined from a theoretical point of view, the formulation is not symmetric, i.e., the two point-sets are not exactly treated equally (as illustrated in Figure 6.1). One

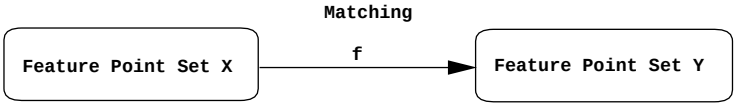

Figure 6.1: Original RPM formulation is not symmetric.

point set is chosen to be warped. The other is chosen to be the target. If the roles of the point-sets are switched, the answers we get are obviously going to be different. Though this non-symmetry is in fact probably what is required by most of the image feature registration tasks, there are de£nitely registration situations where symmetry is required.

With the symmetry being the theoretical motivation, there are other possible improvements that we can make from practical considerations, such as reducing the overall computational cost. When the total number of feature points reaches into the thousands, evaluating correspondence and transformation will be computationally very expensive.

Of course, the complexity of the problem can be reduced if we use a smaller number of points through subsampling. But then resolution and accuracy are sacri£ced. We are interested in achieving a compromise between reducing the computation cost and sacri£cing accuracy.

Another practical consideration arises from the problem of estimating an average shape given multiple shape point-set samples. In that case, simultaneous matching of multiple point-sets is required. Since normal point matching algorithms are only capable of matching two point-sets, we have been seeking ways to extend the point matching algorithm to be able to accomplish the task of symmetric multiple point-set matching.

Out of these considerations, or rather curiosities, we set out to add modi£cations to our basic RPM framework. We arrive at the following three new formulations:

1. A symmetric RPM Formulation.
2. A symmetric basic joint cluster-matching formulation.
3. A symmetric super joint cluster-matching formulation.

All of the these new formulations are symmetric in the sense described above. The £rst is a straightforward modi£cation of the original RPM algorithm. Though it is symmetric, it doesn't reduce the computational cost. The latter two approaches are both based on a joint clustering and matching strategy. The original point-sets, instead of being used directly for matching, are clustered down clustered down to fewer cluster centers. Subsequently, the cluster centers are used for the matching. The key is that the clustering is performed simultaneously with the matching so that the loss of

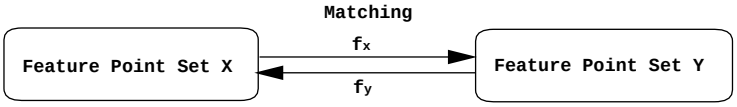

Figure 6.2: The Symmetric RPM formulation (SRPM). The original RPM formulation is modi£ed to be symmetric with two transformations.

accuracy is minimized. The third approach is termed “super joint clustering-matching formulation” because it allows multiple point-sets to be simultaneously matched while an average point-set is also recovered.

Since these algorithm share a lot in common with the original RPM algorithm, we will not rederive them from £rst principles. Instead, we focus on those aspects that differs most from the original RPM algorithm. The details are below.

6.2 A Symmetric RPM Formulation

As we just discussed, we intend to extend the original RPM algorithm and seek a more symmetric formulation.

The main requirement is that in a symmetric formulation, the results should be invariant to role reversal of the point-sets. It should not matter which point-set is the template and which one is the target.

In the original RPM algorithm, one point set is regarded as the template to be warped while the other is chosen to be the target. To achieve symmetry, a simple modi£cation is to make each point-set to be both the template and the target at the same time. So given two point-sets X and Y, we need two transformation modules f_x and f_y, instead of one as in the original formulation. One transformation f_x warp the point-set X to Y, and the other f_y warp Y to X. This idea is illustrated in Figure 6.2.

For simplicity, we use the transformation symbol f_x and f_y to represent the complete parametric transformation function and omit the introduction of further parameters (such the parameter α used before). Other than the need to solve for these two transformations (instead of one), this new formulation is the same as the original RPM framework. We call the new one the Symmetric RPM (SRPM) formulation.

Before we write down SRPM’s energy function, let’s brie¤y review our notations. Suppose the point-set X consists of N points $\{x_i, i = 1, 2, \dots N\}$ and, the point-set Y consists of K points $\{y_j, j = 1, 2, \dots K\}$. It is possible that N is not equal to K. We use an matrix M to represent the correspondence between these two point-set. The matrix M is an $(N + 1) \times (K + 1)$ matrix with entries m_{ai}. The forward transformation ($X \rightarrow Y$) is denoted as

f_x and the reverse transformation ($Y \rightarrow X$) as f_y. The new energy function is the following:

$$\begin{aligned} E_{SRPM}(M, f_x, f_y) &= \sum_{i=1}^{N+1}\sum_{a=1}^{K+1} \frac{m_{ai}}{2}\left(||x_i - f_x(y_a)||^2 + ||f_y(x_i) - y_a||^2\right) \\ &\quad +\lambda T||Lf_x||^2 + \lambda T||Lf_y||^2 \\ &\quad +T\sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}\log m_{ai} \\ &\quad +T_0\sum_{a=1}^{K} m_{a,N+1}\log m_{a,N+1} + T_0\sum_{i=1}^{N} m_{K+1,i}\log m_{K+1,i} \end{aligned} \tag{6.1}$$

where once again the parameter λ is a weight constant, and the variable $m_{ai} \in [0, 1]$ is subject to the following constraints,

$$\sum_{i=1}^{N+1} m_{ai} = 1, \quad \text{for} \quad a \in \{1, 2, \ldots, K\}$$
$$\sum_{a=1}^{K+1} m_{ai} = 1, \quad \text{for} \quad i \in \{1, 2, \ldots, N\}$$

The resulting algorithm (as shown below) is almost the same as the original RPM except that in the alternating update (step 2) for the transformation, two transformations now have to be updated instead of one.

The Symmetric RPM Algorithm Pseudo-code:

Initialize parameters $T = T_0$ and λ.

Initialize parameters f_x and f_y (e.g., identity transformation).

Begin A: Deterministic Annealing.

> **Begin B: Alternating Update.**
>
> Step 1: Update correspondence M based on current f_x and f_y.
>
> Step 2: Update transformation f_x and f_y based on current M.
>
> **End B**

Decrease $T \leftarrow T \cdot r$ until T_{final} is reached.

End A

The two transformations f_x and f_y involved here obviously are not unrelated. Ideally, they should be inverse of each other (provided the inverses exist). Instead of using f_x and f_y, we could use one transformation f as f_x, and replace the f_y with the inverse f^{-1}. The choice whether to enforce this constraint will depend on the problem at hand [14].

6.3 A Symmetric Joint Clustering-Matching Formulation

A Joint Clustering-Matching Formulation

At this juncture, it is not clear if the symmetric RPM formulation actually may only give us a new form that is neat mathematically without any practical gain in the sense of achieving a more ef£cient algorithm. The next modi£cation that we are interested in is to increase the ef£ciency of the algorithm.

Reducing the number of feature points to match will greatly lessen the burden of computation. For example, there is a big difference in speed between using the TPS-RPM algorithm to match 100 points versus 1000 points. Evaluating a correspondence matrix of 100×100 is de£nitely much faster than working on a matrix of 1000×1000. Calculation of the TPS parameters involves the inverse computation of matrices. While inverting a 100×100 matrix takes less than one tenth of a second in MATLAB^{TM} on a desktop PC, inverting a 1000×1000 matrix will take more than a minute.

One way to reduce the number of points is through clustering (or more generally, subsampling). Thousands of the original points can be clustered down to, say, a few hundreds of *cluster centers*. Then the cluster centers, which will still roughly represent the original pattern, can be used in lieu of the original point-set for the matching.

However, the coarser representation using the cluster centers will also cause the matching result to be less accurate. An important reason is the *subsampling problem*: the subsampled points (in our case, the cluster centers) normally can no longer be matched exactly. The problem is illustrated in Figure 6.3. The shape of the original points is a simple contour pattern. The points are subsampled twice (through the same clustering algorithm but with slightly different initializations). The subsampled points are obviously different. If we try matching the subsampled points exactly to each other, the transformation is not going to be zero (which is equivalent to an identity transformation). This clearly is a sub-optimal solution since the subsampled points are from exactly the same source. The warping attempts to compensate for subsampling errors which seems to be a futile exercise.

Subsampling is needed because of the computational gain. The main question before us is how the subsampling errors can be reduced. In other words, we need to subsample the points in a way such that the subsample points are still exactly matchable. Posed in this manner, we claim that the question largely answers itself. We need to perform the subsampling process in lockstep with the matching process while allowing the two processes to communicate with each other. In this way, we obtain two sets of subsampled points that are exactly matchable. The overall matching process will also be more ef£cient.

We chose clustering as our subsampling procedure. Since our RPM algorithm is effectively a clustering

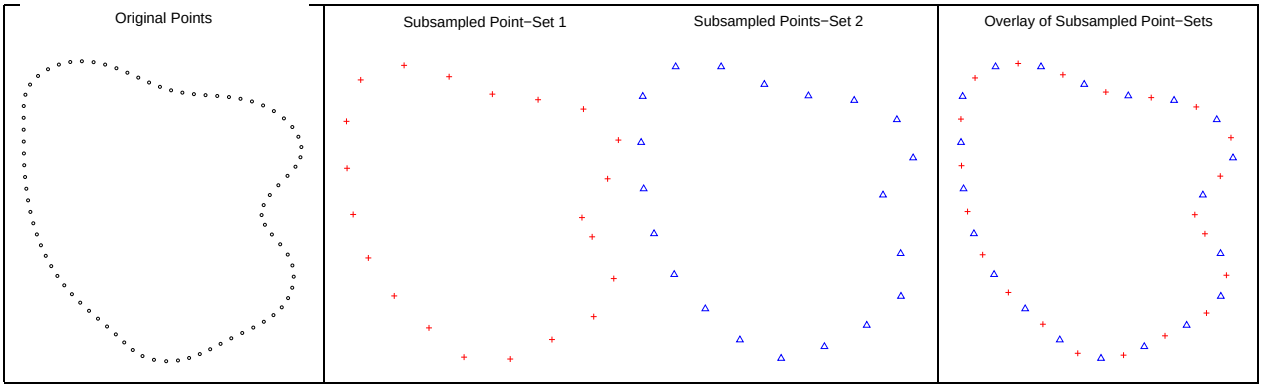

Figure 6.3: The subsampling problem. From left to right: i) the original point-set with densely distributed points; ii, iii) two subsampled point-sets; iv) the overlay of the subsampled point-sets. Note the difference between the two subsampled point-sets.

algorithm already (as discussed in Chapter 3 and the Appendix), minimal modi£cations are required.

Beginning with two original point-sets X and Y, we derive two sets of cluster centers V and U. The outlier cluster structure in the original RPM is conveniently inherited. Each cluster center set will have an outlier cluster to account for the spurious data points. The rest of the cluster centers would represent the common structures shared by both original point-sets. These cluster centers are then matchable. Only these matchable cluster centers are used in the matching. The correspondence matrices M^x and M^y are used to represent the membership information between the cluster center set and its own point-set. We also make the formulation symmetric by putting in two transformations, one for the forward and one for the reverse. We call the resulting algorithm the joint clustering-matching algorithm (JCM). The setup of JCM is shown is Figure 6.4.

The Symmetric Joint Clustering-Matching (JCM) Energy Function

The joint clustering and non-rigid deformation estimation framework is equivalent to the minimization of the following objective function:

$$\begin{aligned} E(V,U,f_x,f_y,M^x,M^y) \quad = \quad & \sum_{i=1}^{N_x}\sum_{a=1}^{K+1} m_{ai}^x||x_i - v_a||^2 + \sum_{j=1}^{N_y}\sum_{a=1}^{K+1} m_{aj}^y||y_j - u_a||^2 \\ & + \sum_{a=1}^{K}||u_a - f_x(v_a)||^2 + \sum_{a=1}^{K}||v_a - f_y(u_a)||^2 \\ & + \lambda||Lf_x||^2 + \lambda||Lf_y||^2 \end{aligned}$$

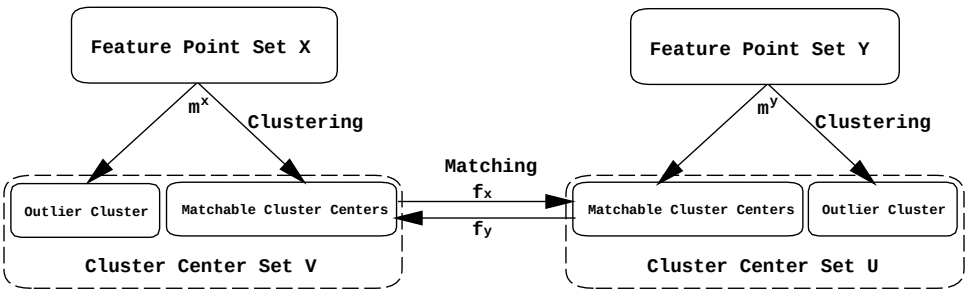

Figure 6.4: The joint clustering-matching formulation. Each original point-set (X and Y) is being clustered down to a set of cluster centers ($X \rightarrow V$ and $Y \rightarrow U$). Each cluster set has an outlier cluster (v_{K+1} in V, u_{K+1} in U) to account for the possible spurious points in each point-set. The rest of the cluster centers ($\{v_a, a = 1, 2, \dots K\}$ and $\{u_a, a = 1, 2, \dots K\}$) are being matched. So they are called the "matchable" clutter centers because they represent the common structures within the two point-sets.

$$
\begin{aligned}
&+T\sum_{i=1}^{N_x}\sum_{a=1}^{K} m_{ai}^{x}\log m_{ai}^{x} + T\sum_{j=1}^{N_y}\sum_{a=1}^{K} m_{aj}^{y}\log m_{aj}^{y} \\
&+T_0\sum_{i=1}^{N_x} m_{K+1,i}^{x}\log m_{K+1,i}^{x} + T_0\sum_{i=1}^{N_y} m_{K+1,j}^{y}\log m_{K+1,j}^{y}
\end{aligned}
\tag{6.2}
$$

where $m_{ai}^{x} \in [0,1]$ and $m_{aj}^{y} \in [0,1]$ should satisfy, $\sum_{a=1}^{K+1} m_{ai}^{x} = 1$ and $\sum_{a=1}^{K+1} m_{aj}^{y} = 1$.

The objective function for JCM in (6.2) is closely related to the objective function of RPM. Here we intend to give brief but more intuitive explanations for all the terms.

The £rst two terms are average distance measures between the data and the cluster centers. Note that the memberships present in the distance measure are themselves unknown. These two terms basically measure the degree of £delity of the cluster centers (V and U) to the data (X and Y) respectively.

The third and the forth terms try to £nd the best deformation (both the forward and the reverse at the same time) to match the two sets of cluster centers. Instead of matching the original data points, the deformation estimation attempts to match the cluster centers.

The £fth and the sixth terms play the role of regularization. The parameter λ is a weight parameter which controls the degree of deformation; larger the regularization parameter, smaller the extent of deformation and vice-versa.

The rest of the terms come from deterministic annealing. The temperature parameter T controls the fuzziness

of the membership matrices: higher the temperature, greater the fuzziness. This form of entropy term effectively leads to Gaussian clusters. The parameter T can also be regarded as the size of the clusters. The £ne control of the fuzziness can be achieved by gradually reducing T. Again, the outlier cluster is assigned with a separate temperature T_0 with a large value to account for all possible outlier points.

Note that JCM only requires one more parameter (the total number of cluster centers K) than the original RPM formulation.

Alternating Updates

There are three pairs of unknown parameters inside this setup. The £rst pair are the cluster center sets V and U. The second are the membership matrices M^x and M^y. The third are the forward and reverse transformations f_x and f_y. We can derive an alternating update scheme similar to RPM using the inter-relationships between these parameter sets.

i) Update the Membership Matrices

The membership matrices (M^x, M^y) are determined by the original data point-sets (X, Y) and the currently estimated cluster centers (V, U). We inherit the deterministic annealing to control the fuzziness of the membership matrices and subsequently to control the clustering process. This effectively leaves us:

$$m_{ai}^x = \frac{q_{ai}^x}{\sum_{a=1}^{K+1} q_{ai}^x}, \qquad \text{for } a = 1, 2, \ldots K+1 \text{ and } i = 1, 2, \ldots N, \tag{6.3}$$

where,

$$q_{ai}^x = e^{-\frac{||x_i - v_a||^2}{T}}, \qquad \text{for } a = 1, 2, \ldots K \text{ and } i = 1, 2, \ldots N, \tag{6.4}$$

$$q_{K+1,i}^x = e^{-\frac{||x_i - v_{K+1}||^2}{T_0}}, \quad \text{for the outlier } a = K+1 \text{ and } i = 1, 2, \ldots N. \tag{6.5}$$

Note that these two matrices now represent the membership information between the original points and the cluster centers. One cluster center can have multiple points as members. The original two way constraints for the correspondence matrix are no longer needed. The only constraint the membership matrices need to satisfy now (other than positivity) is the row normalization constraint: $\sum_{a=1}^{K+1} m_{ai}^x = 1$. This requirement is already included in the update equations above and there is no need for further Sinkhorn normalization. Exactly the same rules apply for M^y.

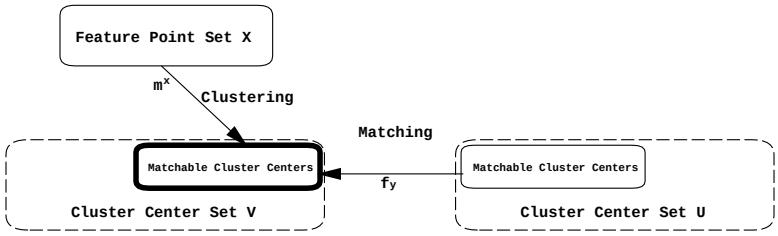

Figure 6.5: Matchable cluster centers in V. A matchable cluster center v_a is in¤uence both by its original data points X through the membership matrix M^x, as well as its counterpart u_a from the other set through the transformation f_y.

ii) Update the Cluster Center Sets

As before, the outlier cluster still serve the same purpose of accounting for the outlier points. The key of this joint clustering-matching algorithm is the estimation for the matchable cluster centers. As we explained just above, they should represent the original data and also be exactly matchable.

We have removed part of the diagram in Figure 6.4 to illustrates this idea. It is shown in Figure 6.5. The cluster centers for each point set have to assume positions that most faithfully represent the original data. The crucial piece of information that helps decide cluster placement are the membership matrices (e.g. M^x). What's more, those positions of the cluster centers from both sets should also be closely related so that they are matchable in the sense described above.

In sum, we are gradually making a transition from the earlier RPM formulation to a new constrained clustering formulation wherein the RPM correspondence matrix has been replaced by two cluster membership matrices and where clustering is performed simultaneously with the matching (warping).

To derive the formal update equation, we use one cluster center v_a as an example. Given its set of memberships and the original data, v_a should be placed at the membership weighted centroid of the data, i.e., $\frac{\sum_{i=1}^{N} m_{ai}^x x_i}{\sum_{i=1}^{N} m_{ai}^x}$ if the membership information M^x is given. Further more, we also have to keep it consistent with the counterpart cluster center u_a. Given the warping information, v_a should also be placed close to $f_y(u_a)$. For these two requirements to be both satis£ed to a certain extent, we update the matchable cluster centers according to the following:

$$v_a = \frac{1}{2}\left(\frac{\sum_{i=1}^{N} m_{ai} x_i}{\sum_{i=1}^{N} m_{ai}} + f_y(u_a)\right) \tag{6.6}$$

As before, the same rules applies for the set of cluster centers U.

Since the clustering and the matching are simultaneously performed, a cluster center v_a is always in exact correspondence with the cluster center u_a (the one with the same index). This correspondence is implicitly maintained throughout the algorithm and deemed known. In a way, the role of the correspondence has been "phased out" in this formulation.

iii) Update the Transformation

Since the correspondence between the matchable cluster centers are known, the update of the transformations again will become a least-squares spline £tting problem.[1]

$$f_x = \arg\min_{f_x} \left(\sum_{a=1}^{K} ||u_a - f_x(v_a)||^2 + \lambda ||L f_x||^2 \right), \tag{6.7}$$

$$f_y = \arg\min_{f_y} \left(\sum_{a=1}^{K} ||v_a - f_y(u_a)||^2 + \lambda ||L f_y||^2 \right). \tag{6.8}$$

The Symmetric joint Clustering-Matching Algorithm

After a little arrangement of the three update steps, we have our joint clustering-matching algorithm. The pseudocode is below.

The Symmetric Joint Clustering-Matching Algorithm Pseudo-code:

Initialize parameters $T = T_0$ and λ.

Initialize membership matrices (M^x, M^y) (e.g., use uniform matrix).

Initialize transformations (f_x, f_y) (e.g., use Identity transformation).

Begin A: Deterministic Annealing.

> **Begin B: Alternating Update.**
>
> Step 1: Update cluster centers (V, U) based on current (M^x, M^y) and (f_x, f_y).
>
> Step 2: Update transformation (f_x, f_y) based on current V and U.
>
> Step 3: Update clustering membership (M^x, M^y) based on current V and U.
>
> **End B**

Decrease $T \leftarrow T \cdot r$ until T_{final} is reached.

End A

[1]The update equation used for the cluster centers has been slightly simpli£ed. The deformations are held £xed when the cluster centers are updated despite the fact that the deformation (TPS) is actually a function of the cluster centers. Since the deformations are being slowly re£ned during the entire iterative procedure, this approximation is justi£ed and considerably simpli£es the calculation.

