A Feature Registration Framework using Mixture Models

Haili Chui and Anand Rangarajan*
Departments of Electrical Engineering and Diagnostic Radiology
Yale University, New Haven, CT 06520, USA

Abstract

We formulate feature registration problems as maximum likelihood or Bayesian maximum a posteriori estimation problems using mixture models. An EM-like algorithm is proposed to jointly solve for the feature correspondences as well as the geometric transformations. A novel aspect of our approach is the embedding of the EM algorithm within a deterministic annealing scheme in order to directly control the fuzziness of the correspondences. The resulting algorithm—termed mixture point matching (MPM)—can solve for both rigid and high dimensional (thin-plate spline-based) non-rigid transformations between point sets in the presence of noise and outliers. We demonstrate the algorithm's performance on 2D and 3D data.

1 Introduction

Feature-based registration problems frequently arise in the domains of computer vision and medical imaging. With the salient structures in two images represented as compact geometrical entities (e.g. points, curves, surfaces), we need to find the spatial *transformation/mapping* as well as the *correspondence* between them. Point features, represented basically by the point locations, are the simplest form of features. However, the resulting point matching problem can be quite difficult because of various factors.

One common factor is noise arising from the processes of image acquisition and feature extraction. The presence of noise makes it difficult to decide on the extent to which the features should be exactly matched. Another factor is the existence of outliers—many point features may exist in one point-set that have no corresponding points (homologies) in the other and hence need to be rejected during the matching process. Finally, the geometric transformations may need to incorporate high dimensional non-rigid mappings in order to account for deformations of the point-sets. Consequently, a general point feature registration algorithm needs to address all these issues. It should be able to solve for the correspondences between two point-sets, reject outliers and determine a good rigid or, if needed, non-rigid transformation that can map one point-set onto the other.

Various methods have been proposed to attack the point matching problem. Earlier methods used heuristic-driven approaches (often relying on empirically determined parameters) which are hard to generalize. On the other hand, maximum likelihood or Bayesian approaches have proven to be very useful theoretical frameworks. Such probabilistic models equip us with a set of mature and powerful techniques with a sound philosophical basis. For example, parameter estimation can then be approached as a standard hypothesis testing problem resulting in a more systematic treatment. When objective functions are derived from maximum likelihood or Bayesian principles, all the assumptions are clear and on the table. All the domain dependent knowledge and requirements are conveniently incorporated leaving room for principled generalizations.

This paper closely follows the authors' previous work [12, 2, 3], where point matching is interpreted as a combined linear assignment—least squares optimization problem and solved using two techniques, namely the softassign [9] and deterministic annealing [8]. In this paper, we show that point matching can be re-interpreted as a mixture density estimation problem and solved in an EM-like fashion. The idea of treating the point matching problem in such a probabilistic manner is actually quite similar to previous approaches proposed in [14, 5, 11], which also used mixture models as well as the EM algorithm to solve the feature matching problem. However, [14]'s algorithm is limited to be a local refinement tool in a larger object recognition problem. Both [14] and [5] only solve for rigid transformations. The approach in [11], which was designed for hand-written character recognition, combines mixture modeling for the correspondences, an affine geometric mapping and an object model via B-splines. While the mapping is still not non-rigid, object deformation is handled by the B-splines.

In this paper, we show that the mixture model can be used to develop a general point matching framework,

*Corresponding author: anand@noodle.med.yale.edu

namely mixture point matching (MPM), which includes a unified probabilistic treatment of noise, outliers and the regularization parameters. Also, as a methodological difference from others, we use deterministic annealing to simplify the mixture model and to enhance the robustness by directly controlling the fuzziness of the correspondence. The mixture point matching framework is designed to be general and can be applied to both rigid and non-rigid point matching problems. Interestingly, the resulting algorithm is in fact quite similar to the popular iterative closest point (ICP) algorithm [1, 7], which uses the closest neighbor heuristic to estimate the correspondence. Instead of treating the correspondence as a probabilistic variable, ICP always assigns binary correspondences. We show that ICP's over-simplified treatment of the correspondence directly leads to poorer performance.