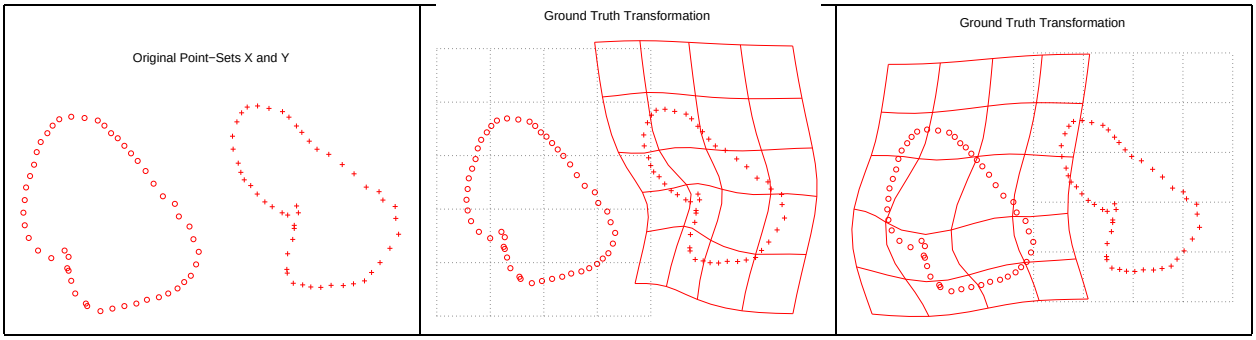

Figure 6.6: A simple example to test the JCM algorithm. From left to right: i) the two original point-sets; ii,iii) the ground truth forward and reverse transformations.

Examples for JCM

We present a few demonstrative examples for JCM here. We start with a simple example (as shown in Figure 6.6) where the ground truth is known beforehand. Each of the original data point-set has 50 points. The results of JCM using $K = 10$ cluster centers are shown in Figure 6.7.

From this simple example, we see that the JCM algorithm is working exactly according our design. The estimated cluster centers are placed at almost exactly the corresponding positions. Though the matching is done with only one £fth of the original data, it recovered most of the desired transformation.

No matter how many points the original point-sets have, we are free to choose the number of cluster centers, K. Using smaller K's greatly simpli£es the computation, while bigger K's will improve the provide more detailed representation of the original data to improve the registration. The effect of using different values of K's are compared in Figure 6.8. This new aspect of the JCM clearly indicated its potential for a further combined coarse-to-£ne match strategy by gradually increasing the value of K.

We use the face images (used before in chapter 4) as a real data example for JCM. In this case, the pattern to be matched is slightly more complicated and the ground truth is unknown. Results of using different values of K are compared in Figure 6.9. In a way, the cluster center sets estimated at different levels of K can be regarded as concise descriptions of the face pattern at different levels of detail. The original point-sets have roughly 150 points. From the experiments, we see that at around 20 clusters, the results are satisfactory [2].

[2]Careful readers will notice that the improvement is not exactly linear as K increase this time. Actually when $K = 7$,the result seems rather poor. This has to do with the phenomenon of "phase transition" in the deterministic annealing. When interpreted in a clustering context, a phase transition occurs when clusters suddenly split as the symmetry breaks. Such a sequence of splitting will allow the originally crowded clusters (as we discussed before, they will all be near the center of mass at high temperature at £rst) to break away from each other and compose £ner and £ner

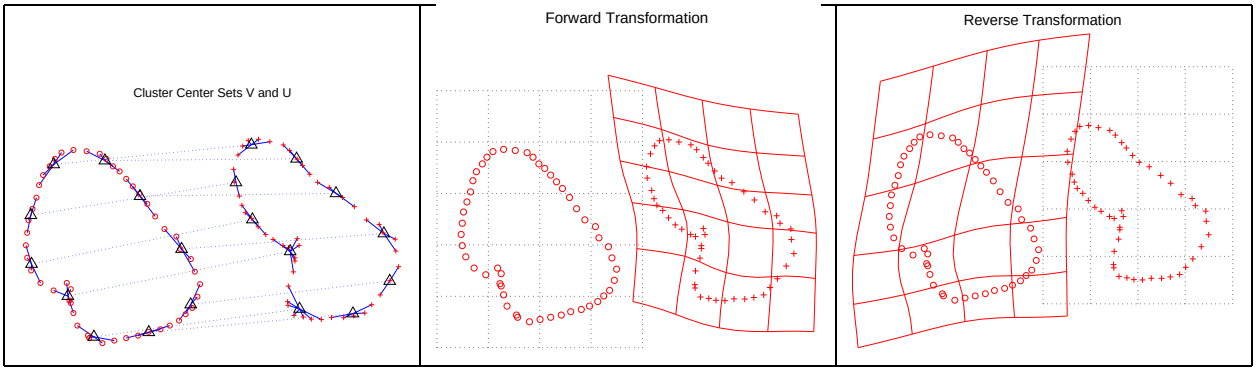

Figure 6.7: The results of JCM using $K = 10$ clusters. From left to right: i) the estimated cluster centers. The membership between the clusters and their member points are shown as solid lines. The correspondences between the clusters (implicitly maintained in JCM) are shown as dotted lines; ii,iii) the estimated forward and reverse transformations. Note that these estimated transformations are pretty close to the ground truth. Also note that in i), the cluster are all put at the corresponding locations. Only the matchable cluster centers are drawn here. The outlier cluster center is omitted.

In the next Chapter, we will simplify the JCM algorithm even more and utilize it for one of the main application in this thesis — brain anatomical feature registration. More details and example of JCM will also be discussed there.

6.4 A Symmetric Super Clustering-Matching Formulation

The Motivation For Multiple Point-set Matching

Through the £rst two extensions, we have demonstrated how two point-sets can be symmetrically matched in different ways. Now we further explore the possibility of matching multiple (more than two) point-sets simultaneously.

The multiple point-set matching idea is inspired by the "shape average" problem encountered in the £eld of shape analysis. The problem can be brie¤y summarized as the following: given P shape patterns, $S_1, S_2, \ldots S_P$, compute the average shape S that minimizes the joint distance $\sum_{p=1}^{P} d^2(S, S_p)$. Computation of an average shape has many applications in various £elds where shape analysis is required. It is especially important in medical imaging. For example, the average shape can be used to represent a normal anatomy within a population. Base on the average shape,

representation for the original data points as the temperature drops. Because of certain symmetries existed in the data points, some splittings may happen together. If the number of clusters are not big enough to accommodate such splittings, the clustering will not be optimal. In turn, this will also affect the matching result.

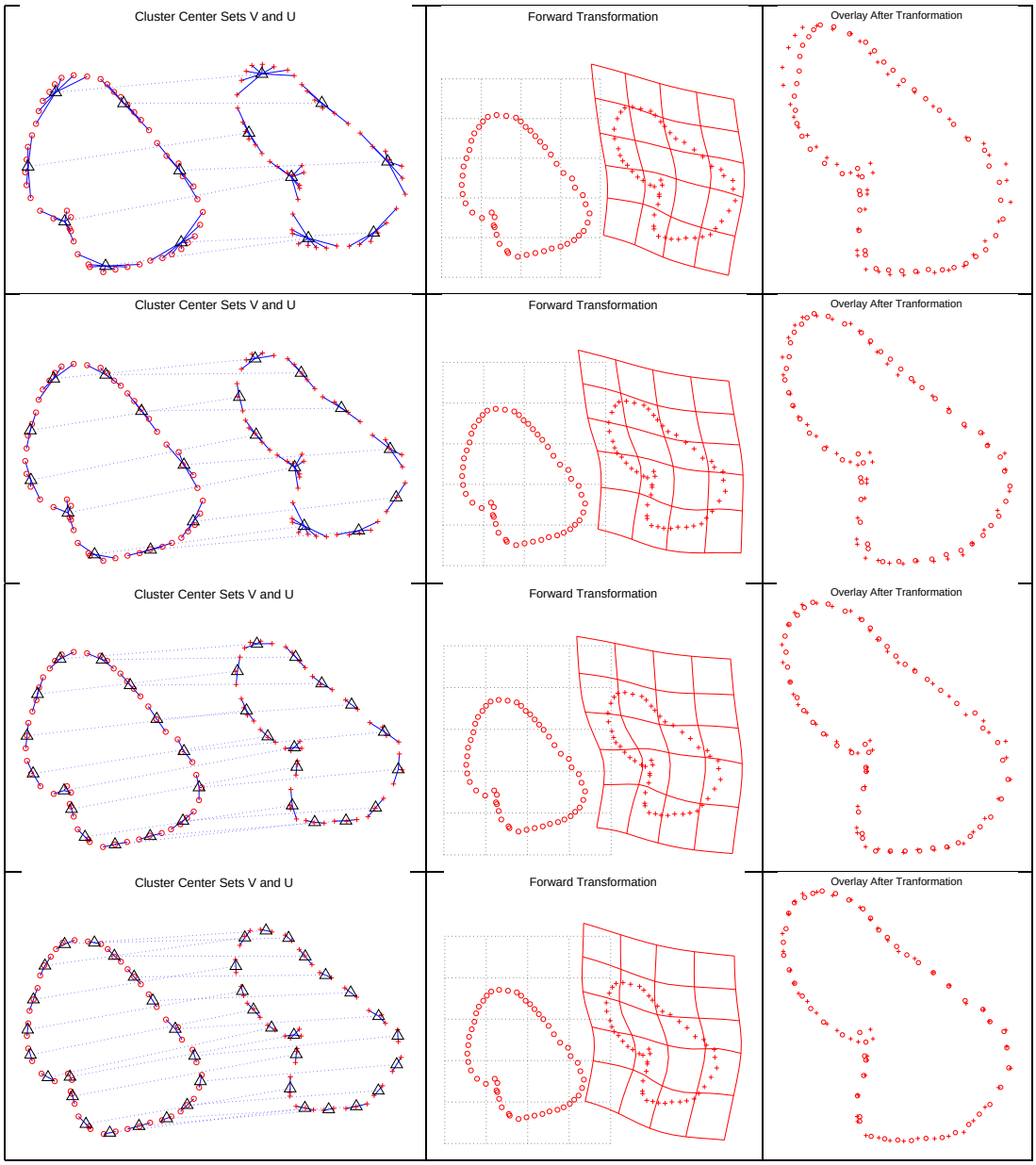

Figure 6.8: JCM with different number of cluster centers, K. From top to bottom, each row shows results using a different value of K, in the order of $7, 10, 15$ and 20. Within each row, from left to right: i) the point-sets and the cluster center sets; ii) the estimated forward transformation; iii) the overlay of the original point-sets under the estimated transformation (point-set X is being warped). Note the improvement of the alignment as K increases.

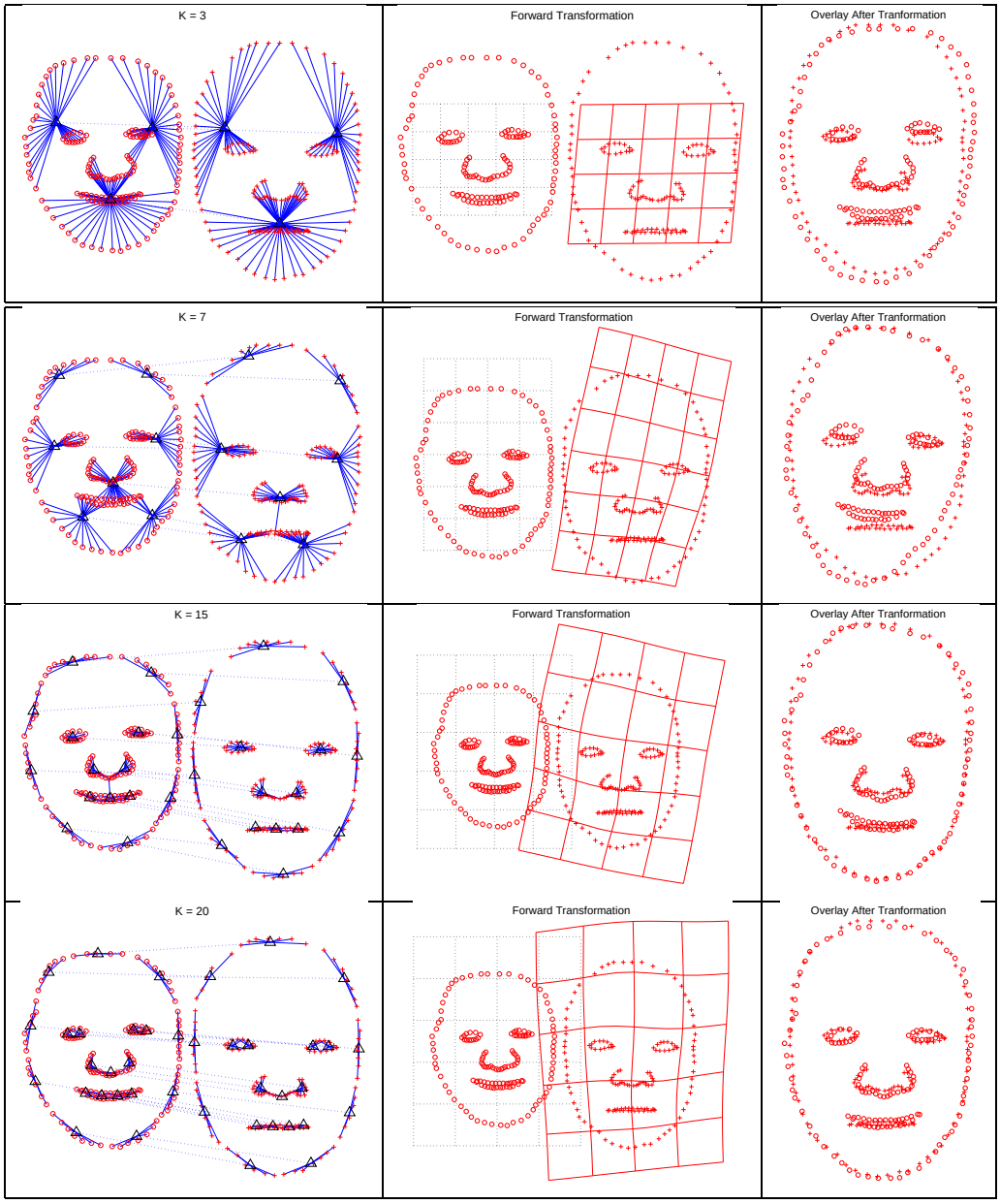

Figure 6.9: JCM used to match the face feature points. Again, from top to bottom, each row shows results using a different value of K, in the order of $3, 7, 15$ and 20. Within each row, from left to right: i) the point-sets and the cluster center sets; ii) the estimated forward transformation; iii) the overlay of the original point-sets under the estimated transformation. Note the interesting placements of the clusters as K increases.

the abnormal deviations can be identi£ed. Correlating such abnormal deviations with diseased states can potentially lead to powerful diagnostic measures. In this whole process, the key step is to establish an accurate representation of the average shape.

To be able to compute such an average shape S, the data shapes have to be properly aligned £rst. Ideally, they should all be aligned with the average set S. However, since S is unknown, this has been regarded as an intractable problem that is impossible to be solved directly [71]. Here, we show that it is not necessarily so in the case of point matching.

If we regard each point-set as a shape pattern, the shape average problem then becomes a multiple point-set matching problem: all the known point-sets are required to be matched to an average point-set, which is unknown and need to be estimated.

Herein lies the dilemma. Since the average point-set does not exist when we begin, how are we to perform the matching from each point-set to the average? Obviously, this is yet another chicken and egg problem. However, we have successfully negotiated chicken and egg problem before (in the context of correspondence and warping) by replying on the approach of alternating estimation.Would such an approach work here? Recall that in the second symmetric matching algorithm, the two sets of cluster centers are always in lock-step with the point correspondences deemed known. Since the correspondence is known, they can be used to compute an average cluster center set. Once the average set is known, we can then match all the "matchable cluster sets" to this average. The alternating update strategy will perfectly serve this purpose.

The setup is similar to the joint clustering-matching algorithm. The main modi£cation is the introduction of a new "average point-set"[3]. The original point-sets are clustered £rst to form cluster centers. The matchable cluster centers are then used to estimated the average point-set. The transformations are now put between the cluster center sets and the average point-set. Adding more point-sets to this formulation is obvious. Since this new formulation allows multiple point-sets to be matched simultaneously to an unknown average point-set, we call it the "super" clustering-matching algorithm (SCM). The setup is illustrated in Figure 6.10. For simplicity, the example shows the matching of only two point-sets ($P = 2$). Extending the setup to the matching of more than two point-sets is obvious.

The Symmetric Super Clustering-Matching (SCM) Energy Function

For generalization of the matching of arbitrary number of point-sets, we use a new set of notations for SCM. Suppose there are a total of P sets of points, $X^1, X^2, \ldots X^p, \ldots, X^P$. Each point-set is represented as X^p, consisting of

[3]Or rather, a new "average cluster set", since it is estimated from the cluster centers.

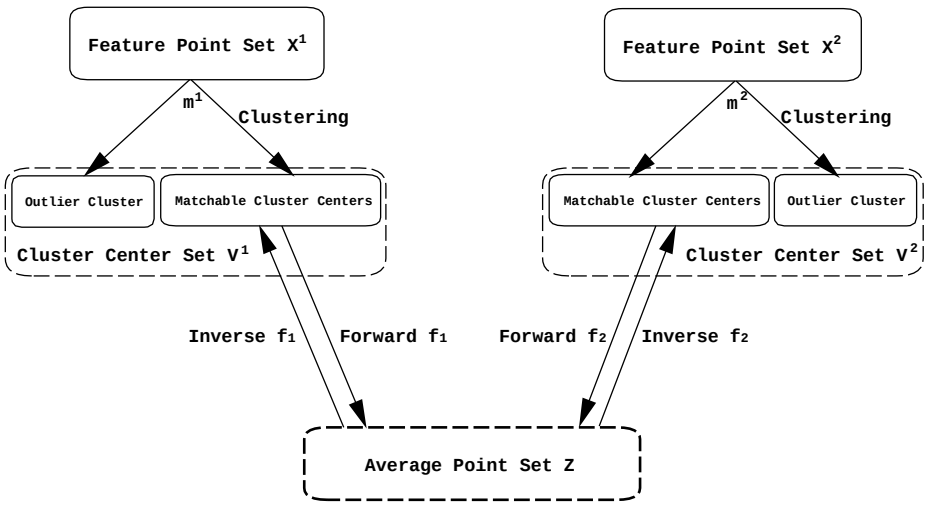

Figure 6.10: The super clustering-matching algorithm. Each original point-set (X^1 and X^2) is being clustered down to a set of cluster centers ($X^1 \rightarrow V^1$ and $X^2 \rightarrow V^2$). Each cluster set has an outlier cluster (v^1_{K+1} in V^1, v^2_{K+1} in V^2) to account for the possible spurious points in each point-set. The rest of the cluster centers ($\{v^1_a, a = 1, 2, \dots\ K\}$ and $\{v^2_a, a = 1, 2, \dots\ K\}$) are being matched to the average point-set Z.

points $\{x^p_i, i = 1, 2, \dots\ N_p\}$. For each point-set X^p, there are three related variables: i) the cluster center set V^p with K clusters $\{v^p_a, a = 1, 2, \dots\ K\}$; ii) the membership matrix M^p with entries m^p_{ai}; iii) a forward transformation f_p that warps the cluster set V^p to the common average point-set Z. The reverse transformation is optional here since the formulation is already symmetric even without it. For the sake of completeness, we still introduce the reverse transformation as the inverse of f_p here and denote it as f_p^{-1}.

The super clustering-matching objective function is almost the same as before in Equation (6.2) except that now the transformations are between the cluster sets $\{V^p\}$ and the average Z.

$$E(Z, \{V^p\}, \{M^p\}, \{f_p\}) = \sum_{p=1}^{P} E_p(Z, V^p, M^p, f_p) \tag{6.9}$$

where,

$$\begin{aligned} E_p(Z, V^p, M^p, f_p) \quad = \quad & \sum_{i=1}^{N_p} \sum_{a=1}^{K+1} m^p_{ai} ||x^p_i - v^p_a|| \\ & + \sum_{a=1}^{K} ||z_a - f_p(v^p_a)||^2 + \sum_{a=1}^{K} ||f_p^{-1}(z_a) - v^p_a||^2 \end{aligned}$$

$$+\lambda||Lf_p||^2$$
$$+T\sum_{i=1}^{N_p}\sum_{a=1}^{K} m_{ai}^p \log m_{ai}^p + T_0 \sum_{i=1}^{N_p} m_{K+1,i}^p \log m_{K+1,i}^p \tag{6.10}$$

where $m_{ai}^p \in [0,1]$ and satisfies the constraint $\sum_{a=1}^{K+1} m_{ai}^p = 1$.

Alternating Updates

In addition to the three groups of unknown variables in JCM, $\{V^p\}$, $\{M^p\}$ and $\{f_p\}$, now we have one more variable, the average point-set Z. We can derive an suitable alternating update scheme similarly.

i) Update the Membership Matrices

The membership matrix M^p is determined by each of the original data point-sets X^p and each of the currently estimated cluster center sets V^p. This has not been changed and the update is exactly the same as in JCM.

ii) Update the Cluster Center Sets

Now the clusters centers will depend on the original data points as well as the average points. Suppose that the membership matrix M^p the average point-set Z are given, we update the matchable cluster center set V^p accordingly. For example, one of the cluster center v_a^p inside V^p would be updated as the following:

$$v_a^p = \frac{1}{2}\left(\frac{\sum_{i=1}^{N_p} m_{ai}^p x_i^p}{\sum_{i=1}^{N_p} m_{ai}^p} + f_p^{-1}(z_a)\right) \text{ for } a = 1, 2, \dots K \text{ and } p = 1, 2, \dots P. \tag{6.11}$$

iii) Update the Average Point-Set

The transformation f_p bring each cluster center set V^p to match to the average point-set Z. Given the clusters centers with implicit known correspondence, we compute their average positions (after the transformations are applied) as the locations for our average point-set.

$$z_a = \frac{1}{P}\sum_{p=1}^{P} f_p(v_a^p) \text{ for } a = 1, 2, \dots K. \tag{6.12}$$

iv) Update the Transformation

The transformation is now placed between each cluster center set and the average point-set. It is again a least-squares spline £tting problem.

$$f_p = \arg\min_{f_p} \left(\sum_{a=1}^{K} ||z_a - f_p(v_a)||^2 + \lambda ||L f_p||^2 \right) \text{ for } a = 1, 2, \ldots K. \tag{6.13}$$

The Symmetric Super Clustering-Matching Algorithm

With the addition average point-set variable Z, we will have four alternating update steps. Except for that, the algorithm is pretty much the same as RPM. The pseudocode is below.