2 Point Matching as Mixture Density Estimation

2.1 Mixture Model for Points

In this section, we discuss an asymmetric point matching case where one point set serves as a *template* with sparsely distributed points while the other set has relatively larger number of densely distributed points and serves as *data*. We will denote the template point-set as V consisting of points $\{v_a\ , a = 1, 2 \ldots K\}$ and the data point-set as X consisting of $\{x_i\ , i = 1, 2, \ldots, N\}$. For the sake of simplicity, we will assume that the points are in 2D. In our experiments, we do not impose this restriction.

Suppose the transformation can be abstractly represented 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 type of setup basically forms a template matching problem, which is quite common in many object recognition applications. The template represents the object that has to be identified and is defined 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 model to get the best fit of the model to the part of the data that truly corresponds to it. Inspired by the asymmetric nature of the setup, we use the Gaussian mixture density (as in [14]) to model the probabilistic distribution of the data points, while the Gaussian clusters centers are determined by the template points. The whole data point set's probability (likelihood) is the following:

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

where,

$$\begin{aligned} & p(x_i | f(v_a, \alpha), \Sigma_a) \\ = \quad & \frac{1}{2\pi |\Sigma_a|^{\frac{1}{2}}} e^{-\frac{(x_i - f(v_a, \alpha))^T \Sigma_a^{-1} (x_i - f(v_a, \alpha))}{2}}. \end{aligned} \qquad (2)$$

There are two sets of parameters associated with the Gaussian mixture models. The first set of parameters are the covariance matrices (denoted as Σ composed of Σ_a). The second set are "occupancy" parameters (denoted as π composed of π_a). Each π_a tells us the percentage of the data which belong to a certain Gaussian cluster.

To account for the spurious points (outliers) in the data, we need an extra outlier cluster for 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:

$$p(X|V, \alpha, \Sigma, \pi) = \prod_{i=1}^{N} \sum_{a=1}^{K+1} \pi_a p(x_i | f(v_a, \alpha), \Sigma_a). \qquad (3)$$

2.2 Point Matching as a MAP Problem

Different problems require different types of transformations. The choice of using a particular type of transformation reflects 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 reguarization* 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.

$$p(\alpha) = \frac{e^{-\lambda ||Lf||^2}}{Z_{part}} \qquad (4)$$

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:

$$p(\alpha | X, V, \Sigma, \pi) = \frac{p(X|V, \alpha, \Sigma, \pi) p(\alpha)}{p(X)}. \qquad (5)$$

The term $p(X)$ can be omitted since it doesn't depend on α. To find the best fit, we need to solve the maximum *a posteriori* (MAP) problem associated with (5). This is equivalent to minimizing the following log-posterior energy function.

$$E_1(\alpha, \Sigma, \pi) = -\log p(X|V, \alpha, \Sigma, \pi) - \log p(\alpha). \qquad (6)$$

2.3 EM Algorithm and Complete Data Posterior Energy Function

The Expectation-Maximization (EM) algorithm is an elegant algorithm to solve this MAP problem [10]. It introduces an intermediate binary variable Z (*latent data*) with elements z_{ai}. The variable Z represents the classification information of the data point x_i. For example, $z_{ai} = 1$ implies that data point i belongs to the ath Gaussian cluster. The variable Z essentially plays the same role as the correspondence. With the help of the latent variable Z, the minimization of the original log-posterior energy function (also termed the *incomplete data posterior energy function*) in (6) can be converted into the minimization of the *complete data posterior energy function* [10].

$$
\begin{aligned}
& E_2(M, \alpha, \Sigma, \pi) \\
= \; & \sum_{i=1}^{N} \sum_{a=1}^{K+1} \frac{m_{ai}[x_i - f(v_a, \alpha)]^T \Sigma_a^{-1} [x_i - f(v_a, \alpha)]}{2} \\
& + \sum_{i=1}^{N} \sum_{a=1}^{K+1} m_{ai} \log m_{ai} - \sum_{i=1}^{N} \sum_{a=1}^{K+1} \frac{m_{ai}}{2} \log |\Sigma_a| \\
& - \sum_{i=1}^{N} \sum_{a=1}^{K+1} m_{ai} \log \pi_a + \lambda ||Lf||^2 \qquad (7)
\end{aligned}
$$

where m_{ai} satisfies $\sum_{a=1}^{K+1} m_{ai} = 1$ for $i \in \{1, 2, \ldots, N\}$ with $m_{ai} > 0$.