The Symmetric Super Clustering-Matching Algorithm Pseudo-code:

Initialize parameters $T = T_0$ and λ.

Initialize all membership matrices $\{M^p\}$ (e.g., use uniform matrix).

Initialize all transformation matrices $\{f_p\}$ (e.g., use Identity transformation).

Initialize the average point-set Z (e.g., use the center of mass of all point-sets).

Begin A: Deterministic Annealing.

> **Begin B: Alternating Update.**
>
> Step 1: Update cluster centers $\{V^p\}$ based on current $\{M^p\}$, $\{f_p\}$ and Z.
>
> Step 2: Update average point-set Z based on cluster centers $\{V^p\}$ and $\{f_p\}$.
>
> Step 3: Update transformations $\{f_p\}$ based on current V^p and Z.
>
> Step 4: Update clustering membership $\{M^p\}$ based on current $\{V^p\}$.
>
> **End B**

Decrease $T \leftarrow T \cdot r$ until T_{final} is reached.

End A

Example of SCM

Again, we present a simple but demonstrative example for the SCM algorithm. Given nine data point-sets (Figure 6.11), the average point-set is computed using SCM (Figure 6.12). While each of the original point-set represents a sample of the shape pattern, the resulting average point-set can be regarded as a representation of the average shape. The correspondences (or rather, the estimated membership) and the transformations between the original point-sets and the average point-set are shown in Figure 6.13 and 6.14. This example demonstrates that the SCM algorithm can truly match multiple point-sets simultaneously and compute a meaningful point-set to represent the average shape.

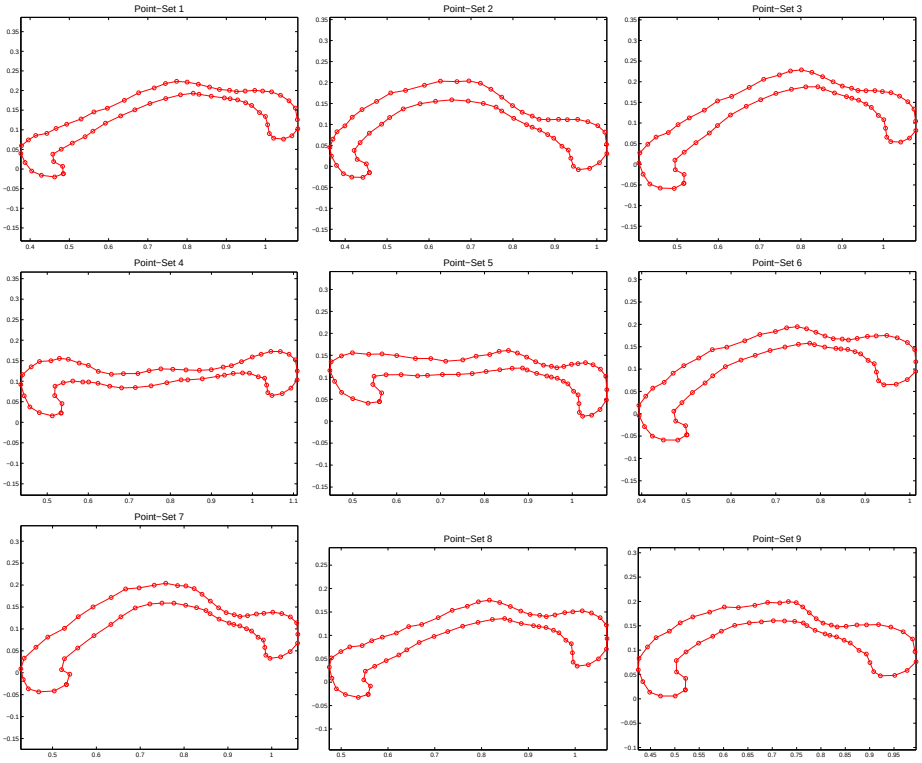

Figure 6.11: A group of point-sets for SCM to match. The data comes from one of brain structures–corpus callosum. Each point-set represents a different shape pattern. Only the coordinates of the points (circles) are known. The points connected just for better visualization. Note the shape difference between these patterns.

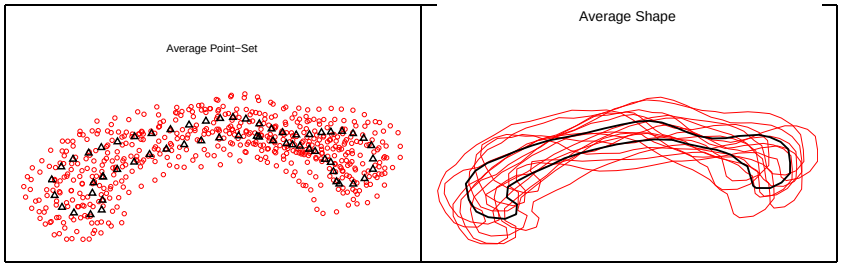

Figure 6.12: The average shape estimated by SCM. The average point-set (black triangles) is shown with the original point-sets (red circles) on the left. The corresponding line plot is shown on the right. For this example, a total number of $K = 60$ clusters are used in SCM.

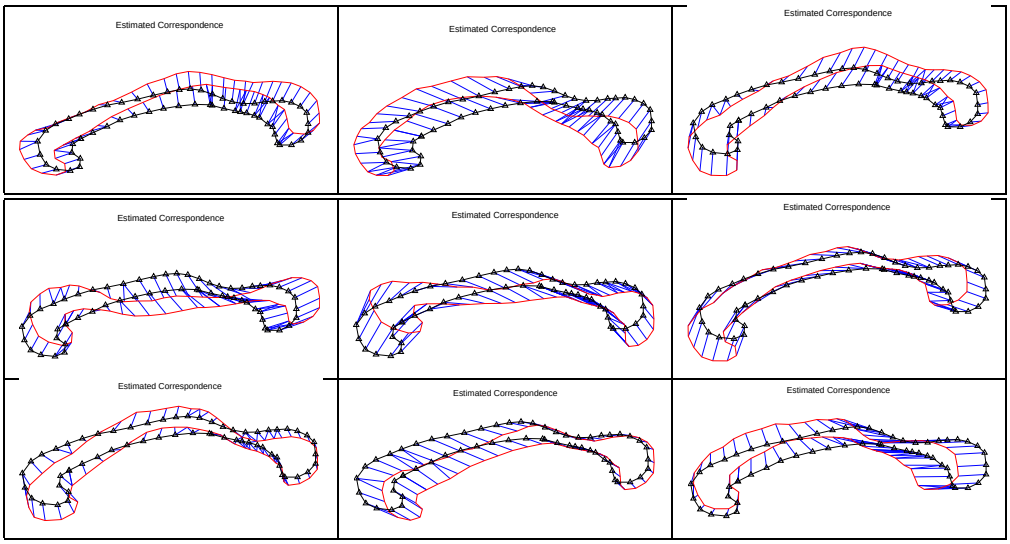

Figure 6.13: Estimated correspondences between each point-set (red line) and the average point-set (black line with triangles). Note the accuracy of the detailed matching.

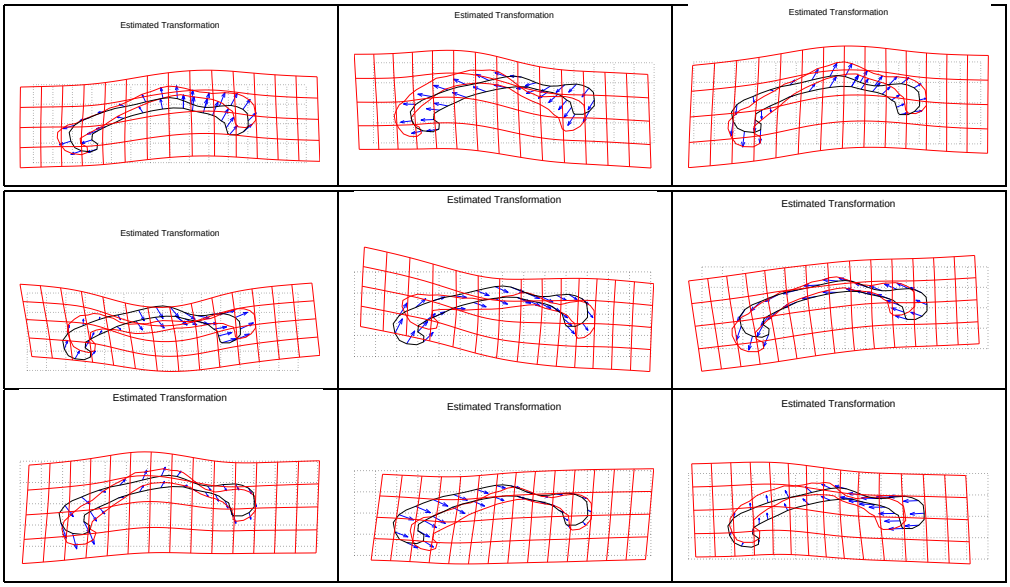

Figure 6.14: Estimated transformations between each point-set (red line) and the average point-set (black line). The transformations that warp the average onto each of the original point-sets are shown. Arrows are placed along the average shape to indicate local displacement direction (not the actual exact magnitude).

Chapter 7

Application: Brain Anatomical Registration

7.1 Introduction

In this section, we explain how we apply the the non-rigid point matching algorithm to solve an important problem in medical imaging—inter-subject non-rigid brain anatomical feature registration. We start with a careful examination of the brain registration problem.

7.1.1 Why Do We Need to Do Brain Registration ?

It is interesting to see how over time, human beings have acquire vast amount of knowledge about the world around us, but at the same time, how we still know so little about the single most important component which makes all this possible, our brains. Throughout the human history, the mechanism of the human brain and its role in the formation of memory and cognition has always remained a mystery largely unexplored. The recent development in brain imaging technologies has provided us, for the £rst time, with a chance to actually observe and study phenomena occurring at the brain level that are highly correlated with our thoughts and feelings. This kind of rich information, never before thought possible, has opened the door for quite a few new research areas. Among these areas, one of the most active is the study of human brain-mind interface.

Reason 1: Study the Structure-Function Connection

The association of brain function with speci£c anatomical brain structures is a widely held belief. A thorough understanding of such a correlation between brain anatomical structures and functionalities promises enormous bene£t for

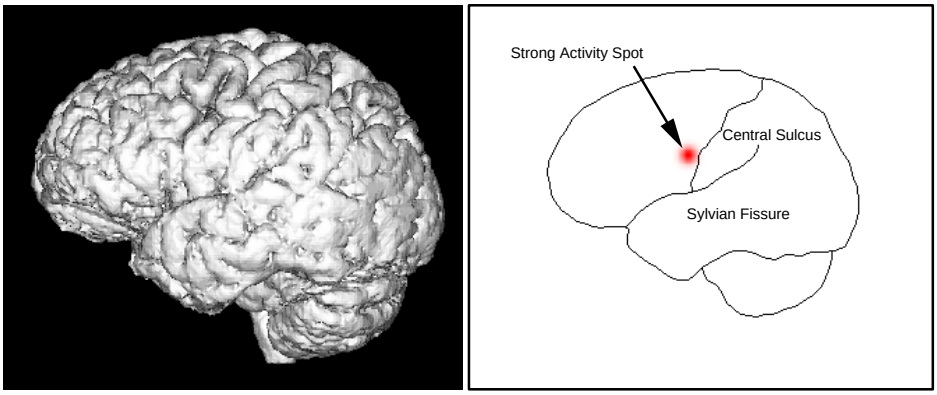

Figure 7.1: Measure the brain activity. An image of a real brain's surface rendering is shown on the left. Modern brain functional imaging can measure local activity of the brain while the subject is performing a certain functional task. Strong activity spot associated with such a functional task can then be identi£ed. The right panel shows one example where the red region represents one such strong activity spot. The drawing also shows the outline of a normal brain with some salient structure boundaries, such as the sylvian £ssure and the central sulcus, also plotted.

clinical practice. Several brain functional imaging techniques, such as fMRI, PET and SPECT, have been developed to measure the activities of the brain while it is performing a certain functional task. An simple illustrative example is shown in Figure 7.1.

The *structure-function connection* can then be established by comparing the functional activity data between different subjects. However, because of the considerable size and shape differences among the brains, direct comparison has been deemed inappropriate. This illustrated by Figure 7.2 and 7.3.

Statistically and quantitatively more accurate studies require the brain data to be placed into a common coordinate system with the anatomic variabilities removed [85]. In other words, we need to transform the brain data geometrically so that they are better aligned/registered, which, in turn, can leads to better correspondence of functionally homologous brain regions across subjects. In fact, the need to standardize functional imaging data, as explained just above, initially motivated the research work on brain registration, which is basically the process of aligning the brain data. The process is illustrated in Figure 7.4, 7.5 and 7.6.

Besides the analysis of functional data, there are several other driving forces behind the need for brain registration. We discuss here brie¤y three of them: probabilistic atlases, deformable atlases and the tracking of temporal changes.

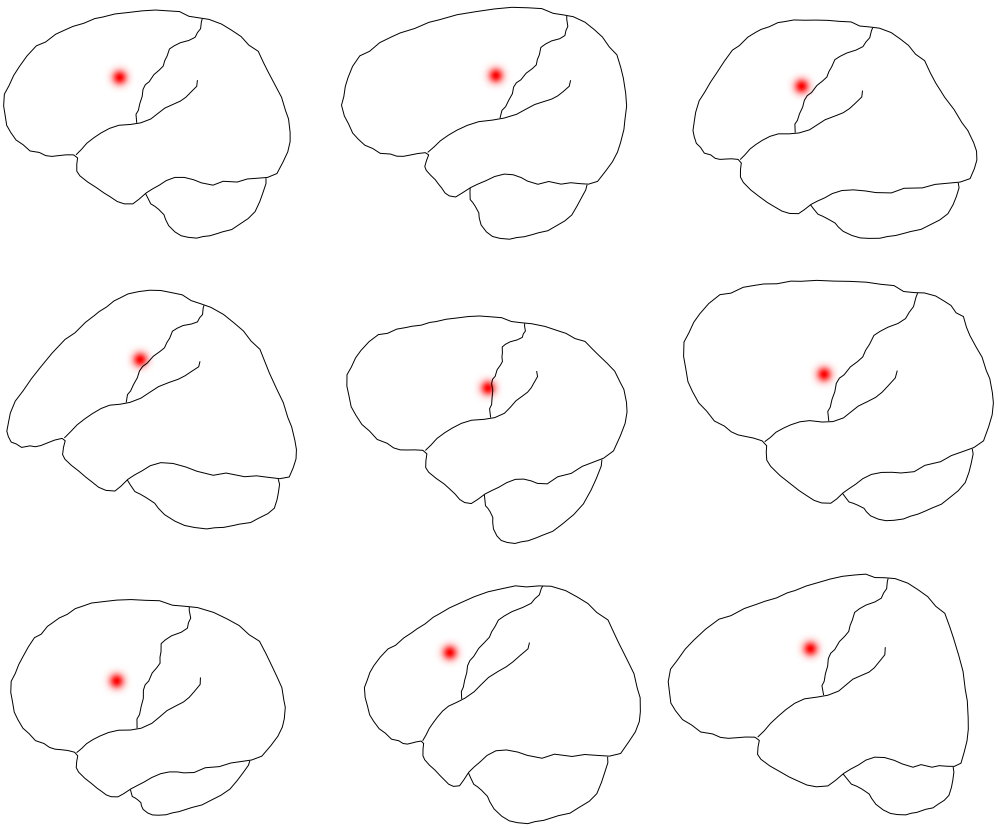

Figure 7.2: Comparison of nine different subjects' activity patterns that are related to the same functional task. The brain structural region that consistently shows strong activities across subjects very likely has strong connection with the functional task being studied. However, because of the obvious shape difference between the brains of different subjects, further analysis is dif£cult. For example, it is hard to conduct a quantitative comparison of the nine graphs shown here, even though visually it is clear the the activity is always focused at frontal brain region that is just before the central sulcus.

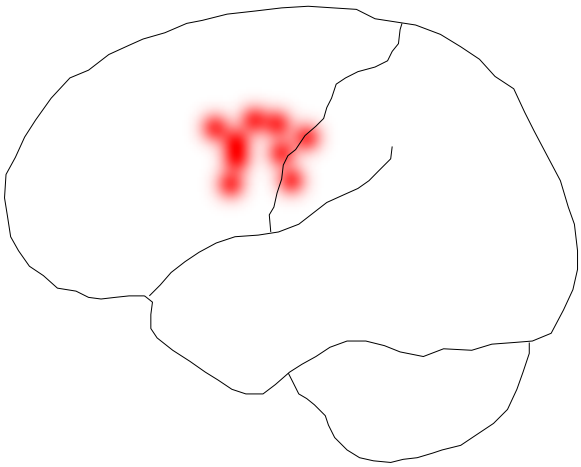

Figure 7.3: The distribution of the activity pattern without alignment. Nine different subjects' patterns are simply overlayed together and averaged. The £rst subject's brain is also shown as a reference. Without proper alignment, the variability between the brain shapes heavily contributes to an arti£cial variability in function area. This example demonstrates how the lack of proper alignment hinders the task of pooling data from multiple subjects.

Reason 2: Generation of Probabilistic Atlases

The study of neuroanatomy requires standardized coordinate systems of reference in order to facilitate quantitative analysis. This task is commonly referred to as the generation of brain atlases. While geographical atlases, such as maps, are relatively easy to construct because there is a single and constant physical object (e.g. earth) to model, the same is not true for the brain. As pointed out in [58], anatomical atlases for the brain must deal with the fact that "*there are a potentially in£nite number of physical realities which must be modeled to obtain an accurate, probabilistic representation of the entire population*". To construct such an probabilistic atlas, the corresponding brain structures from different individuals have to be aligned £rst. Again, appropriately applied brain warping(registration) algorithms are needed prior to the atlas formation.

Reason 3: Creating Deformable Atlases

Other than the probabilistic atlas which directly retain information on population variability, another way to represent the complex variations between subjects is to construct a single atlas which is more ¤exible and can adapt to the anatomy of new subjects with the help of brain warping algorithms. Such an atlas is called the deformable atlas. By

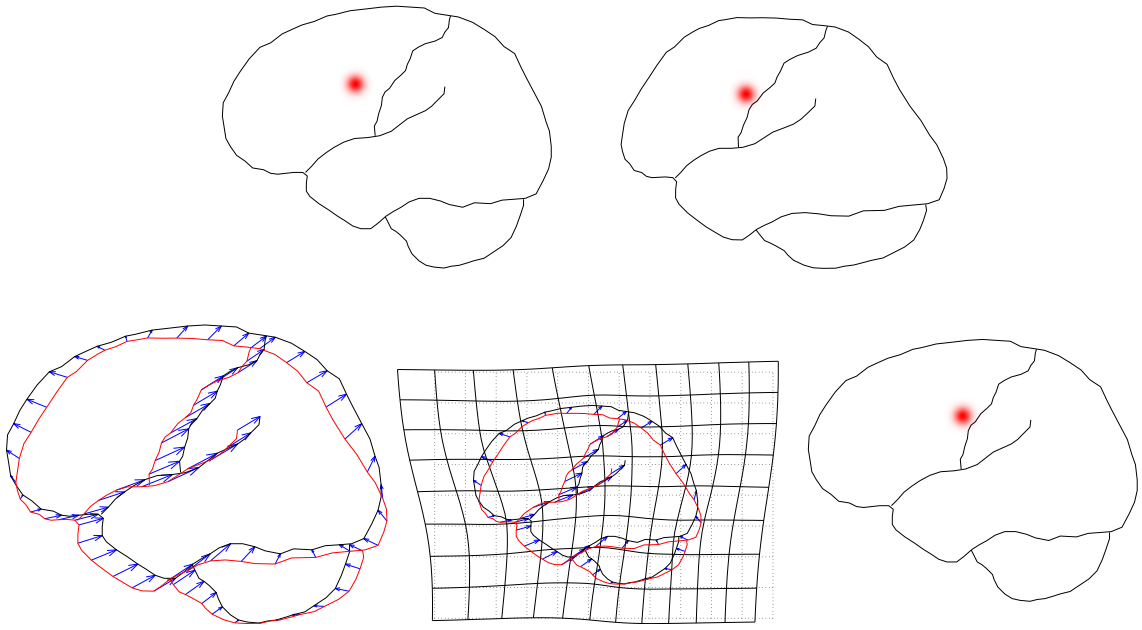

Figure 7.4: Alignment of two subjects' brain data. The originals, as shown in the £rst row, clearly have different anatomical shapes. For the purpose of alignment, we use the £rst brain as our reference. The £gures in the second row shows the alignment process, from left to right: i) the correspondence of common structures shared by both brains; ii) the non-rigid transformation that can warp the second brain to match to the shape of the £rst (the reference); iii) with the transformation, we can map the original data of the second brain so that the shape variance is appropriately reduced. Note that the second brain shows activity closer to the central sulcus. This property -re¤ects the subject's true individuality and is maintained in the normalized data.

warping the atlas to £t an incoming scan, we can transfer all the information in the brain atlas to any given subject. Among all its applications, the potential that the deformable atlas can carry pre-segmented digital anatomic models into new patients scans, automatically segmenting and labeling their anatomy, has greatly intensi£ed research in this area [3, 19, 29, 30, 15].

Reason 4: Tracking Temporal Changes

Further more, not only are two brains not identical, each individual brain also changes considerably at £rst due to ontogenesis and then gradually during the course of person's life. While the extent to which a person's experiences shape his/her brain structures is unknown, it is an undeniable fact that brain anatomical changes do occur normally

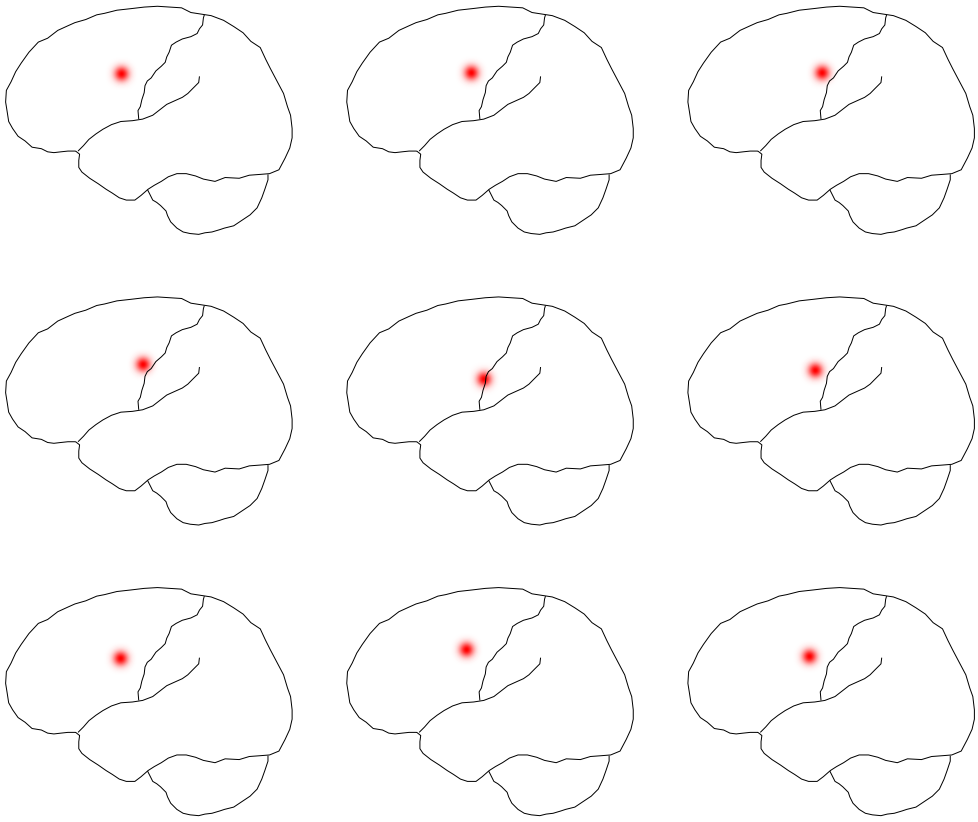

Figure 7.5: Comparison of nine different subjects' activity patterns after alignment (normalization). The difference and the similarity of the subjects are now very clear. These normalized images also provide better data for further quantitative or statistical analysis.

or due to various diseases such as Alzheimers. Studying the changes of the same person's brain structures over time provides us with valuable information for understanding the brain's normal/abnormal development.

Brain warping algorithm can be used to align two scans taken at different times and give us the deformations between them. These deformations can then be analyzed in a statistical framework to accurately detect and track subtle brain structure alterations. Using this method, a recent study by [82] indicates that during the early stage of brain development (age 3-15), different regions of the brain have different growth rate. For example, they £nd that the center of the brain, where most of the language skill related structures are located, grows signi£cantly faster than other parts between age 6 and 13 and not so after age 13. So, after all, the widely shared wisdom of learning new languages early in life, might have some biological reason behind it.

Apart from its research values, tracking brain changes over time has also been long regarded as a useful tool

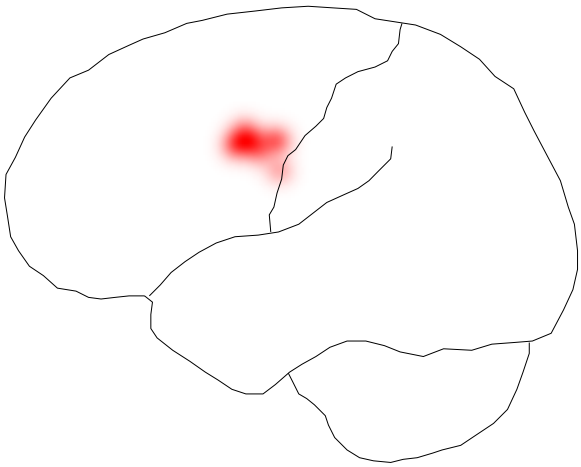

Figure 7.6: The distribution of the activity pattern after non-rigid alignment. The average is calculated after nine subjects' patterns are aligned. The £rst subject's brain is shown as a reference. Now the activity pattern become more concertrated. Non-rigid alignment plays an important role in statistically £rming up a structure-function connection, which is also more accurate (more localized) when compared to the case without alignment.

in clinical diagnosis to study, for example, the growth of brain tumors.

7.1.2 De£nition of Brain Registration

A general de£nition for brain registration can be expressed in the following way: given two brain images, £nd a good geometric transformation so that when it is applied to one of the images (the reference image), the corresponding structures between the warped reference image and the other image (the target image) are better aligned.

The two images involved in the registration process can be of many different types. According to their types, we can divide the resulting registration problem into different categories. For example, if the images are both in 2D, the corresponding registration is a 2D alignment problem. Images in 3D will then lead to 3D registration. Because of the images can comes from different imaging techniques, they can be of different modalities as well. Aligning images from the same modality results in intra-modality registration while the attempts to align images from different modalities are often called multi-modality registration. Further more, the images can come from different individuals, or from the same individual but take at different times. The former is normally referred to as intra-subject registration while the later as inter-subject registration.

The degree of geometric transformations in the brain registration can range from simple linear re-positioning (rigid) to complex nonlinear high dimensional warping (non-rigid). The higher the degree of the transformation complexity, the higher the order of variability that the transformation can handle and hence remove. A widely used example for the simple rigid transformations is the Talairach system [81], which includes only a set of rigid-body rotations and linear scalings. It has been shown that even such a simple method, which deals only with relatively low order af£ne transformations, can be helpful in analyzing functional imaging data. Clearly, non-rigid (higher order) transformations should be even more helpful if they can be reliably estimated [98]. In this work, we are mainly interested in the more complex case where high dimensional non-rigid transformations are involved.

7.1.3 How Is Brain Registration Normally Done ?

To be able to fuse information from different modalities and different subjects, a complete brain registration framework should have both the inter-subject capability and the multi-modality capability. To make the problem less dif£cult, it can be broken into two sub-problems: i) inter-subject registration of individual anatomical MRI brain images to a single reference MRI, and then ii) multi-modality registration of the other modalities (e.g. fMRI, PET) to the corresponding subject's anatomical MRI.

In this way, the £rst step only has to deal with inter-subject registration within the same image modality. Similarly, the second step only needs to handle multi-modality registration but within the same subject. The anatomical images (MRI) are used basically as the basis for the registration process. The advantage lies in the fact that MRI images normally have higher spatial resolutions and better image contrast.

7.1.4 Why Is Brain Registration Dif£cult ?

Part of the dif£culty for the brain registration problem stems from the second step of the multi-modality registration, mainly because of the fact that the images came from different modalities have quite different gray level intensity distributions. Aligning these images requires the use of robust similarity measures, which are somewhat invariant to such differences. This problem has been largely overcome by the introduction of a statistical similarity measure — mutual information, which has its origin from information theory [97, 21, 78].

The biggest challenge now is the £rst step — the inter-subject anatomical registration of the MRI images. This particular problem is the focus of this work.

The inter-subject anatomical registration is a very dif£cult task mainly due to the extreme complexity and variability present within the brain structures. Such complexity and variability are most obvious in the cortical regions.

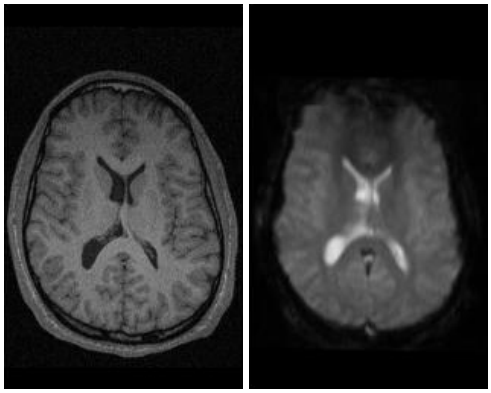

Figure 7.7: Registration of multi-modality brain images. A typical MRI image of a normal subject is shown on the left. One fMRI image from the same subject is shown on the right. Note the difference between their intensity distributions.

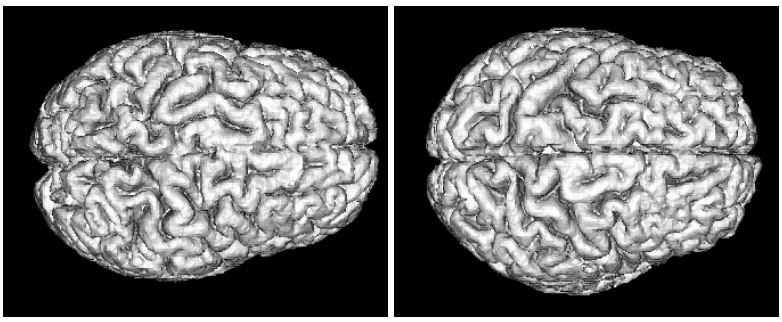

Figure 7.8: Two typical brains are shown here and the foldings (sulci) on the surfaces can clearly be seen. The complexity and variability of brain cortical structures can be appreciated.

The foldings of the cortical surfaces — the sulci and gyri — vary dramatically from person to person and, in some cases [85], are not even always present in each subject. Two typical brains are shown in Figure 7.8. From these images, the reader can appreciate how different the patterns can be.

However, the folding patterns are not completely arbitrary. Major sulci have relatively consistant shapes and often serve as important landmarks. Furthermore, there have been many associations established between the cortical areas and some of the most critical brain functionalities (vision, language, motor control etc.) with the sulci often representing important functional boundaries. So in spite of its dif£culty, a suitable inter-subject registration method that can align both the cortical as well as the sub-cortical structures is highly desired.

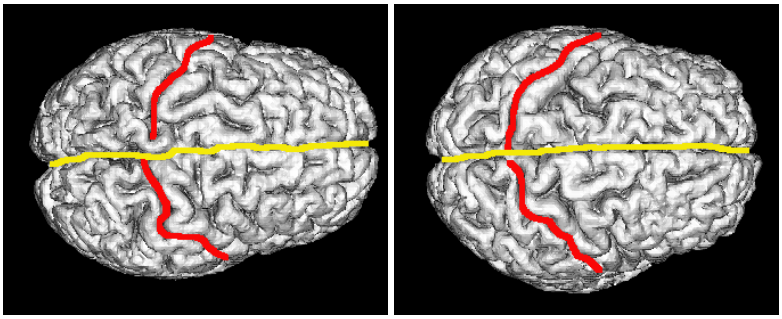

Figure 7.9: Two brains with their major sulci. For each brain, three major sulci are shown (from top to bottom of their positions): the left central sulcus, the interhemisphere £ssure and the right central sulcus. The patterns of major sulci are more consistent from brain to brain.

7.1.5 How to Overcome the Dif£culties of Inter-Subject Brain Anatomical Registration ?

The earlier methods, which only use rigid transformations to align the brains from different subjects, have long been found in-suf£cient to handle the extreme complexity and variability of the brain structures. These methods, such as the Talairach system [81], were and still are widely used mainly due to the lack of suitable substitutes.

This situation changed after the researcher started experimenting with non-rigid transformations [3]. The extra ¤exibility of the non-rigid transformation, also often referred to as warping, provides much better alignments than the rigid transformations. The recent efforts in brain registration have been trying to push the non-rigid technique forward in two directions. The £rst direction is along the line of increasing the ¤exibility. Highly complex models, such as the viscous ¤uid model, have been introduced to allow even more drastic deformations for the better alignment of two brain images, even for the small local details. The optimal deformation is usually found so that a local similarity based on the image intensity can be maximized. Another direction tries to £nd better anchors for the non-rigid registration. Geometrical features representing important and consistent structures are carefully chosen and extracted. The alignment is then done based on those features.

All the currently available inter-subject brain registration methods can be categorized into either one of these two types.

Methods completely based on raw intensity information, are termed intensity-base methods. Feature-based methods in contrast involve anatomical features in the registration process. This basic difference also re¤ect researchers' different beliefs and understandings of the brain registration problem. While the feature-based methods are more concerned with the accuracy of the non-rigid registration base on established neuroanatomy knowledge,

the intensity-based methods emphasize more the ability of powerful deformation models to overcome the structural complexity and variability. We will discuss each of them in more detail in the next section.

7.1.6 What Has Been Improved in Our Method ?

Our method basically belongs to the feature-based camp. As with other feature-based methods, we use *consistent* and *important* anatomical features to model the brain structures. The registration of the brain structures are then achieved by non-rigid alignment of the features.

However, our method improves upon the previous methods in the following ways.

i) Our method is a more uni£ed framework that can incorporate different types of geometrical feature together for the registration.

ii) Our method takes into account the *spatial inter-relationships* between different types of features by simultaneously performing the alignment in a joint homogeneous point representation space. As we will show through our experiments, the combination of different features provides important mutual anchoring information (for the features) and improves the registration.

iii) Our method provides a more symmetric formulation so that two participating images are treated equally. This is done by solving for the forward and the reverse transformations together in the matching process.

iv) The joint clustering-matching algorithm in our method provides an ef£cient way to accurately match a large number of data points while reducing the computational complexity. It also overcomes the problem of sub-sampling.

v) For the £rst time, a systematic comparison has been carried out to investigate different features' ability in brain anatomical registration.

Before we discuss the details of our method, we begin by brie¤y examining the basic elements of other current brain registration strategies.

7.2 Review

As we discussed above, most of the current efforts at inter-subject non-rigid anatomical brain registration can be broadly classi£ed into intensity-based and feature-based methods.

Intensity-based approaches try to £nd the best deformation such that an image intensity similarity measure is maximized. The formulation is very much like the optical ¤ow problem in computer vision. Most methods in

this class allow highly complex volumetric deformations in order to account for the anatomical variability. In these methods, spline models [21], elastic media models [3, 31], viscous ¤uid models [15, 14] or other local smoothness models [19, 18] are introduced as constraints to guide the non-rigid spatial mapping. All of these models draw from the fact that the image intensities are locally smooth.

Although the results of these methods clearly demonstrate the power of highly complex volumetric non-rigid deformations, a concern has been the paucity of image intensity as a feature. An important tacit assumption in intensity-based methods is that most of the brain structures are *matchable* as long as there is enough ¤exibility provided by the spatial deformation. It is already known that minor sulcal patterns on the brain cortex may not always be consistent [85], i.e., a minor sulcus in one person's brain may not exist in another person. By forcibly matching such non-corresponding anatomical structures, the extra ¤exibility afforded by the complex volumetric intensity-based deformation may make the results unpredictable and hence less reliable.

This particular problem calls for more careful treatment of different brain structures when used for the purpose of brain registration. There are major brain structures which are consistent across subjects and are important anatomically or functionally. In contrast, other minor structures may either be inconsistent, and hence not matchable, or relatively unimportant and hence should not be considered as they unnecessarily increase the registration complexity. By only using the features which satisfy both the consistency and the importance criteria, we can design a reliable method to handle the extreme variability in the brains, and also reduce the computational complexity. This brings us to the feature-based brain registration method.

Feature-based methods model the important brain structures as compact geometrical entities. The features run the gamut of landmark points [10], curves [24] or surfaces [68, 84, 83, 23, 91]. After feature extraction, these methods then attempt to solve the resulting feature matching problem (point matching, curve matching or surface matching) for the optimal spatial mapping between the features. The spatial mapping resulting from feature matching is then propagated to the whole volume. With the recent improvements in brain segmentation using deformable models [99, 105, 56, 95] and in feature extraction [92, 50, 104], more anatomical features are readily available. The question at hand is how do we fully utilize these different types of features for brain registration.

Feature-based methods, when applied individually to the brain surface, or the sulci have been devised. But in our view, not enough attention has been paid to feature integration. While each type of feature is useful in registration, most earlier feature-based methods neglect the opportunity of simultaneously taking all the features into account during registration. The integration of different features is important. The spatial inter-relationship existed between different features can provide valuable information for the alignment. So far, it has been largely neglected. This problem can

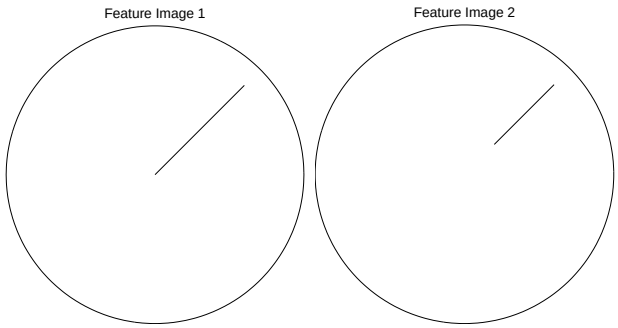

Figure 7.10: An example of using feature inter-relationship to better anchor registration. Two feature images shown here need to be aligned. Each contains two types of a features: a circle and a line segment. Suppose we are only trying to solve a rigid 2D transformation (rotation plus translation) in this case. If we use the circle feature alone, we would not be able to determine the proper rotation. On the other hand, if we rely solely on the lines, the translation would then be uncertain. Both of these problems can be solved by taking into account of both features together. This simple example demonstrates how the inter-relationship between the features helps to anchor the registration.

be illustrated by a simple example as shown in Figure 7.10.

More recent research efforts have begun to capitalize on such information. For example, to improve the cortical alignment, incorporation of sulcal features into the overall matching framework has attracted much attention. In [68], [83] and [91], major corresponding sulcal curves are used along with the outer cortical surface. By enforcing the alignment of corresponding sulcal curves, it was shown that the alignment of the cortex was improved. However, these methods cannot be applied in more general feature-based registration situations where multiple, disconnected surfaces are present.

We propose a more general framework to attack this problem. The essential idea is in fact quite straightforward. To achieve the combination and subsequent joint registration of different types of features, we fuse them together into a common point representation. After the features are combined (while preserving their distinct anatomical labels), we the only need to solve the joint point matching problem to align them.

To demonstrate this idea, we choose in this work two types of surface-based features — the smoothed outer cortex surface [105] and several major sulcal ribbons [104]. After these features have been extracted, we fuse these two different type of features together by converting them into a common point representation. The ¤exibility of using point representations easily overcomes the otherwise dif£cult problem of feature data fusion. To solve the resulting matching problem between the hundreds of resulting points, we devised a new iterative joint point clustering and

matching algorithm, which is closely related to our previous work on robust point matching (RPM) [36, 65, 16, 17]. The clustering is carried out simultaneously during the estimation of the deformation so that the accuracy is improved while the computational complexity is greatly reduced.

7.3 A Uni£ed Feature Registration Method

The overall scheme of our method can be divided into three stages:

1. Feature Extraction: Choose and extract neuroanatomical features.
2. Feature Fusion: Fuse these features into a common point-set.
3. Feature Matching: Solve for the spatial deformation between two feature point-sets through point matching.

We explain each stage below.

7.3.1 Feature Extraction

At present, we choose the outer cortical surface as well as major sulcal ribbons as the dominant features. The outer cortical surface has been widely used for brain registration since it provides a very good model for the global shape of the brain. Automated extraction of the brain surface is based on the coupled surface-based brain segmentation method as described in [105]. Since the minor sulci cannot be expected to be consistent across subjects, we decided to further smooth the cortical surface. The smoothing is done so that the surface still closely wraps over the brain but with all the sulci £lled up (as shown in Figure 7.11). The major sulci are chosen because of their well known importance and relative consistency across individuals. Instead of using curves (as in most previous methods [68, 83, 91, 16]), we use ribbons, which provides a better representation of the sulci's deep 3D structure (as shown in Figure 7.11). The interactive extraction of these ribbons is done with the help of a ribbon extraction method developed in [104].

7.3.2 Feature Fusion

Once all the features are extracted, we have a set of different 3D surfaces in the form of the closed outer cortical surfaces and open major sulcal ribbons. We then run a sampling procedure to convert each of the feature surfaces, which are parameterized as polygonal meshes, into points. This is done by dividing the space into small cubes of equal sizes, and then computing the average of all the vertices lying inside each cube. The average point-set for each feature surface are then combined into a super feature point-set, which is used as the common point-based representation for

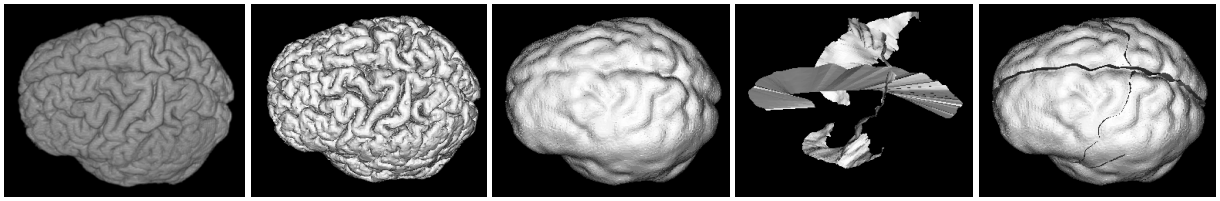

Figure 7.11: Feature extraction. From left to right: 1) original MRI brain volume; 2) extracted outer cortical surface without smoothing; 3) smoothed outer cortical surface; 4) extracted major sulcal ribbons; 5) the outer cortical surface together with the sulcal ribbons.

the purpose of registration. The fusion/sampling process is demonstrated in Figure 7.12. In this work, the £nal super point-set would typically has around 2,000 points in it.

7.3.3 Feature Matching via the Joint Clustering-Matching Algorithm

After £nishing the £rst two steps, we have a complex 3D point-set, which is comprised of hundreds of points, to represent each brain. We then use the joint clustering-matching (JCM) algorithm developed earlier in this thesis to solve for a good non-rigid deformation that aligns the point-sets.