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 minima of the original incomplete data posterior energy function.

The first 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:

$$
m_{ai} =< z_{ai} >= \frac{\pi_a p(x_i | f(v_a, \alpha), \Sigma_a)}{\sum_{b=1}^{K+1} \pi_b p(x_i | f(v_b, \alpha), \Sigma_b)}. \qquad (8)
$$

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 (consisting of $\{m_{ai}\}$) can be regarded as a fuzzy estimation of the binary classification/correspondence Z.

The second step is called the *M-step.* Given a fixed 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.

3 Mixture Point Matching: Energy Function and Algorithm

3.1 Problems with Mixture Model and EM Algorithm

Even though the combination of the mixture model and the EM algorithm is a powerful tool for probability density estimation, it is, in its current state, rather ill-suited for the task of non-rigid point matching. We found that including the covariance and the occupancy parameters as free parameters as in the standard mixture model is problematic. First, these parameters are not closely related our purpose of registration. Secondly, additional parameters can cause more local minimum and make the optimization harder.

This problematic aspect of mixture modeling (as density estimation) is demonstrated through a clustering experiment. We use 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 predefined template clusters, we cluster the data points to find our template, i.e., we let the template 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 final configurations found by the EM algorithm are shown in the first row in Figure 1. With the extra flexibility provided by the covariance and the occupancy parameters, it is clear that the answers can be quite different. This is a problem specifically for non-rigid point matching because the template can then be transformed in different ways and still fit the data quite well.

3.2 A Simplified Mixture Model

To overcome the problem caused by these extraneous parameters, we need a simplified mixture model.

We first put a direct control on the covariance parameter Σ through deterministic annealing. The idea comes from the following observations. When the values of the covariance matrices $\{\Sigma_a\}$ are large, the Mahalanobis distance in (7) between any pair of points x_i and $u_a = f(v_a, \alpha)$ is close to zero. The resulting estimation of correspondence $\{m_{ai}\}$ are then nearly uniform (values close to $\frac{1}{K+1}$). It means that points are allowed to match to many cluster centers (including those which are far away) with a fuzzy degree of confidence. On the other hand, if the magnitudes of the covariance matrices are very small, $\{m_{ai}\}$ will be close to binary

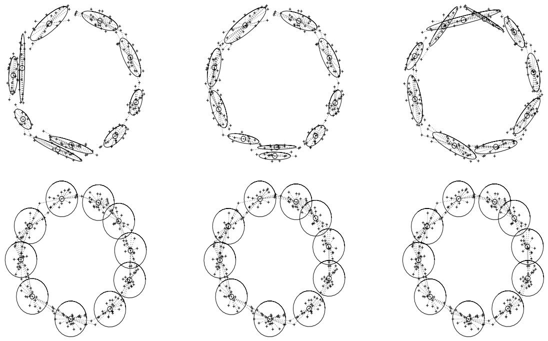

Figure 1: **Top Row**: The clusters estimated by the standard EM algorithm. **Bottom Row**: Modified EM algorithm (with annealing) results.

and the points will only be allowed to match to the closest cluster centers. These observations suggests that the covariance parameter Σ is basically acting like a *search range* parameter. When the covariance magnitudes are big, the template cluster centers search globally for possible member data. When the covariance magnitudes are small, the template cluster centers look for candidates at a much more localized level. If we set up a scheme for the covariance variable so that its magnitude gradually goes from large to small, it clearly would lead to a global-to-local match strategy.