The choice of using the joint clustering-matching (JCM) algorithm, instead of using the original robust point matching algorithm (RPM), is based on several practical considerations. Matching and evaluating non-rigid transformations between thousands of points is computationally very expensive. As we discussed before, the JCM algorithm provides an ef£cient way to cut down the computation cost through clustering. The clustering is carried out jointly with the matching process, ensuring consistent placement of the cluster center to avoid the subsampling problem. There is, yet, another way to appreciate the JCM algorithm, which also counted as an important factor to convince us of using JCM.

Joint Estimation of the Deformation and its Control Points

In order to de£ne the non-rigid deformations (usually in the form of splines) for the alignment of the points, we are usually required to specify a set of *spatial control points*. The choice of the control points is vital because they directly affect the behavior of the deformation. First, their right placement provides necessary ¤exibility for the spline/deformation to capture the variability of the data. However, too ¤exible a deformation is unstable and will be easily disturbed by small amounts of noise and possible outliers. So, the control points should also provide a reasonable amount of regularization for the deformation as well. These two con¤icting requirements make the choice

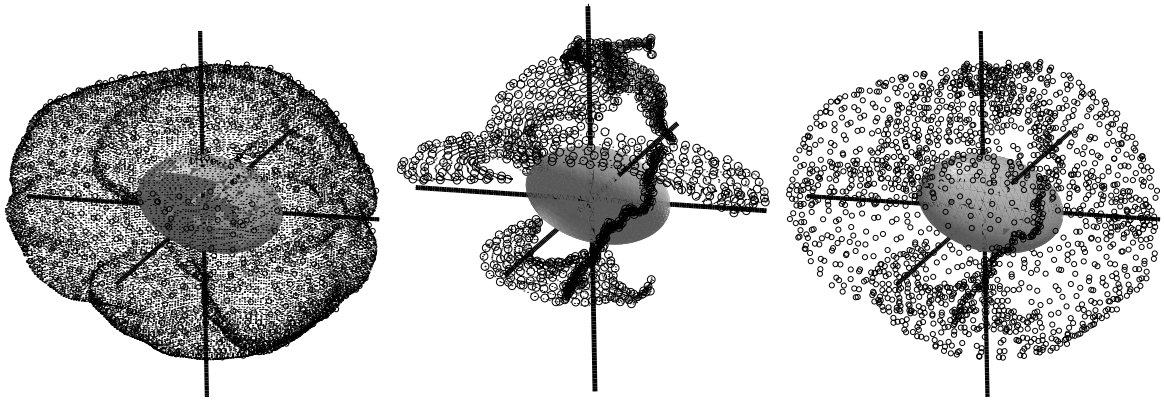

Figure 7.12: Feature fusion / sampling. From left to right: i) sampling the outer cortical surface vertices. The original dense surface vertices are shown as dots and the sampled average points are shown as circles. The middle solid ellipsoid is just for visualization purposes; ii) sampling the sulcal ribbons. iii) the super feature point-set. The outer cortex is reduced from the order of 50,000 ˜ 80,000 original vertices to approximately 1000 average points. The ribbon surfaces are sampled with smaller sized cubes since they represent much smaller structures. Each ribbon, with originally about 1000 ˜ 2000 vertices, is eventually represented by roughly 100 ˜ 200 average points.

of the control points dif£cult. Since we usually do not have much *apriori* information about the deformation, using prede£ned control points is problematic and arbitrary. Instead, we can include the control points as unknown variables, along with the deformation, in our matching algorithm.

Based on the observation that the control points can better accomplish their purpose when placed in the more densely distributed data areas, and also in the areas where the deformation is more complex, we can cluster the data and use the cluster centers as our control points for the deformation. Clustering is done during the process of estimating the deformation, which ful£lls the requirement that there be feedback of information from the deformation estimation into the estimation of the control points. The idea is demonstrated in Figure 7.13.

Joint Clustering-Matching Energy Function

Since the features used in this work for the brain registration task are all matchable, we can slightly simplify the previous JCM framework by taking out the outlier terms. More formally, the joint clustering and non-rigid deformation

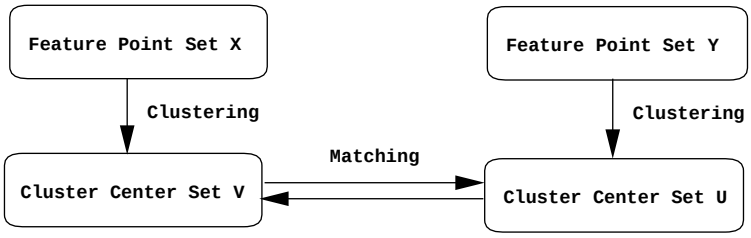

Figure 7.13: The joint clustering-matching algorithm. Each of the two point-sets to be aligned is clustered to achieve a set of cluster centers, which are also used as control points to de£ne the deformation. The cluster centers serve a dual purpose. They provide not only a concise representation of the original point data but also an optimal control point-set for the deformation.

estimation framework is equivalent to the minimization of the following objective function.

$$\begin{aligned} E \quad (v,u,f_x,f_y,m^x,m^y) = \quad & \sum_{i=1}^{N_x}\sum_{a=1}^{K} m_{ai}^x ||x_i - v_a||^2 + \sum_{j=1}^{N_y}\sum_{a=1}^{K} m_{aj}^y ||y_j - u_a||^2 \\ & + \sum_{a=1}^{K} ||u_a - f_x(v_a)||^2 + \sum_{a=1}^{K} ||v_a - f_y(u_a)||^2 \\ & + \lambda ||Lf_x||^2 + \lambda ||Lf_y||^2 \\ & + T\sum_{i=1}^{N_x}\sum_{a=1}^{K} m_{ai}^x \log m_{ai}^x + T\sum_{j=1}^{N_x}\sum_{a=1}^{K} m_{aj}^y \log m_{aj}^y \end{aligned} \tag{7.1}$$

where $m_{ai}^x \in [0,1]$ and $m_{aj}^y \in [0,1]$ satisfy the constraints:

$$\sum_{a=1}^{K} m_{ai}^x = 1, \; for\, i = 1, 2, \ldots \; N_x, \tag{7.2}$$

$$\sum_{a=1}^{K} m_{aj}^y = 1, \; for\, j = 1, 2, \ldots \; N_y. \tag{7.3}$$

The formulation is almost the same as before except the absence of the outlier terms. To refresh our memory, we brie¤y explain the notation.

We begin with two point-sets. The reference point-set X (which we seek to warp onto a target) has N_x points. The target point-set Y has N_y points. Note that N_x can be different from N_y. We represent them as $\{x_i, i = 1, 2, \ldots, N_x\}$ and $\{y_j, j = 1, 2, \ldots, N_y\}$, while each x_i (and each y_j) represents a single point location in 3D.

A set of cluster centers (with a total of K) is associated with each point-set. The cluster centers arising from the reference point-set X is called V, consisting of K centers $\{v_a, a = 1, 2, \ldots, K\}$. The cluster centers arising from the target point-set Y is called U, consisting of corresponding K centers $\{u_a, a = 1, 2, \ldots K\}$. The variables m^x and m^y represent the clustering membership information between the data and the clusters. For example, if $m^x_{ai} = 1$, x_i belongs to the a^{th} cluster v_a. The membership variables are introduced as intermediate variables here since they enable us to calculate the cluster center locations from the original feature point locations.

Again, the framework is generally applicable for any type of deformation parameterization, we use the abstract notation f_x and f_y for the deformation functions (with the actual parameterization yet to be speci£ed). We call the deformation that maps the reference data to the target data the forward deformation f_x, and the deformation in the opposite direction the reverse deformation f_y. When applied to a point v_a, the deformation f_x maps the point to a new location $f_x(v_a)$. The regularization measures, which normally accompany the non-rigid deformations, are represented as $||Lf_x||^2$ and $||Lf_y||^2$. At present, we do not enforce the constraint that f_x and f_y be inverses of one another (such as in [14]) but such a constraint can be enforced if necessary.

The rest of the terms come from deterministic annealing.

Deterministic Annealing Revisited

Some interesting properties of membership variables can give us some more insight into the workings of DA. Our clustering membership variables m^x and m^y are continuous variables in the interval $[0, 1]$, which still satisfy the constraints that the total membership of each data point in all clusters is one.

The continuous value of the membership variable re¤ects the "fuzziness" in our clustering model. For example, if all m^x_{ai} are the same, the membership of a data point in a cluster center is uncertain. A more careful calculation shows that this effectively causes all the cluster centers to lie at the center of mass of the data point-set. At the other end of the spectrum, if all m^x_{ai} are close to binary values (either 0 or 1), each cluster center will represent a separate subset of the data points resulting in a good representation of the shape of the data point-set. Between these two extremes, the membership matrix m^x_{ai} of some intermediate fuzziness generates a set of cluster centers which can capture the shape of the data points at some intermediate level.

The shapes of the intermediate levels are very helpful in our quest for a good non-rigid deformation. They are much simpler than the actual data shape, which make them easier and more stable to match. On the other hand, they resemble the actual data shape to some extent. So the answer for the deformation found at a lower less detailed level can be used as a good initialization for the deformation at a more detailed level. If we have a way to gradually

reduce the fuzziness under a controlled manner while progressively improving our estimation of the deformation, it obviously leads roughly to a coarse-to-£ne, scale-space like strategy. Again, the £ne control of the fuzziness can be simply achieved by gradually reducing the temperature parameter T.

The Simpli£ed Joint Clustering-Matching Algorithm (SJCM)

Without the outlier components, the joint clustering-matching algorithm implemented for brain registration is even simpler than before with only two iterative update steps: a clustering step and a matching step. We merged two of the previous update step (step 1 and step 3) in JCM together to form a clustering step. The matching step is kept the same as the step (step 2) that updates the transformations.

The update of each variable is calculated basically by differentiating the energy function (7.1) w.r.t. that variable and setting the result to zero. Again, the iterative update is embedded within an annealing scheme.

Step 1. Clustering: Update membership matrices and cluster centers.

$$m_{ai}^x = \frac{q_{ai}^x}{\sum_{a=1}^K q_{ai}^x},\ m_{aj}^y = \frac{q_{aj}^y}{\sum_{a=1}^K q_{aj}^y}, \tag{7.4}$$

where,

$$q_{ai}^x = e^{-\frac{||x_i - v_a||^2}{T}},\ q_{aj}^y = e^{-\frac{||y_j - u_a||^2}{T}}, \tag{7.5}$$

and then,

$$v_a = \sum_{i=1}^{N_x} \frac{m_{ai}^x x_i}{2} + \frac{f_x(u_a)}{2},\ u_a = \sum_{j=1}^{N_y} \frac{m_{aj}^y y_j}{2} + \frac{f_y(v_a)}{2}. \tag{7.6}$$

Step 2. Matching: Update the deformation function. This is a standard least-squares spline £tting problem.

$$f_x = \arg\min_{f_x} \left(\sum_{a=1}^K ||u_a - f_x(v_a)||^2 + \lambda ||L f_x||^2 \right), \tag{7.7}$$

$$f_y = \arg\min_{f_y} \left(\sum_{a=1}^K ||v_a - f_y(u_a)||^2 + \lambda ||L f_y||^2 \right). \tag{7.8}$$

As we explained before, the update presented for the cluster centers has been slightly simpli£ed. The deformations

are held £xed when the cluster centers are updated despite the fact that the deformation is actually a function of the cluster centers. Since the deformations are slowly re£ned during the entire iterative procedure, this approximation is justi£ed and considerably simpli£es the calculation.

The summary of the SJCM algorithm is as follows.

Simpli£ed Joint Clustering-Matching Algorithm Pseudo-Code:

Initialize T, f_x and f_y.

Dual Update:

- Clustering step to update m_{ai}^x, m_{ai}^y, v_a and u_a.
- Matching step to update f_x and f_y.

Lower $T \leftarrow T \cdot r$ until T_{final} is reached.

Choice of Splines to Model the Deformation

We now specify the deformation parameterization in order to complete the algorithm speci£cation. We implemented two types of radial basis function splines [94].

Given a set of control points $\{v_a, a = 1, 2, \ldots n\}$, a radial basis function basically de£nes a spatial mapping which maps any location x in space to a new location $f(x)$, represented by,

$$f(x) = \sum_{a=1}^{n} c_a \phi(||x - v_a||) \tag{7.9}$$

where $||\cdot||$ denotes the usual Euclidean norm in 3D and $\{c_a\}$ is a set of mapping coef£cients. The kernel function ϕ assume different forms. If we choose $\phi(r) = \exp(-r^2/\sigma^2)$, it becomes a Gaussian Radial Basis Function (GRBF). The parameter σ controls the locality of each kernel function. A small value of σ generates more localized and hence less smooth warping. A different choice of $\phi(x) = r$ leads to another type of radial basis function called the Thin Plate Spline (TPS). Compared to the GRBF, TPS has a more global nature—a small perturbation of one of the control points always affects the coef£cients corresponding to all the other points as well.

The details for the solution of TPS and GRBF are included in Chapter 3 and the Appendix section. Here we would like to just point out that there are closed form analytic solutions available for both GRBF and TPS [94] for the spline £tting problem, as formulated in (7.7) and (7.8).

7.4 Experiments and Results

We conducted experiments on both synthetic and real data to evaluate our algorithm. Because of the lack of ground truth information for real inter-subject brain registration, the synthetic data provides a good alternative for validation purposes. We intend to investigate the following questions: i) does the fusion of different types of features improve the registration or not? ii) if it does, what is the degree of improvement ?

7.4.1 The Design of the Synthetic Experiment

The synthetic experiments for registration are normally carried out in the following steps:

1. Construct a template.
2. Construct a target from the template via a synthetic deformation (ground truth).
3. Recover a good deformation (via the algorithm) to match the template to the target.
4. Examine the errors between the solution and the ground truth deformation.

Construction of the Template

We choose one normal male brain MRI (without the skull) as the raw data for our template. For the purpose of registration, as explained before, the smoothed outer cortex surface and a set of major sulcal ribbons are extracted as features.

We need to prepare the template for later error measurement as well. Apart from these features used for the registration, we asked a neuroanatomy expert to extract a different set of landmark points over the whole brain volume to get a rough error measurement. The landmark points include two sub-groups: one group distributed on the outer cortex surface and another group distributed at critical locations of the sub-cortical structures (as shown in Figure 7.14. The idea is that while the error calculated over the whole landmark point-set will give us a rough global estimate of how good the alignment is, the errors from each sub-group can tell us a little bit more about where the error comes from.

To provide even more detailed measurement than landmarks, we also have the MRI volume fully segmented and labeled with the help of a neuroanatomy expert. A label is assigned to each voxel according which structure the voxel belongs to. A full hierarchy of structures are used. The volume is £rst segmented into background and the brain. The brain part is then divided into cortex and sub-cortex. The cortex is subdivided into different lobes with gray/white

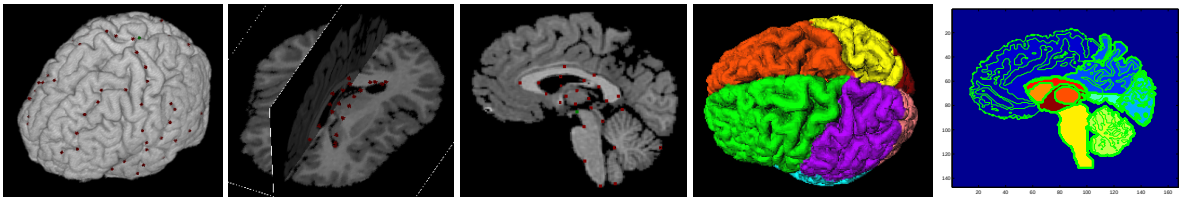

Figure 7.14: Template for the Synthetic Study. From left to right: 1) the cortical landmarks; 2) the subcortical landmarks in 3D view; 3) the sub-cortical landmarks again in a 2D view; 4) the fully segmented brain MRI volume; 5) a 2D slice showing the segmentation.

matter segmentation performed for each lobe. Within the sub-cortex volume, a list of important sub-cortical structures are segmented including: thalamus, caudate, putamen, brain stem and the ventricles. With this £nely segmented brain volume template, we can then make very detailed error measurement.

Using GRBF as Synthetic Deformation to Construct the Target

Based on the same consideration of the possible prior bias, we used one spline (GRBF) to construct the synthetic target data and a different spline (TPS) for recovery. The GRBF spline is chosen as the synthetic deformation to generate the target data mainly because GRBF can easily generate both local and global warping (Figure 7.15) with its locality parameter σ. For the recovery, we chose TPS and speci£ed the total number of cluster centers K to be 150 to give TPS enough ¤exibility without incurring too much computational cost. The value of the regularization parameter λ is chosen by hand. A matching example of using both the outer surface and the sulcal ribbons is demonstrated in FigureStandard

Using TPS to Recover the Deformation

We choose TPS as the deformation module in our feature registration/point clustering-matching algorithm. Though using TPS, instead of GRBF again, obviously makes the problem more dif£cult, it is necessary since it provides more unbiased evaluation for the algorithm. The advantage of using TPS is the smaller number of free parameter compared to GRBF. Apart from the annealing parameters, the only extra parameter that TPS needs is the smoothness weight parameter λ. The value for λ is chosen by hand. The annealing parameters are set as discussed before. We specify the total number of cluster centers K to be 150 to give TPS enough ¤exibility while without taking too long to compute.

We run the algorithm with different settings to see if the combination of features really improve the registration or not. Three different choices of features are compared— the outer surface alone, the sulcal ribbons alone and

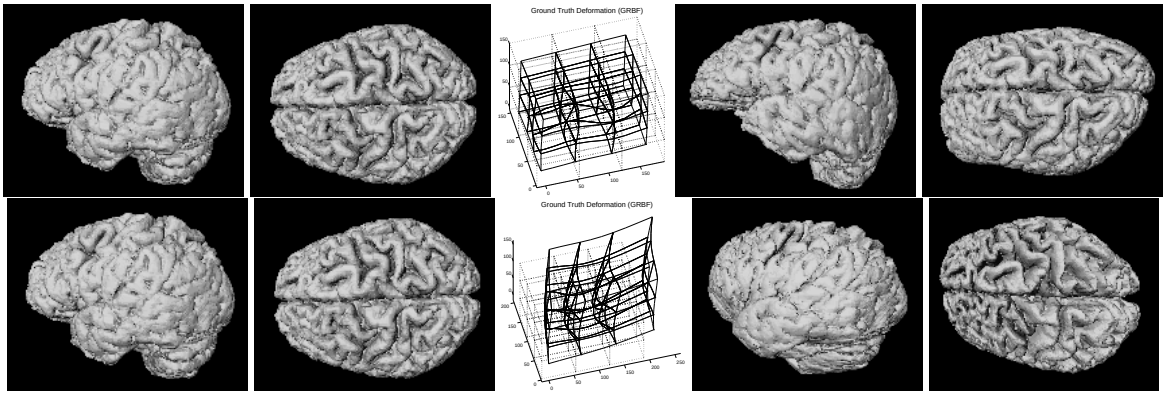

Figure 7.15: Ground Truth GRBF (Local Warp vs. Global Warp). **Top Row**: from left to right: 1,2) the original brain volume; 3) a randomly generated GRBF with a small $\sigma = 30$ (local warp). The original space is shown by the regular dotted 3D grid and the warped space by the solid deformed grid; 4,5) the warped brain volume by GRBF. **Bottom Row**: same as the top row except that a GRBF with a big $\sigma = 60$ (global) warp is used.

the combination the outer surface and the sucal ribbons. One matching example of using both the outer surface and the sulcal ribbons is demonstrated in Figure 7.16 and 7.17.

Examination of the Errors

Errors are calculated based on both the landmarks as well as the fully segmented and labeled volume. To measure the errors on landmarks, we first warp the original landmarks with the ground truth GRBF to get the ground truth set A. After the TPS registration, we warp the original landmarks again with the TPS recovered by the algorithm to get the solution set B. Errors are then calculated between A and B as the Euclidean distances between the two warped landmark point-sets. A similar procedure is performed for the labeled volume as well. The only difference between the landmark error and the volume error measurements is that in the latter, the error is measured as the ratio of misaligned voxels between the two warped labeled volumes for a certain segmented structure. First the misaligned voxels are counted for a particular structure. The error ratio is calculated by dividing such the misaligned count by the actual total number of voxels in that structure. Such a error ratio basically provides a relative measurement for the alignment of each structure.

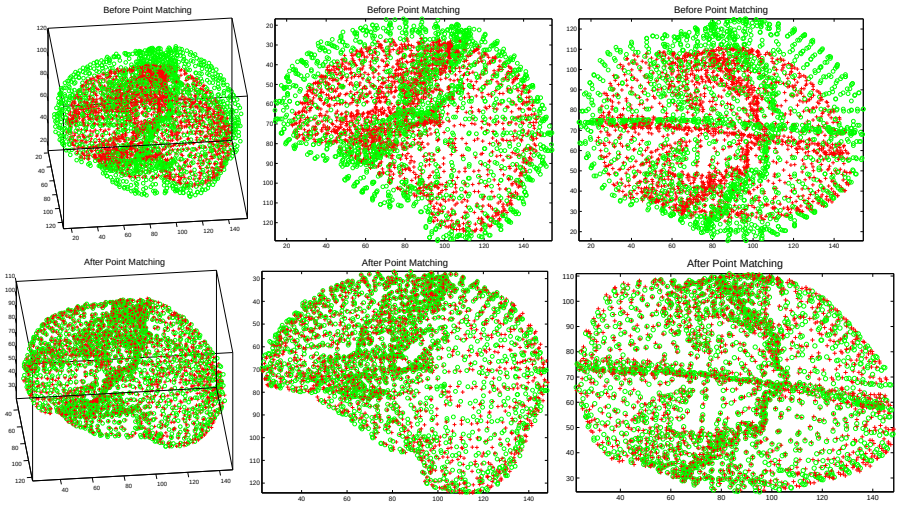

Figure 7.16: Joint point clustering-matching using TPS. **Top Row**: 2 feature point-sets before the point matching and the ground truth GRBF. One point-set is shown as cross and another as circles. **Bottom Row**: the feature point-sets after the point matching with the recovered TPS deformation.

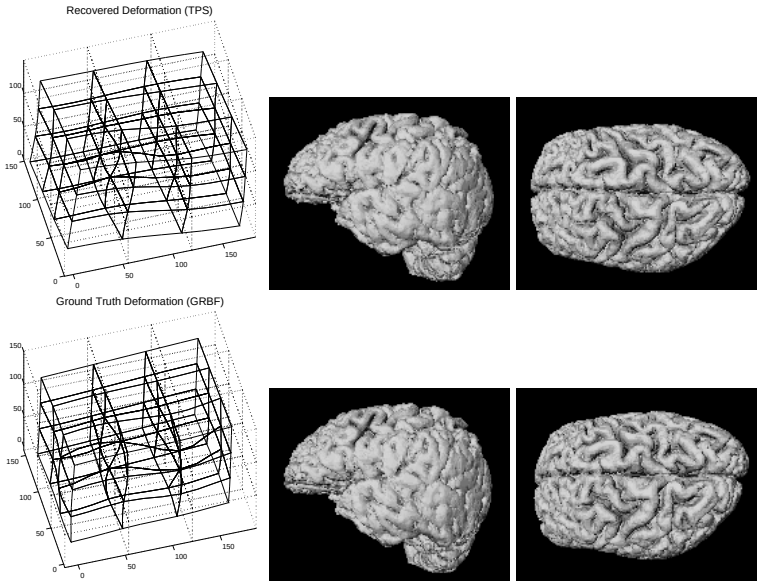

Figure 7.17: Comparison of TPS and GRBF. **Top Row**: 1) TPS deformation recovered from the point matching; 2,3) the warped brain volume using the TPS. **Bottom Row**: 1) the ground truth GRBF deformation; 2,3) the warped brain with the GRBF. Note that though the warped volumes from TPS and GRBF look almost the same on the surface, the two deformations are still slightly different over the whole space. It shows that the two splines have different behaviors.

(Mean/Std., Unit: Voxel)	Method I	Method II	Method III
All Landmarks	2.83/0.61	1.96/0.29	1.58/0.30
Cortical Landmarks	2.23/0.48	1.88/0.29	1.15/0.30
Sub-Cortical Landmarks	3.46/0.97	2.05/0.46	2.03/0.38

Figure 7.18: The error statistics based on the landmarks of test series 1 (local GRBF warp). Method I: using outer cortex surface alone. Method II: using sulcal ribbons alone. Method III: using both together. Method III gives the smallest errors. Also note that the cortical landmark errors tend to be smaller than those of the sub-cortical landmarks.

7.4.2 Experiments and Results

We carried out 2 series of synthetic experiments: one with a smaller value of $\sigma = 30$ in GRBF for more localized warping and one with a larger value of $\sigma = 60$ for global warping. Ten randomly generated trials (with randomly generated GRBF coef£cients) are included in each series. The algorithm is run 3 times for each trial; each time with a different choice of feature—outer surface alone (method I), sulcal ribbon alone (method II) and the combination of outer surface and sulcal ribbons (method III). The errors for each method are averaged over the 10 trials to get both the mean and the standard deviation. The error statistics are shown in Figure 7.18 and 7.19.

Evaluation Based on Landmarks

From the error statistics based on landmarks (FigureLatexCommand “ref{£g:error␣landmark}) Since all our features are mostly located in the cortical regions, it is not surprising that the alignment of the cortical landmarks tends to be much better than that of the sub-cortical landmarks. Including further features to represent the sub-cortical structures should certainly help. Another interesting fact is that method II with the sulcal ribbons actually outperforms method I, which uses the outer cortical surface. This is not what we originally expected for a sparse feature representation as the major sulcal ribbons. Our speculation is that, compared to the cortex surface, the major sulcal ribbons are extended more into the brain, hence better placed. From the experiments, it seems that this better 3D placement may outweigh their disadvantages stemming from their sparseness. Also, note that the round shape of the cortical shape can cause errors in rotation estimation.

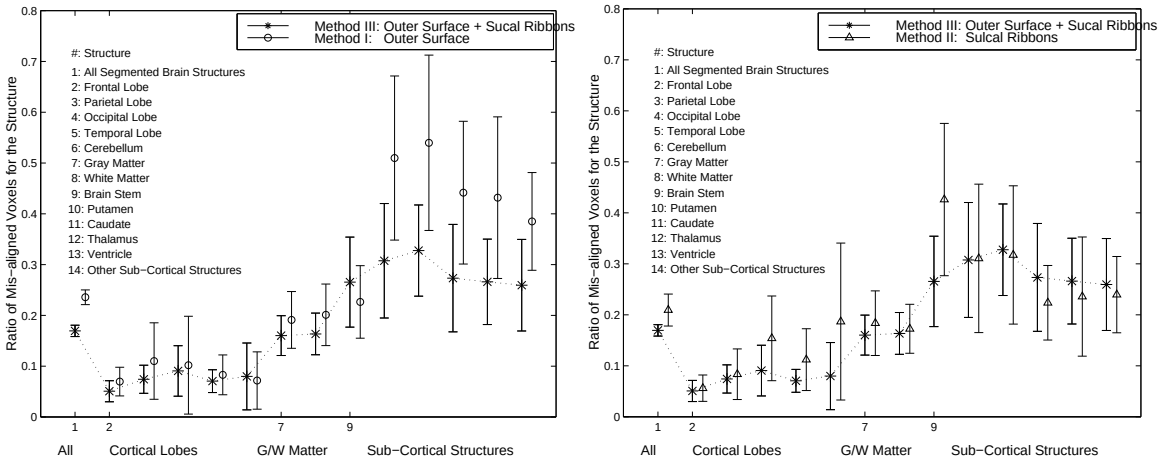

Figure 7.19: The error statistics based on the volume of test series 1 (local GRBF warp). Left, comparison of method III and I. Right, comparison of method III and II. Method I, with only the outer surface, yields much larger errors for all sub-cortical structures. Method II, with only the sulcal ribbons, tend to do worse near the rear region of the brain (occipital lobe, parietal lobe, cerebellum and brain stem), from where all the ribbons are relatively far away. Note: when measuring errors on the brain lobes, the distinction between the gray matter and the white matter is neglected to provide a more global and overall evaluation of the alignment.

Evaluation Based on Segmented Volume

As shown in Figure 7.19, the error statistics on the segmented volume data not only con£rm all our £ndings based on the landmarks but also provide more detailed information. While its errors on the cortical lobe regions are comparable to other methods, method I (with only the outer cortex surface) simply cannot provide enough anchoring information when it comes to the sub-cortical structures. On the other hand, method II (with only the sulcal ribbons) clearly suffers from the sparseness of its feature representation. For any structure that is relatively far away from the sulcal ribbons (e.g. cerebellum) , method II leads to large errors. All these problems can be avoided by combining them together in method III. The global nature of the TPS causes it to behave slightly different from the GRBF especially when the value of the locality parameter σ is small. Such small differences can accumulate a larger error in convoluted structures like the gray/white matter interface.

For global GRBF in test series 2, the results are very similar to the test series 1 except that TPS provides a better approximation for the global GRBF, leading to smaller errors.

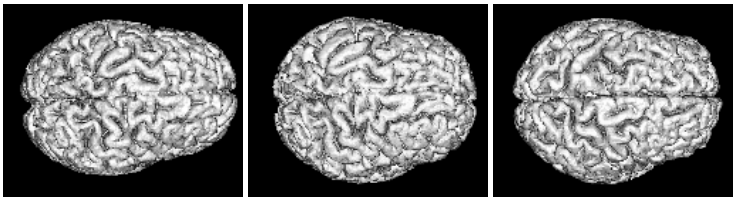

Figure 7.20: An experiment with two real brain MRIs. From left to right: 1) the reference brain; 2) warped reference brain (warped to match the target); 3) the target brain. Note the change in the brain's global shape.

Some results on real brain data

We conducted some preliminary experiments on a few pairs of real brain data as well. At the present stage, our feature representation £ts well for the global alignment of the brain. We include one example based on the currently available feature representation here in Figure 7.20.

We believe that to provide a more adequate representation of the complex brain structures, more detailed brain structural features are needed, especially within the sub-cortical regions. This is an important part of the future work.

7.5 Discussion and Conclusion

The experiments show that the combination of different features inherits each feature's merits while avoiding the problems when using each feature in isolation. As we discussed above, though the outer cortex provides a good model for the overall brain shape, its ellipsoidal shape is vulnerable to rotation errors. This weakness can be eliminated with the help of the sulcal ribbons. On the other hand, without the outer cortical surface, the sulcal ribbons alone are too sparse a feature representation. By jointly registering the features, the alignment is better constrained. Our uni£ed feature registration method provides a simple and effective way to accomplish this goal. Even though we only discussed the usage of two types of surfaces in this paper, the idea of fusing different features into a common point representation space is general and can be easily applied to other features such as the subcortical surfaces as well. It is also possible to add attribute factors to each type of features thereby attaining a better balance.

Future work on brain registration will include improving feature extraction to include more detailed anatomical feature, investigating further the behavior of different non-rigid deformation models and conducting more experiments on real brain data (possibly including brain fMRI data for the structure-function study).

Chapter 8

Conclusions

8.1 Conclusions and Contributions

Point matching becomes much more dif£cult when the transformation to be estimated is non-rigid rather than rigid. We have demonstrated in this thesis that the non-rigid point matching problem can be solved by our novel robust point matching algorithm (RPM). The RPM algorithm is capable of jointly estimating both the correspondence and the non-rigid transformation between two sets of points in the presence of both noise and outliers. The solution obtained by RPM is not the optimal one but in many cases has been shown to be a satisfactory, sub-optimal one.

Robustness is achieved through two techniques implemented within the RPM algorithm: soft assign and deterministic annealing. By relaxing the correspondence to be fuzzy, soft assign always allows more room for improvement in the optimization. Deterministic annealing provides £ne control over the fuzziness of the correspondence. Combined together, these two techniques has been demonstrated to effectively produce a coarse-to-£ne matching strategy, which can overcome many local minima encountered in the matching. The above is an empirical fact con£rmed through hundreds of experiments. Insofar as RPM has been empirically shown to be an ef£cient algorithm, it would certainly be very helpful to have theoretical results that are in sync with the experimental results.

We have also developed several other extensions for RPM, such as the joint clustering-matching algorithm. These extensions are aimed at effectively reducing the computational cost as well as augmenting the functionality, such as the ability to match multiple point-sets simultaneously while estimating an average super point-set.

We applied the joint clustering-matching algorithm to an important medical imaging problem — human brain anatomical feature registration. By using a common point feature representation, we were able to come up with

a uni£ed feature registration framework that can fuse and align different types of anatomical features together. In this way, the spatial inter-relationship between different features can be better utilized to determined a more optimal non-rigid deformation for the registration task. We also presented a carefully designed synthetic experiment, which, for the £rst time, systematically compared different types of features' ability to anchor the brain alignment separately and together.

8.2 Future Work

There is still plenty of room for further improvement. Better feature representation can de£nitely be explored. While the point feature representation has the advantage of being simple and ¤exible, more detailed shape information (e.g. local or global shape context as in [5, 6]) would certainly enhance the its representative ability and makes the matching problem easier.

Image intensity based registration and feature based registration have long been regarded as two totally different approaches. It doesn't have to be so. Insofar as a digital image is treated as nothing more than a set of intensity values de£ned over a regular grid of pixels, the image gray level intensity is just a third dimension. Viewed in this light, an intensity image is merely a 3D surface. 3D surfaces can be conceivably approximated by a set of 3D point nodes with spline interpolants £ling the gaps. Then the spline function coef£cients can be used as attributes for the points. In this way, intensity based registration becomes equivalent to the matching of the 3D surfaces, and hence to the matching of these 3D point nodes with attributes. The robust non-rigid point matching algorithm, which is originally designed for feature based registration, can be easily extended to intensity based registration as well. The possible merging of these two seemingly different registration £elds may invoke a lot of other innovations that can greatly bene£t the research £elds of medical imaging and computer vision.

Further work is also needed to £nd better transformation models for the non-rigid deformations. Most current deformation models used are very generic and not problem speci£c. Physic-based information is needed to build more realistic models. However, such information can hardly be captured during the image acquisition. If experimental facts about the objects' properties when in motion are known, these can certainly be used to build better transformation models.

We have brie¤y discuss the possible improvement needed for the brain anatomical feature registration at the end of Chapter 7. Better feature extraction and more experiments on real brain data is clearly needed to show that feature-based registration can be used in clinic.

Finally, in conclusion, the robust point matching algorithm and its extensions have been developed as rather general frameworks. They all has enormous potential applicability to a variety of registration problems in medical imaging and, to shape matching an recognition problems in computer vision. We hope that some this potential can actualized in the years to come.

Chapter 9

Appendix

9.1 Deterministic Annealing

Deterministic Annealing (DA) has been proposed as a useful *heuristic* to avoid local minima for a variety of optimization problems [44, 27, 62, 74, 32, 35, 43, 102, 67, 87]. It has proven to be especially successful in dealing with dif£cult combinatorial optimization problems, such as the Traveling Salesman Problem (TSP) [37, 27], which is very similar to our point matching problem.

Suppose that we start with some energy minimization problem at hand, which requires us to £nd a certain value or a con£guration of a discrete variable x that minimizes the energy function of $E(x)$.

$$\widehat{x} = \arg\min_{x} E(x) \tag{9.1}$$

Minimization over the discrete variable directly can be quite dif£cult. By adding an entropy-like term to the original discrete energy function, Deterministic Annealing (DA) can transform the discrete problem into a continuous problem (problem with continuous variables). The entropy term is often controlled by an external temperature parameter T. If we represent the original energy function (without the entropy term) as $E(x)$, and the entropy term as $TS(x)$, we will form a new free energy function:

$$F(x) = E(x) - TS(x) \tag{9.2}$$

Obviously, minimization of the original energy function $E(x)$ is equivalent to the minimization of this new free energy function $F(x)$ when $T = 0$.

The main bene£t of using this newly formed free energy function $F(x)$ for minimization is to overcome local minima. The minimization is £rst carried out with a large value of T. The added entropy term with a large value of T essentially convexify the original energy function. This guarantees that the local minimum obtained is in fact the global minimum at that temperature. The local minimum is then tracked as the T is gradually lowered. Generally, the energy function becomes non-convex during this process. However, it is hoped that the local minimum being tracked remains a good approximation to the global minimum as T approaches zero.

The above assumption doesn't always hold. It is still possible that the local minimum formed at a higher temperature can be far away from the global minimum. In such a case, DA will follow that local minimum and fail to approach the global minimum. In this sense, though DA offers an effective way to avoid many local minima in the optimization, it is in general suboptimal.

Softassign used within deterministic annealing has been shown to be guaranteed to £nd the global optimal solution to the simpler combinatorial problems such as the linear assignment problem [52], where the bene£t matrix is given beforehand. In our case, since the bene£t matrix depends on the transformation, which is also being optimized along with the correspondence, the usage of deterministic annealing no longer guarantees the answer to be globally optimal. Our approach is only guaranteed to £nd a local minimum for the mapping and the correspondence.

Using the additional entropy term to help the optimization in DA may seem a little bit arbitrary at £rst glance. It is not. Motivation for adding this extra term can be understood from different points of views.

From the probabilistic point of view, it is motivated by the maximum entropy principle [47, 87, 67], which basically says that maximum uncertainty should be encouraged whenever possible to eliminate restrictive biases. Since entropy is, naturally, the way to measure uncertainty/randomness, the motivation and the purpose of adding such an term is, again, natural.

DA can also be understood from a statistical physics point of view. People have long observed the fact that the perfect energy state (lowest energy) of a physical system can be achieved by maintaining the system at thermal equilibrium, and gradually lowering the temperature. This has been known as the annealing technique. Minimizing the free energy at each temperature is a way of preserving thermal equilibrium. A fundamental principle of statistical physics states: a thermal equilibrium is achieves when the free energy reaches its minimum. The DA method is producing precisely such an physical annealing process.

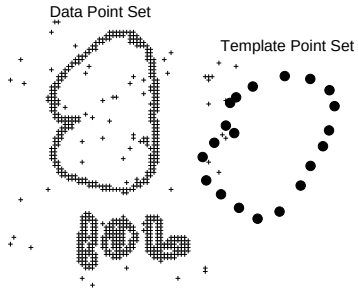

Figure 9.1: Asymmetric point matching: template point set (large solid discs) vs. data point set (small crosses). Note that the template needs to be rotated by a certain angle to be able to £t to part of the data.

9.2 Point Matching as Mixture Density Estimation — the Probabilistic Approach

While developing the robust point matching algorithm (RPM) in Chapter 3, we mainly followed an optimizational approach, treating the non-rigid point matching problem as a hand-crafted linear assignment—least squares problem. There is another way to attack the point matching problem. Here, we show that the point matching problem can also be regarded as a probability density estimation problem by using Gaussian mixture models. The relationship between the previous optimization approach and this probabilistic approach turned out to be particularly useful w.r.t. the outliers and the prior regularization.

9.2.1 Gaussian Mixture Model with Outliers

Notations

First, let's consider an asymmetric point matching case. We have one point set V or $\{v_a\}$ with sparsely distributed points, which serve as cluster centers (*template*). We also have another point set X or $\{x_i\}$ that has a relatively larger number of densely distributed points and serves as *data*. For the sake of simplicity, we will assume that the points are in 2D.

Again, we represent the transformation by a function f with parameters α. Applying it to a model point v_a will map it to its new location $u_a = f(v_a, \alpha)$. The whole transformed model set would then be U consisting of $\{u_a\}$.

This is a typical template matching problem, which is quite common in object recognition. The template represents the object that has to be identi£ed and is de£ned beforehand. The data set often contains not only the

desired object but also outlier points from other objects or background. The task is to transform the template to £nd the best £t to the part of the data that truly corresponds to it.

Gaussian Mixture Model

The asymmetric nature of the problem inspires us to introduce a particular Gaussian mixture density model where the data can be considered to be generated from a mixture of template cluster centers. Under the assumption that there are no outliers, the total probability density of each data point is the combination of all the Gaussian clusters,

$$p(x_i|U,\Sigma) = \sum_{a=1}^{K} \pi_a p(x_i|u_a,\Sigma_a) \tag{9.3}$$

where

$$p(x_i|u_a,\Sigma_a) = \frac{1}{2\pi|\Sigma_a|^{\frac{1}{2}}} e^{-\frac{(x_i-u_a)^T\Sigma_a^{-1}(x_i-u_a)}{2}} \tag{9.4}$$

and

$$u_a = f(v_a,\alpha). \tag{9.5}$$

There are two sets of parameters associated with the Gaussian mixture models. The £rst set of parameters is the set of covariance matrices $\{\Sigma_a\}$. The second is the set of "occupancy" parameters $\{\pi_a\}$. Each π_a informs us how large a percentage of the data belongs to a certain Gaussian cluster speci£ed by (u_a,Σ_a).

Assuming that the data points are independent, the probability density (likelihood) of the data point-set is,

$$\begin{aligned} p(X|V,\alpha,\Sigma,\pi) &= p(X|U,\Sigma,\pi) && (9.6)\\ &= \prod_{i=1}^{N} p(x_i|U,\Sigma,\pi) && (9.7)\\ &= \prod_{i=1}^{N}\sum_{a=1}^{K} \pi_a p(x_i|u_a,\Sigma_a) && (9.8)\\ &= \prod_{i=1}^{N}\sum_{a=1}^{K} \pi_a p(x_i|f(v_a,\alpha),\Sigma_a). && (9.9) \end{aligned}$$

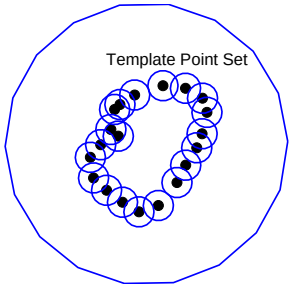

Figure 9.2: Gaussian mixture model for the template. Each template point represent a Gaussian cluster. A much larger outlier cluster is added to account for possible outlier data points.

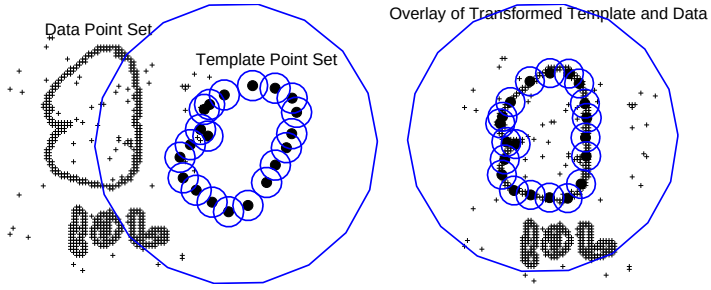

Figure 9.3: Point matching using the Gaussian mixture model. The original position of the model and the data points is shown on the left. We try to move the model around to achieve a better overlay, as shown on the right. At the optimal position, the true template clusters £t part of the data points, while the outlier cluster manage to account for the rest outliers.