We use the isotropic form for the covariance matrix set Σ and represent the magnitude by a newly introduced *temperature* parameter T. The annealing is accomplished by gradually reducing T in the point matching process. The original Mahalanobis distance measure is now simplified 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. So instead of using T, we keep the temperature for the outlier class at a large constant value T_0. The outlier center v_{K+1} is set to be at x_0, which represents the center of mass of all data points. We do not need to apply the non-rigid transformation to the outlier cluster. For the sake of simplicity, however, we continue to use the notion $f(v_{K+1},\alpha)$.

Since we do not have any 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 3.1. The annealing process is performed by gradually lowering T with a pre-specified schedule. As shown in the second row in Figure 1, the clusters found through the modifie mixture model are all evenly distributed. They are also much more consistent with little variations due to initial conditions or symmetry breaking perturbations.

3.3 Mixture Point Matching Energy Function

After making the modifications described above to the complete data posterior energy function in (7) and eliminating some constants, we get the final *mixture point matching energy function*:

$$\begin{aligned} & E_3(M,\alpha) \\ = \; & \sum_{i=1}^{N} \sum_{a=1}^{K+1} m_{ai} ||x_i - f(v_a,\alpha)||^2 \\ & + T \sum_{i=1}^{N} \sum_{a=1}^{K+1} m_{ai} \log m_{ai} - T \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \log T \\ & - T \sum_{i=1}^{N} m_{K+1,i} \log T_0 + \lambda T ||Lf||^2 \end{aligned} \qquad (9)$$

where m_{ai} satisfies $\sum_{a=1}^{K+1} m_{ai} = 1$ for $i \in \{1, 2, \ldots, N\}$ with $m_{ai} > 0$ (except $m_{K+1,N+1} \equiv 0$).

The reader may notice that the regularization parameter λ is no longer a constant but is instead modulated by T. When the temperature is high, the transformation is heavily regularized by the dominant prior term. When the temperature is low, the transformation is more influenced by the data, as the influence 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.

3.4 Mixture Point Matching Algorithm

The resulting mixture point matching algorithm (MPM) is quite similar to the EM algorithm and very easy to implement. It essentially involves a dual update process embedded within an annealing scheme. We briefly describe the algorithm.

E step: Update the correspondence.

$$m_{ai} = \frac{p(x_i | f(va,\alpha), \Sigma_a)}{\sum_{b=1}^{K+1} p(x_i | f(v_b,\alpha), \Sigma_b)} \qquad (10)$$

For the model clusters $a = 1, 2, \ldots K$,

$$p(x_i | f(v_a,\alpha), \Sigma_a) = \frac{1}{2\pi T} e^{-\frac{(x_i - f(v_a,\alpha))^T (x_i - f(v_a,\alpha))}{2T}} \qquad (11)$$

and for the outlier cluster $a = K + 1$,

$$p(x_i|v_{K+1}, \Sigma_{K+1}) = \frac{1}{2\pi T_0} e^{-\frac{(x_i - x_0)^T (x_i - x_0)}{2T_0}} \qquad (12)$$

The normalization of m_{ai} over all clusters (the original template clusters plus the outlier cluster) introduce competition. Unless a data point shows strong evidence that it belongs to one of the non-outlier clusters, it is rejected as an outlier. The competition enables the algorithm to automatically evaluate all the evidence and reject outliers. For the purpose of symmetry breaking, perturbations (Gaussian noise with zero mean and very small standard deviation) are added to m_{ai} after the normalization.

M step: Update the transformation. Solve the following least-squares problem,

$$\min_{\alpha} E_4(\alpha) = \min_{\alpha} \sum_{a=1}^{K} ||y_a - f(v_a, \alpha)||^2 + \lambda T ||Lf||^2 \quad (13)$$

where,

$$y_a = \sum_{i=1}^{N} m_{ai} x_i \qquad (14)$$

After dropping the terms which are independent of α, we arrive at this simple form by utilizing the simplex constraint on m_{ai}, namely $\sum_{a=1}^{K+1} m_{ai} = 1$. The variable y_a can be regarded as our newly estimated positions of the cluster centers for the data.