Complete Gaussian Mixture Model with Outliers

To account for the spurious points (outliers) in the data, we introduce an extra outlier cluster in our template. This is done by introducing a $(K+1)^{th}$ Gaussian cluster with parameters v_{K+1}, π_{K+1} and Σ_{K+1}. The estimation of these parameters will be discussed later. For the time being, they are deemed known. The complete mixture model with the outlier cluster is,

$$p(X|V,\alpha,\Sigma,\pi) = \prod_{i=1}^{N} \sum_{a=1}^{K+1} \pi_a p(x_i|f(v_a,\alpha),\Sigma_a). \tag{9.10}$$

9.2.2 Point Matching as a MAP Problem

Different problems require different types of transformations. The choice of using a particular type of transformation re¤ects our *prior* knowledge on the problem. To ensure that the spatial mapping behaves according to our prior knowledge, we usually place certain constraints on the mappings. For example, certain *smoothness* regularization measures are often used to prevent the non-rigid mappings from behaving too arbitrarily. To this end, we introduce an operator L and represent the regularization on the mapping as $||Lf||^2$. The regularization can be regarded as a prior distribution on the transformation parameters α.

$$p(\alpha) = \frac{e^{-\lambda||Lf||^2}}{Z_{part}} \tag{9.11}$$

where Z_{part} is the partition/normalization function for the prior density $p(\alpha)$ and λ is a weight parameter. We can incorporate this prior to construct a posterior probability using Bayes' rule:

$$p(\alpha|X, V, \Sigma, \pi) = \frac{p(X|V, \alpha, \Sigma, \pi)p(\alpha)}{p(X)}. \tag{9.12}$$

The term $p(X)$ can be omitted since it doesn't depend on α. To £nd the best £t, we need to solve the maximum *a posteriori* (MAP) problem associated with (9.12). This is equivalent to minimizing the following log-posterior energy function:

$$E_1(\alpha, \Sigma, \pi) = -\log p(X|V, \alpha, \Sigma, \pi) - \log p(\alpha) \tag{9.13}$$

$$= -\sum_{i=1}^{N} \log \sum_{a=1}^{K+1} \pi_a p(x_i|f(v_a, \alpha), \Sigma_a) - \log p(\alpha) \tag{9.14}$$

The energy function contains three unknown parameters: the transformation parameter α and the mixture model parameters Σ, π. Again, from the template matching point of view, the transformation parameter tells us how to move our template to £t the data, and the mixture model parameters would tell us more details about the £tting. Overall, we are seeking for the best £t by solving this MAP problem.

9.2.3 EM Algorithm

The Expectation-Maximization (EM) algorithm provides an elegant way to solve this MAP problem [26]. It has been generally employed for a wide variety of parameter estimation problems [40, 59, 42, 66, 86, 13, 48].

Complete Data Posterior Energy Function

The dif£culty, or ambiguity, in the MAP problem originates from the lack of knowledge as to which mixture component a has generated a speci£c data point x_i and, therefore, which set of parameters (v_a, π_a, Σ_a) are in¤uenced by x_i.

EM algorithm resolves this ambiguity by introducing an additional intermediate binary variable Z (*latent data*) or $\{z_{ai}\}$, representing such classi£cation information between the data points and the mixture clusters. For example, $z_{ai} = 1$ implies that data point i (x_i) belongs to the a^{th} Gaussian cluster (v_a). The variable Z essentially plays the same role as the correspondence matrix introduced earlier in (3.1). For this reason, we use the same notation for both.

With the help of the expected value M, which is continuous, of the latent, binary variable Z, the minimization of the original log-posterior energy function (also termed the *incomplete data posterior* energy function) in (9.14) can be converted into the minimization of the *complete data posterior* energy function [40]:

$$\begin{aligned} E_{posterior}(M, \alpha, \Sigma, \pi) &= \sum_{i=1}^{N}\sum_{a=1}^{K+1} \frac{m_{ai}[x_i - f(v_a, \alpha)]^T \Sigma_a^{-1}[x_i - f(v_a, \alpha)]}{2} \\ &\quad - \sum_{i=1}^{N}\sum_{a=1}^{K+1} \frac{m_{ai}}{2} \log|\Sigma_a| - \sum_{i=1}^{N}\sum_{a=1}^{K+1} m_{ai} \log \pi_a \\ &\quad + \sum_{i=1}^{N}\sum_{a=1}^{K+1} m_{ai} \log m_{ai} + \lambda ||Lf||^2 \end{aligned} \tag{9.15}$$

where m_{ai} satis£es a one way constraint $\sum_{a=1}^{K+1} m_{ai} = 1$ for $i \in \{1, 2, \ldots, N\}$ with $m_{ai} \in [0, 1]$. Note that there is a great deal of similarity between this energy function and the one in Equation (3.2) developed through the optimizational approach in Chapter 3.

EM Algorithm

The elegance of the EM algorithm lies in the fact that for the newly formed complete data posterior energy function, there exists a simple dual step update which guarantees that the algorithm converges to a local minimum of the original

incomplete data posterior energy function.

The £rst step is called the *E-step*. Differentiating the above complete data posterior energy function w.r.t. each m_{ai} and setting the result to zero, we get the following update equation:

$$m_{ai} =< z_{ai} >= \frac{\pi_a p(x_i|f(v_a,\alpha),\Sigma_a)}{\sum_{b=1}^{K+1}\pi_b p(x_i|f(v_b,\alpha),\Sigma_b)}. \tag{9.16}$$

We use the notation $< z_{ai} >$ here because m_{ai} is actually the *conditional expected value* of z_{ai} at each step. This is the origin of the term E-step. The matrix M can be regarded as a fuzzy estimation of the binary classi£cation/correspondence Z.

The second step is called the *M-step*. Given a £xed M, this step minimizes the complete data energy function w.r.t. the rest of the parameters α, Σ and π. This can be done by just differentiating the energy function w.r.t. each of the parameters and setting the results to zero. The standard EM algorithm is quite simple. Starting with some initial value for α (e.g., identity transformation), the E-step is used to update the correspondence M and then the M-step is used to update the parameters α, π and Σ. The process is repeated until convergence[1].

9.2.4 Problems with the Mixture Model and the EM Algorithm

So far, the mixture model has proved to be useful. It enables us to directly include the outliers into the model. This probabilistic treatment of the outliers is intuitively quite appealing. It also allows us to regard the regularization term as a Bayesian prior, which de£nitely is more principled way to incorporate such constraint.

However, designed as a powerful tool for probability density estimation, the combination of the mixture model and the EM algorithm, in its current state, is still rather ill-suited for the task of non-rigid point matching.

First, we found it problematic[2] to estimate the covariance and the occupancy parameters in the mixture model. These parameters are not as closely related to the point matching problem, i.e., not as closely as M for the correspondence and α for the transformation. In addition, given M and α's large dimensions, non-rigid point matching is already an under-constrainted problem. Additional parameters can generate more local minima, making the problem

[1] Though there is no guarantee for the convergence to always end up at the global minimum.

[2] In fact, including the covariance and the occupancy parameters into the model may also bring some bene£t since they do make the model to be more ¤exible. However, they also make the algorithm to be less stable in the sense that it is more easily trapped by local minima. In short, there is a trade-off between model ¤exibility and algorithm stability. We chose the later and leave the model ¤exibility to be provided by the non-rigid transformation.

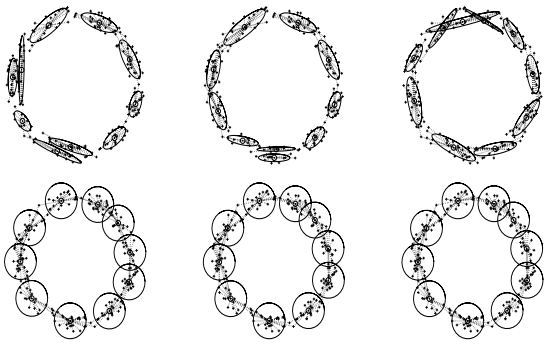

Figure 9.4: **Top Row**: The clusters estimated by the standard EM algorithm. **Bottom Row**: Modi£ed EM algorithm results.

even harder.

This problematic aspect of mixture modeling (as density estimation) is demonstrated through a clustering experiment . We used the standard EM algorithm to estimate the Gaussian mixture density for a contour shaped data point-set. Rather than matching the data points to a set of pre-de£ned template clusters, we cluster the data points to £nd our template, i.e., we let the clusters (no outlier cluster) be estimated as free parameters and there is no transformation involved. Using the same data and the same initialization, with the only difference being the addition of symmetry-breaking random noise to M, the £nal con£gurations found by the EM algorithm are shown in the £rst row in Figure 9.4. With the extra ¤exibility provided by the covariance and the occupancy parameters, it is clear that the answers can be quite different. This is a problem speci£cally for non-rigid point matching because the template can then be transformed in different ways and still £t the data quite well.

There are some other concerns about using the Gaussian mixture model as well. The differentiation between the sparse model and the dense data—implicit in the mixture model—is not always appropriate for the purposes of point matching. When there are roughly equal numbers of points in the data and in the template with outliers in both, this assumption is no longer valid. Furthermore, this differentiation also causes the correspondence to be many-to-one instead of one-to-one because it only enforces a one-way constraint, $\sum_{a=1}^{K+1} m_{ai} = 1$. We need to modify the mixture formulation as well as the EM algorithm to overcome these problems.

9.2.5 Modi£ed EM Algorithm

A Simpli£ed Mixture Model

A careful examination of the complete data posterior energy function in (9.15) and the fuzzy assignment-least squares energy function in (3.2) reveals a great deal of similarity [103] between the two. This similarity also helps us simplify the mixture model and to overcome the problem of the extra parameters in its formulation.

By closely comparing (3.2) and (9.15), we £nd that the temperature parameter T essentially plays the same role as the covariance parameter Σ [103, 87].

The intuition behind annealing can be explained with the following simple observations. When the values of the covariance matrix ($\{\Sigma_a\}$) entries are large, the Mahalanobis distance in (9.15) between any pair of points is close to zero. The resulting estimation of correspondence is then nearly uniform (values close to $\frac{1}{K+1}$). This means that points are allowed to match to many cluster centers (including those that are far away) with a fuzzy degree of con£dence. On the other hand, if the values of the covariance matrix entries are nearly zero, the correspondence will be close to binary[3] and the points will only be allowed to match to the nearest cluster centers. These observations suggest that the covariance matrix set Σ is basically acting like a *search range* parameter. When the covariance magnitudes are large, the cluster centers search globally for possible member data. When the covariance magnitudes are small, the cluster centers look for candidates more locally. Annealing the energy function by gradually reducing the magnitude of the covariances clearly leads to a global-to-local match strategy. This is accomplished by using the isotropic form for the covariance matrix set Σ and by representing the magnitude using a temperature parameter T. The original Mahalanobis distance measure is now simpli£ed to the Euclidean distance in the following way:

$$\frac{[x_i - f(v_a, \alpha)]^T \Sigma_a^{-1} [x_i - f(v_a, \alpha)]}{2} \mapsto \frac{||x_i - f(v_a, \alpha)||^2}{2T}.$$

The outliers need special treatment. Since we have little information about the outliers, the simplest approach is to associate a large covariance Σ_{K+1} to the outlier cluster in order to account for any possible outlier within the point-sets. Instead of using T, we keep the temperature for the outlier class at a large constant value T_0. The outlier class introduces competition. Unless a data point shows strong evidence that it belongs to one of the non-outlier

[3] In fact, what will happen then is that one correspondence value will be much larger than the rest. After normalization (to satisfy the summation constraint), the correspondence values will be close to binary. The correspondence value between the nearest point pair will be close to 1 and all of the rest values will be close to 0.

clusters, it is rejected as an outlier. To summarize, the covariance matrices $\{\Sigma_a, a = 1, 2, \ldots\ K+1\}$ are set as the following:

$$\begin{aligned} \Sigma_a &\rightarrow T, \quad for\ a = 1, 2, \ldots, K, \\ \Sigma_{K+1} &\rightarrow T_0, \quad for\ a = K+1. \end{aligned}$$

The outlier center v_{K+1} is set to the center of mass of all data points. It is also being transformed with the rest of the template clusters $\{v_a, a = 1, 2, \ldots\ K\}$.

We make another simpli£cation to the mixture model. Since we do not have prior knowledge of the occupancy parameter π, we uniformly assign each π_a to be $\frac{1}{K+1}$, i.e., each model cluster (including the outlier cluster) is equally important to us and should be able to explain equal amounts of data.

After substituting T for Σ and eliminating π (π_a are now constants and can be removed), we repeated the same experiment as described above in 9.2.4. The annealing process is performed by gradually lowering T with a pre-speci£ed schedule. As shown in the second row in Figure 9.4, the clusters found through the modi£ed EM algorithm are all evenly distributed. They are also much more consistent with little variations due to initial conditions or symmetry breaking perturbations.

There is still the issue of the asymmetric nature of the mixture model. It is obvious that we will need two outlier clusters—one for each point-set. The requirement of one-to-one correspondence rather than many-to-one is enforced by including the two-way constraints inherent in the linear assignment problem. The Sinkhorn double normalization [75] process is added to the EM algorithm to guarantee that these constraints are alway satis£ed.

Let's summarize all the modi£cations that we have just discussed. We introduced two outlier clusters, one for each of the point-sets. Their centers are represented as v_{K+1} and x_{N+1}, which are both set to be at the center of mass of the original point-set. The outlier clusters' covariances (temperature) are set to be a large scalar T_0 and are kept at this value, while the rest of the template clusters' covariances are set to be T and are being slowly annealed (decreased) during the matching process. Because of this difference between the outlier cluster and the normal template clusters, some of the terms in the previous complete data posterior energy function will have to be split.

$$\begin{aligned} \sum_{i=1}^{N}\sum_{a=1}^{K+1} m_{ai}\frac{[x_i - f(v_a,\alpha)]^T\Sigma_a^{-1}[x_i - f(v_a,\alpha)]}{2} \quad &\rightarrow \quad \sum_{i=1}^{N}\sum_{a=1}^{K} m_{ai}\frac{||x_i - v_a||^2}{2T} \\ &+ \sum_{i=1}^{N} m_{K+1,i}\frac{||x_i - v_{K+1}||^2}{2T_0} \end{aligned}$$

$$
\begin{aligned}
& & & + \sum_{a=1}^{K} m_{a,N+1} \frac{||x_{N+1} - v_a||^2}{2T_0} \\
\sum_{i=1}^{N} \sum_{a=1}^{K+1} m_{ai} \frac{\log |\Sigma_a|}{2} & \rightarrow & \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \frac{\log T}{2} & \\
& & + \sum_{i=1}^{N} m_{K+1,i} \frac{\log T_0}{2} + \sum_{a=1}^{K} m_{a,N+1} \frac{\log T_0}{2} &
\end{aligned}
$$

Also, after we simpli£ed the occupancy parameters π_a to be all equal. It can be effectively dropped out. The £nal modi£cation is that now the membership (correspondence) variable will have to satisfy both the row and the column constraints.

Mixture Point Matching Energy Function

After making these modi£cations to (9.15), we get the energy function used for the probabilistic approach:

$$
\begin{aligned}
E_{MPM}(M, \alpha) \quad = \quad & \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \frac{||x_i - f(v_a, \alpha)||^2}{2T} \\
& + \sum_{i=1}^{N} m_{K+1,i} \frac{||x_i - v_{K+1}||^2}{2T_0} + \sum_{a=1}^{K} m_{a,N+1} \frac{||x_{N+1} - v_a||^2}{2T_0} \\
& + \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \frac{\log T}{2} \\
& + \sum_{i=1}^{N} m_{K+1,i} \frac{\log T_0}{2} + \sum_{a=1}^{K} m_{a,N+1} \frac{\log T_0}{2} \\
& + \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \log m_{ai} + \lambda ||Lf||^2
\end{aligned}
\tag{9.17}
$$

where m_{ai} still satis£es two constraints: $\sum_{i=1}^{N+1} m_{ai} = 1$ for $a \in \{1, 2, \ldots, K\}$ and $\sum_{a=1}^{K+1} m_{ai} = 1$ for $i \in \{1, 2, \ldots, N\}$ with $m_{ai} > 0$ (except that $m_{K+1,N+1} \equiv 0$). Since it is derived from the mixture model, we refer to it as the *mixture point matching (MPM) energy function*. There are a couple interesting things about this energy function.

First, the reader may notice that the outliers are now directly included in the model. This is done through the introduction of the outlier class with associated parameters x_{N+1}, v_{K+1} and T_0 (note that they are all constants and deemed known). Using the row and column constraints, it is not hard to see that this outlier treatment is equivalent to the original robustness control term $-\zeta \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai}$ [as in (3.1) and in (3.2)]. A subtle but important difference is that the weight parameter is no longer a constant but is instead modulated by T. The intuition behind the this outlier

model is clear with the help of the mixture model.

Next, the reader may notice that if we try make the MPM energy function (9.17) more similar to the RPM energy function by multiplying both sides of the by T, we get an *annealed* version of the smoothing prior ($\lambda T||Lf||^2$). This is, in fact, where the "self-relaxing prior" idea in RPM is originated. When the temperature is high, the transformation is heavily regularized by the dominant prior term. When the temperature is small, the transformation is in¤uenced more by the data, as the in¤uence of the prior wanes. It is interesting that this intuitively appealing technique can be formally derived for the point matching problem through the combination of a mixture model and deterministic annealing.

9.2.6 The Mixture Point Matching Algorithm

The resulting mixture point matching algorithm (MPM) is almost the same as the previous RPM algorithm. Compared to the original EM algorithm, it is essentially a deterministic annealing version [103, 87] of it. We brie¤y describe the algorithm.

E Step: Update the Correspondence

For the points $a = 1, 2, \ldots, K$ and $i = 1, 2, \ldots, N$,

$$q_{ai} = \frac{1}{\sqrt{T}} e^{-\frac{(x_i - f(v_a,\alpha))^T (x_i - f(v_a,\alpha))}{2T}} \tag{9.18}$$

and for the outlier entries $a = K + 1$ and $i = 1, 2, \ldots, N$,

$$q_{K+1,i} = \frac{1}{\sqrt{T_0}} e^{-\frac{(x_i - v_{K+1})^T (x_i - v_{K+1})}{2T_0}} \tag{9.19}$$

and for the outlier entries $a = 1, 2, \ldots K$ and $i = N + 1$, q

$$q_{a,N+1} = \frac{1}{\sqrt{T_0}} e^{-\frac{(x_{N+1} - f(v_a,\alpha))^T (x_{N+1} - f(v_a,\alpha))}{2T_0}} \tag{9.20}$$

where v_{K+1} and x_{N+1} are outlier cluster centers. Note that the only difference from the RPM algorithm is the Gaussian normalization factor $\frac{1}{\sqrt{T}}$ (or $\frac{1}{\sqrt{T_0}}$) in front of the exponential form.

Then run the Sinkhorn algorithm (iterated row and column normalization) until convergence is reached,

$$\text{Column normalization: } m_{ai} = \frac{q_{ai}}{\sum_{b=1}^{K+1} q_{bi}}, \tag{9.21}$$

$$\text{Row normalization: } m_{ai} = \frac{q_{ai}}{\sum_{j=1}^{N+1} q_{aj}}. \tag{9.22}$$

For the purposes of symmetry breaking, a very small amount of Gaussian noise (mean zero and small standard deviation σ) can be added to m_{ai} after the double normalization.

M Step: Update the Transformation

After dropping the terms independent of α, we need to solve the following least-squares problem,

$$\min_{\alpha} E(\alpha) = \min_{\alpha} \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} ||x_i - f(v_a, \alpha)||^2 + \lambda T ||Lf||^2. \tag{9.23}$$

Including the outlier points in the least squares formulation is very cumbersome. Again, we simplify it to be the following:

$$\min_{\alpha} E(\alpha) = \min_{\alpha} \sum_{a=1}^{K} ||y_a - f(v_a, \alpha)||^2 + \lambda T ||Lf||^2 \tag{9.24}$$

where,

$$y_a = \frac{\sum_{i=1}^{N} m_{ai} x_i}{\sum_{i=1}^{N} m_{ai}} \tag{9.25}$$

The variable y_a can be regarded as our newly estimated positions of the point-set (within the set $\{x_i\}$) that corresponds to $\{v_a\}$. Extra bookkeeping is needed to check for outliers (if $\sum_{i=1}^{N} m_{ai}$ is too small) and eliminate them. The solution for this least-squares problem depends on the speci£c form of the non-rigid transformation.

Annealing

An annealing scheme (for the temperature parameter T) controls the EM update process. The setting is the same as in the RPM. We just brie¤y re-summarize here.

Starting at $T_{init} = T_0$, the temperature parameter T is gradually reduced according to a linear annealing schedule, $T^{new} = T^{old} \cdot r$ (r is called the annealing rate). The dual updates are repeated until convergence at each temperature. The parameter T is then lowered and the process is repeated until some £nal temperature T_{final} is reached.

The parameter T_0 is set to the largest squared distance of all point pairs. We set r to be 0.93 (normally between $[0.9, 0.99\ ]$) so that the annealing process is slow enough for the algorithm to be robust, and yet not too slow. For the outlier cluster, the temperature is always kept at T_0. Since the data is often quite noisy, matching them exactly to get binary one-to-one correspondences is not always desirable. To prevent over£tting, the parameter T_{final} should be set according to the amount of noise within the data set. In this work, we make a simpli£ed treatment to set T_{final} to be equal to the average of the squared distance between the nearest neighbors within the set of points which are being deformed. The interpretation is that at T_{final}, the Gaussian clusters for all the points will then barely overlap with one another.

Mixture Point Matching Algorithm Pseudo-code

The Mixture Point Matching Algorithm (MPM) Pseudo-code:

Initialize parameters $T = T_0$ and λ.

Initialize parameters α (or M).

Begin A: Deterministic Annealing.

> **Begin B: EM Update.**
>
> E Step: Update correspondence parameter M based on current α.
>
> M Step: Update transformation parameter α based on current M.
>
> **End B**

Decrease $T \leftarrow T \cdot r$ until T_{final} is reached.

End A

9.3 Radial Basis Function Splines

In the point matching process, once the correspondence is known, we then need to £nd the best spatial mapping/transformation to bring the corresponding points (y_a and v_a) together. The generation of a smooth spatial mapping between two sets

of points with known correspondence is a general problem in spline theory.

$$f = \arg\min_f \left(\sum_{a=1}^{K} ||y_a - f(v_a)||^2 + \lambda ||Lf||^2 \right) \tag{9.26}$$

As we mentioned before, once non-rigidity is allowed, there are an in£nite number of ways to map one point-set onto another. The smoothness constraint is necessary because it discourages mappings which are too arbitrary. In other words, we can control the behavior of the mapping by choosing a speci£c smoothness measure, which basically re¤ects our prior knowledge. Different choice of smoothness measures lead to different splines.

We implemented two types of radial basis function splines [94] to parameterize the non-rigid deformation. Given a set of control points $\{w_b, b = 1, 2, \ldots n\}$, a radial basis function basically de£nes a spatial mapping which maps any location v in space to a new location $f(v)$, represented by,

$$f(v) = \sum_{b=1}^{n} c_b \phi(||v - w_b||) \tag{9.27}$$

where $||\cdot||$ denotes the usual Euclidean norm in 3D and $\{c_a\}$ is a set of mapping coef£cients. The kernel function ϕ can assume different forms.

If we choose the form of the kernel to be $\phi(r) = \exp(-r^2/\sigma^2)$, it becomes a Gaussian Radial Basis Function (GRBF). The parameter σ controls the width/locality of each kernel function. A small value of σ generates more localized and hence less smooth warping. A different choice of $\phi(r) = r^2 \log r$ leads to another type of radial basis function called the Thin Plate Spline (TPS). Compared to the GRBF, TPS has a more global nature—a small perturbation of one of the control points always affects the coef£cients corresponding to all the other points as well. Needless to say, the speci£c form of the kernel function depends on the smoothness measure.

We include a brief derivation for the closed form solutions for both GRBF and TPS for the spline £tting problem encountered in our point matching algorithm. Details about these splines can be found in [94].

9.3.1 The Thin-Plate Spline

TPS Smoothness Measure

One of the simplest smoothness measures is the space integral of the square of the second order derivatives of the mapping function [10, 94]. This leads us to the thin-plate spline (TPS). The TPS £ts a mapping function $f(v_a)$

between corresponding point-sets $\{y_a\}$ and $\{v_a\}$ by minimizing the following energy function:

$$f_{TPS} = \arg\min_f E_{TPS}(f) \tag{9.28}$$

$$= \arg\min_f \left(\sum_{a=1}^{K} ||y_a - f(v_a)||^2 + \lambda \int\int [(\frac{\partial^2 f}{\partial x^2})^2 + 2(\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2] dx dy \right) \tag{9.29}$$

TPS Spline

Suppose the points are in 2D ($D = 2$). We use *homogeneous* coordinates for the point-set where a point y_a is represented as a vector $(1, y_{ax}, y_{ay})$. With a £xed weight parameter λ, there exists a unique minimizer f parameterized by α which comprises two matrices d and c, ($\alpha = \{d, c\}$).

$$f_{TPS}(x, \alpha) = f_{TPS}(x, d, c) = x \cdot d + \sum_{b=1}^{K} \phi(||x - v_b||) \cdot c_b \tag{9.30}$$

where d is a $(D+1) \times (D+1)$ matrix representing the af£ne transformation and c is a $K \times (D+1)$ warping coef£cient matrix representing the non-af£ne deformation. The kernel function $\phi(||x - v_b||)$ is a $1 \times K$ vector for each point x, where each entry $\phi_b(x) = ||x - v_b||^2 \log ||x - v_b||$. Note that for TPS, the control points $\{w_b\}$are chosen to be the same as the set of points to be warped $\{v_a\}$, so we already used $\{v_b\}$ in the place of the control points.

If we substitute the solution for f (9.30) into (9.29), the TPS energy function becomes,

$$E_{TPS}(d, c) = ||Y - Vd - \Phi c||^2 + \lambda \,\text{trace}\, (c^T \Phi c) \tag{9.31}$$

where Y and V are just concatenated versions of the point coordinates y_a and v_a, and Φ is a $(K \times K)$ matrix formed from the $\phi(||v_a - v_b||)$. Each row of each newly formed matrix comes from one of the original vectors. The matrix Φ represents the TPS *kernel*. Loosely speaking, the TPS kernel contains the information about the point-set's internal structural relationships. When it is combined with the warping coef£cients c, a non-rigid warping is generated.

A nice property of the TPS is that it can always be decomposed into a global af£ne and a local non-af£ne component. Consequently, the TPS smoothness term in (9.29) is solely dependent on the non-af£ne components. This is a desirable property, especially when compared to other splines, since the global pose parameters included in the af£ne transformation are not penalized.

Closed Form Solution for TPS

The separation of the af£ne and non-af£ne warping space is done through a QR decomposition [94].

$$V = [Q_1|Q_2] \begin{pmatrix} R \\ 0 \end{pmatrix} \tag{9.32}$$

where Q_1 and Q_2 are $K \times (D+1)$ and $N \times (K-D-1)$ orthonormal matrices, respectively. The matrix R is upper triangular.

With the QR decomposition in place, (9.31) becomes,

$$E_{TPS}(\gamma, d) = ||Q_2^T Y - Q_2^T \Phi Q_2 \gamma||^2 + ||Q_1^T Y - Rd - Q_1^T \Phi Q_2 \gamma||^2 + \lambda \gamma^T Q_2^T \Phi Q_2 \gamma \tag{9.33}$$

where $c = Q_2\gamma$ and γ is a $(K-D-1) \times (D+1)$ matrix. Setting $c = Q_2\gamma$ (which in turn implies that $V^T c = 0$) enables us to cleanly separate the £rst term in (9.31) into a non-af£ne term and an af£ne term [£rst and second terms in (9.33) respectively].

The least-squares energy function in (9.33) can be £rst minimized w.r.t γ and then w.r.t. d. The £nal solution for c and d are,

$$\widehat{c} = Q_2(Q_2^T \Phi Q_2 + \lambda I_{(K-D-1)})^{-1} Q_2^T Y, \tag{9.34}$$

$$\widehat{d} = R^{-1}(Q_1^T Y - \Phi \widehat{c}). \tag{9.35}$$

We call the minimum value of the TPS energy function obtained at the optimum $(\widehat{c}, \widehat{d})$ the "bending energy":

$$E_{bending} = \lambda \,\text{trace}\, [Q_2(Q_2^T \Phi Q_2 + \lambda I_{(K-D-1)})^{-1} Q_2^T Y Y^T]. \tag{9.36}$$

9.3.2 The Gaussian Radial Basis Function Spline

The Gaussian radial basis function spline is quite similar to the TPS. Since there is no decomposition between the rigid and the non-rigid components within the transformation, we denote the transformation parameters just as c. A GRBF

spline can be written as,

$$f_{GRBF}(x,\alpha) = f_{GRBF}(x,c) = \sum_{b=1}^{n} \phi(||x - w_b||) \cdot c_b \tag{9.37}$$

where a set of prede£ned points $\{w_b\}$ are serving as control points[4].

The GRBF energy function is,

$$E_{GRBF}(c) = ||Y - \Phi c||^2 + \lambda \,\text{trace}\,(c^T \Phi c) \tag{9.38}$$

Since there is no decomposition, the solution for GRBF is straight forward.

$$\widehat{c} = (\Phi^T \Phi + \lambda \Phi)^{-1} \Phi^T Y \tag{9.39}$$

[4]This is different from TPS. For TPS, the point set that is being warped also serves as the control point set. Here for GRBF, normally a seperate set of control points are needed.

Bibliography

[1] Y. Amit and A. Kong. Graphical templates for model recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(4):225–236, 1996.

[2] H. Baird. *Model-Based Image Matching Using Location*. MIT Press, Cambridge, MA, 1984.

[3] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing*, 46:1–21, 1989.

[4] D. H. Ballard. Generalized Hough transform to detect arbitrary patterns. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(2):111–122, 1981.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape context: a new descriptor for shape matching and object recognition. In *Proceedings to Neural Information Processing Systems–NIPS'2000*, 2000.

[6] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *International Conference on Computer Vision–ICCV'2001*, 2001. submitted.

[7] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 14(2):239–256, Feb. 1992.

[8] R. Bhatia. *Matrix Analysis*, volume 169 of *Graduate Texts in Mathematics*. Springer, New York, NY, 1996. page 38.

[9] C. M. Bishop, M. Svensen, and C. K. I. Williams. Gtm: the generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[10] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(6):567–585, June 1989.

[11] L. G. Brown. A survey of image registration techniques. *Computing Surveys*, 24(4):325–376, December 1992.

[12] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(10):1042–1052, Nov. 1993.

[13] W. Bryne. Alternating minimization and boltzman machine learning. *IEEE Trans. Neural Networks*, 3:612–620, 1992.

[14] G. Christensen. Consistent linear-elastic transformations for image matching. In *Proceedings of Information Processing in Medical Imaging—IPMI 99*, pages 224–237. Springer-Verlag, 1999.

[15] G. Christensen, S. Joshi, and M. Miller. Volumetric transformation of brain anatomy. *IEEE Trans. Med. Imag.*, 16(6):864–877, 1997.

[16] H. Chui, J. Rambo, J. Duncan, R. Schultz, and A. Rangarajan. Registration of cortical anatomical structures via robust 3D point matching. In *Proceedings of Information Processing in Medical Imaging–IPMI 99*, pages 168–181. Springer-Verlag, 1999.

[17] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition–CVPR 2000*, volume 2, pages 44–51. IEEE Press, 2000.

[18] D. Collins, G. Goualher, and A. Evans. Non-linear cerebral registration with sulcal constraints. In W. Wells, A. Colchester, and S. Delp, editors, *Proceedings of Medical Image Computing and Computer-Assisted Intervention–MICCAI 98*, pages 974–984. Springer, 1998.

[19] D. Collins, C. Holmes, T. Peters, and A. Evans. Automatic 3D model-based neuro-anatomical segmentation. *Human Brain Mapping*, 3(3):190–208, 1995.

[20] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models: Their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[21] B. Kim C.R. Meyer, J. L. Boes and P. H. Bland. Demonstration of accuracy and clinical verstility of mutual information for automatic multimodality image fusion using af£ne and thin plate spline warped geometric deformations. *Medical Image Analysis*, 1(3):195–206, 1997.

[22] A. D. J. Cross and E. R. Hancock. Graph matching with a dual-step EM algorithm. *IEEE Trans. Patt. Anal. Mach. Intell.*, 20(11):1236–1253, 1998.

[23] C. Davatzikos. Spatial transformation and registration of brain images using elastically deformable models. *Computer Vision and Image Understanding: Special Issue on Medical Imaging*, 6(2):207–222, 1997.

[24] C. Davatzikos and J.L. Prince. Brain image registration based on curve mapping. *Proc. of the IEEE Workshop on Biom. Image Anal.*, pages 245–254, 1994.

[25] D. DeCarlo and D. Metaxas. The integration of optical ¤ow and deformable models with applications to human face shape and motion estimation. In *CVPR 96*, pages 231–238, 1996.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. Ser. B*, 39:1–38, 1977.

[27] R. Durbin, R. Szeliski, and A. L. Yuille. An analysis of the elastic net approach to the traveling salesman problem. *Neural Computation*, 1:348–358, 1989.

[28] J. Feldmar and N. Ayache. Rigid, af£ne and locally af£ne registration of free-form surfaces. *Intl. J. Computer Vision*, 18(2):99–119, May 1996.

[29] A. H. Gee, S. Aiyer, and R. W. Prager. An analytical framework for optimizing neural networks. *Neural Networks*, 6:79–97, 1993.

[30] A. H. Gee and R. W. Prager. Limitations of neural networks for solving traveling salesman problems. *IEEE Trans. on Neural Networks*, 6(1):280–282, Jan. 1995.

[31] J. Gee. *Probabilistic Matching of deformed images*. PhD thesis, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1995.

[32] D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRFs: Surface reconstruction. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(5):401–412, May 1991.

[33] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos. Dynamic-programming for detecting, tracking, and matching deformable contours. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(3):294–302, March 1995.

[34] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(4):377–388, 1996.

[35] S. Gold and A. Rangarajan. Softassign versus softmax: Benchmarks in combinatorial optimization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 626–632. MIT Press, Cambridge, MA, 1996.

[36] S. Gold, A. Rangarajan, C. P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2-D and 3-D point matching: pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031, 1998.

[37] B. I. Golden and W. R. Stewart. Empirical analysis of heuristics. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem*, chapter 7, pages 207–249. John Wiley and Sons, Chichester, 1985.

[38] E. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge, MA, 1990.

[39] E. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Patt. Anal. Mach. Intell.*, 9:468–482, 1987.

[40] R. Hathaway. Another interpretation of the EM algorithm for mixture distributions. *Statistics and Probability Letters*, 4:53–56, 1986.

[41] L. S. Hibbard and R. A. Hawkins. Objective image alignment for three–dimensional reconstruction of digital autoradiograms. *J. Neurosci. Methods*, 26:55–75, 1988.

[42] G. Hinton, C. Williams, and M. Revow. Adaptive elastic models for hand-printed character recognition. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 512–519. Morgan Kaufmann, San Mateo, CA, 1992.

[43] T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(1):1–14, Jan. 1997.

[44] J. J. Hop£eld and D. Tank. ‘Neural’ computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[45] R. Hummel and H. Wolfson. Af£ne invariant matching. In *Proceedings of the DARPA Image Understanding Workshop*, pages 351–364, Cambridge, MA, 1988.

[46] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(9):850–863, Sep. 1993.

[47] E. T. Jaynes. On the rationale of maximum-entropy methods. *Proc. IEEE*, 70:939–952, 1982.

[48] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, March 1994.

[49] S. Joshi and M. Miller. Landmark matching via large deformation diffeomorphisms, 1998. (preprint).

[50] N. Khaneja, M. I. Miller, and U. Grenander. Dynamic programming generation of curves on brain surfaces. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(1):1260–1265, 1998.

[51] T. Kohonen. *Self-Organization and associative memory*. New York: Springer-Verlag, 3rd edition, 1989.

[52] J. J. Kosowsky and A. L. Yuille. The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural Networks*, 7(3):477–490, 1994.

[53] Y. Lamdan, J. Schwartz, and H. Wolfson. Object recognition by af£ne invariant matching. *IEEE Conf. Comp. Vision, Patt. Recog.*, pages 335–344, 1988.

[54] G. Lohmann and D. von Cramon. Sulcal basin and sulcal strings as new concepts for describing the human cortical topography. In *Workshop on Biomedical Image Analysis*, pages 41–54. IEEE Press, June 1998.

[55] D. Luenberger. *Linear and Nonlinear Programming*. Addison–Wesley, Reading, MA, 1984.

[56] D. MacDonald, N. Kabani, D. Avis, and A. Evans. Automated 3d extraction of inner and outer surfaces of cerebral cortex from mri. *NeuroImage*, 12:340–356, 2000.

[57] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2:1–36, 1998.

[58] J. Mazziotta, A. Toga, A. Evans, P. Fox, and J. Lancaster. A probabilistic atlas of the human brain: theory and rationale for its development. *NeuroImage*, 2(2):89–101, 1995.

[59] G. J. McLachlan and K. E. Basford. *Mixture models: inference and applications to clustering*. Marcel Dekker, New York, 1988.

[60] D. Metaxas, E. Koh, and N. I. Badler. Multi-level shape representation using global deformations and locally adaptive £nite elements. *Intl. J. Computer Vision*, 25(1):49–61, 1997.

[61] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.

[62] C. Peterson and B. Soderberg. A new method for mapping optimization problems onto neural networks. *Intl. Journal of Neural Systems*, 1(1):3–22, 1989.

[63] A. Rangarajan, H. Chui, and F. Bookstein. The softassign Procrustes matching algorithm. In *Information Processing in Medical Imaging (IPMI '97)*, pages 29–42. Springer, 1997.

[64] A. Rangarajan, H. Chui, and J. Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, 1999. (in press).

[65] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. Goldman-Rakic, and J. Duncan. A robust point matching algorithm for autoradiograph alignment. *Medical Image Analysis*, 4(1):379–398, 1997.

[66] M. Revow, C. K. I. Williams, and G. Hinton. Using generative models for handwritten digit recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(6):592–606, June 1996.

[67] K. Rose, E. Gurewitz, and G. Fox. Constrained clustering as an optimization method. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(8):785–794, 1993.

[68] S. Sandor and R. Leahy. Surface based labeling of cortical anatomy using a deformable atlas. *IEEE Trans. Med. Imag.*, 16(1):41–54, 1997.

[69] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(6):545–561, Jun. 1995.

[70] G. Scott and C. Longuet-Higgins. An algorithm for associating the features of two images. *Proc. Royal Society of London*, B244:21–26, 1991.

[71] T. B. Sebastian, J. J. Crisco, P. N. Klein, and B. B. Kimia. Construction of 2d curve altases. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis–MMBIA 2000*, pages 70–77, June 2000.

[72] L. Shapiro and J. Brady. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, 10:283–288, 1992.

[73] L. G. Shapiro and R. M. Haralick. Structural descriptions and inexact matching. *IEEE Trans. Patt. Anal. Mach. Intell.*, 3(9):504–519, Sept. 1981.

[74] P. D. Simic. Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimisations. *Network*, 1:89–103, 1990.

[75] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879, 1964.

[76] S. N. Steketee and N. I. Badler. Parametric keyframe interpolation incorporating kinetic adjustment and phrasing control. In *SIGGRAPH 85*, pages 255–262, 1985.

[77] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40, 1987.

[78] C. Studholme. *Measures of 3D medical image alignment*. PhD thesis, University of London, London, U.K., 1997.

[79] R. Szeliski and S. Lavallee. Matching 3D anatomical surfaces with non-rigid deformations using octree splines. *Intl. J. Computer Vision*, 18:171–186, 1996.

[80] H. Tagare. Shape-based nonrigid correspondence with application to heart motion analysis. *IEEE Transactions on Medical Imaging*, 18(7):570–579, July 1999.

[81] J. Talairach and G. Szikla. *Co-planar stereotaxic atlas of the human brain*. Thieme Medical Pub., New York, 1988.

[82] P. Thompson, J. Giedd, R. Woods, D. MacDonald, A. Evans, and A. Toga. Patterns in the developing human brain detected using continuum-mechanical tensor mapping. *Nature*, 404(6774), March 2000.

[83] P. Thompson, D. MacDonald, M.S. Mega, C.J. Holmes, C.J. Evans, and A.W. Toga. Detection and mapping of abnormal brain structure with a probabilistic atlas of cortical surfaces. *Journal of Computer Assisted Tomography*, 21(4):567–581, 1997.

[84] P. Thompson and A. W. Toga. A surface-based technique for warping three-dimensional images of the brain. *IEEE Trans. Med. Imag.*, 5(4):402–417, August 1996.

[85] A. Toga and J. Mazziotta. *Brain Mapping: The Methods*. Academic Press, 1996.

[86] H. G. C. Traven. A neural network approach to statistical pattern classi£cation by semiparametric estimation of probability density functions. *IEEE Trans. Neural Networks*, 2:366–377, 1991.

[87] N. Ueda and R. Nakano. Deterministic annealing em algorithm. *Neural Networks*, 11:271–282, 1998.

[88] S. Ullman. Aligning pictorial descriptions: An approach to object recognition. *Cognition*, 32(3):193–254, 1989.

[89] S. Umeyama. Parameterized point pattern matching and its application to recognition of object families. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(1):136–144, Jan. 1993.

[90] A. Utsugi. Density estimation by mixture models with smoothing priors. *Neural Computation*, 10:2115–2135, 1998.

[91] M. Vaillant and C. Davatzikos. Hierarchical matching of cortical features for deformable brain image registration. In *Proceedings of Information Processing in Medical Imaging–IPMI 99*, volume 1613, pages 182–195. Springer-Verlag, 1999.

[92] M. Vaillant, C. Davatzikos, and R. Bryan. Finding 3D parametric representations of the deep cortical folds. In A. Amini, F. L. Bookstein, and D. Wilson, editors, *Proc. of the Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 151–159. IEEE Computer Society Press, 1996.

[93] P. Van den Elsen. *Multimodality matching of brain images*. PhD thesis, Utrecht University, Utrecht, Netherlands, 1993.

[94] G. Wahba. *Spline models for observational data*. SIAM, Philadelphia, PA, 1990.

[95] Y. Wang and L. H. Staib. Boundary £nding with prior shape and smoothness models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):738–743, July 2000.

[96] W. Wells. Statistical approaches to feature-based object recognition. *Intl. J. Computer Vision*, 21(1/2):63–98, 1997.

[97] W. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–52, 1996.

[98] R. P. Woods, S. T. Grafton, J. D. Watson, N. L. Sicotte, and J. C. Mazziotta. Automated image registration: Ii. intersubject validation of liear and nonlinear models. *J. Computer Assisted Tomography*, 22:155–165, 1998.

[99] C. Xu, D. Pham, and J. L. Prince. Reconstruction of the central layer of the human cerebral cortex from mr images. In *Proceedings of Medical Image Computing and Computer-Assisteed Intervention–MICCAI 98*, pages 481–488, 1998.

[100] A. L. Yuille and N. M. Grzywacz. A mathematical analysis of the motion coherence theory. *Intl. J. Computer Vision*, 3(2):155–175, June 1989.

[101] A. L. Yuille and J. J. Kosowsky. Statistical physics algorithms that converge. Technical Report 92–7, Harvard Robotics Laboratory, 1992.

[102] A. L. Yuille and J. J. Kosowsky. Statistical physics algorithms that converge. *Neural Computation*, 6(3):341–356, May 1994.

[103] A. L. Yuille, P. Stolorz, and J. Utans. Statistical physics, mixtures of distributions, and the EM algorithm. *Neural Computation*, 6(2):334–340, March 1994.

[104] X. Zeng, L. H. Staib, H. Tagare R. T. Schultz, and J. S. Duncan. A new approach to 3d sulcal ribbon £nding from mr images. In *Proceedings of Medical Image Computing and Computer Assisted Intervention–MICCAI 99*, pages 148–157, 1999.

[105] X. Zeng, L. H. Staib, R. T. Schultz, and J. S. Duncan. Segmentation and measurement of the cortex from 3d mr images using coupled-surfaces propagation. *IEEE Transaction on Medical Imaging*, 18(10):927–937, 1999.