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 dual updates are repeated until convergence at each temperature. The parameter T is then lowered and the process is repeated until some final temperature T_{final} is reached.

The parameter T_0 is set to the largest square distance of all point pairs. 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 always kept at T_0. To prevent overfitting, the parameter T_{final} should be set according to the amount of noise within the data set. In this work, we make a simplified 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.

We call the resulting algorithm *Mixture Point Matching algorithm (MPM)*.

```
MPM Algorithm pseudo-code:
    Initialize parameters $T$, $\alpha$.
    Alternating update:
        – E-step to update $m_{ai}$.
        – M-step to update $\alpha$.
    Lower $T = T \cdot r$ until $T_{final}$ is reached.
```

4 Applications and Experiments

In this section, we apply the MPM algorithm to different feature matching problems encountered in the domain of medical imaging and conduct comparisons with ICP.

4.1 Rigid Point Matching

First, we use the MPM algorithm to solve for a 2D rigid mapping (rotation and translation). When applied to a point v_a , the transformed point is as follows,

$$f(v_a, \theta, t) = R(\theta) \cdot v_a + t \qquad (15)$$

where θ is the rotation angle, $R(\theta)$ is the 2D rotation matrix and $t = (t_x, t_y)$ is the 2D translation. The solution for θ and t for (13) is quite straight forward [12]. Incorporating this rigid transformation into the MPM framework is straightforward. The only difference is that we do not need the regularization term in the mixture point matching energy funciton any more since this is a rigid transformation.

The high resolution autoradiographs for non-human primates' brain carries rich information about cortical activity. One problem that we worked on in the past is to automatically align thousands of these 2D autoradiograph slices (to form a 3D stack) by matching edge features [12]. One such case is shown in Figure 2. The matching process, as shown in Figure 2, clearly demonstrates MPM's ability to handle large numbers of outliers.

To provide more quantitative evaluations, we also test the algorithm on many synthetic examples. The template point set, which was originally in perfect alignment with the data, was first randomly transformed (random rotation θ^{true} sampled from a uniform distribution within $[-r_\theta, r_\theta]$, $r_\theta = 45^o$, and translation t^{true} within $[-r_t, r_t]$, $r_t = 100$). Then, different amounts of outliers or noise are added to the originally clean data set to get the *target* data set. Both MPM and ICP are used to align the transformed template with the target data. 100 such random experiments were run for each setting. The error between the known true transformation parameters $(\theta^{true}, t^{true})$ and the ones recovered by the algorithm (θ, t) was computed as $error = mean(\frac{abs(\theta^{true} - \theta)}{r_\theta}) + mean(\frac{abs(t^{true} - t)}{r_t})$. The results are shown in Figure 3. ICP's performance deteriorates much

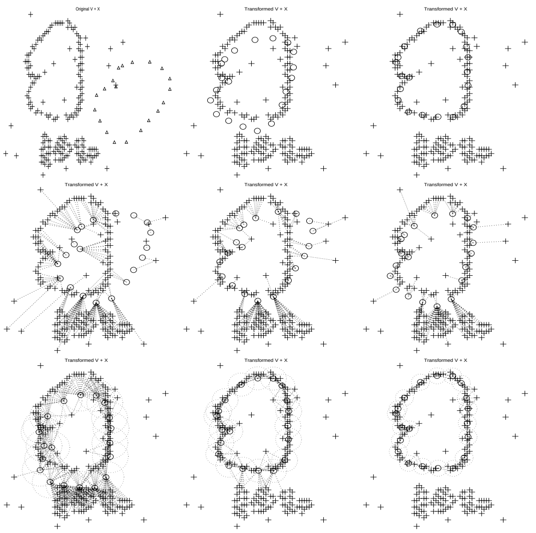

Figure 2: **Rigid Point Matching Example. Top Left**: Original position of the template V (triangles) and the data X (crosses). **Top Middle:** ICP result. The transformed template U (circles) are shown with the data. **Top Righ**t: MPM result. **Middle Row:** Sequential intermediate states of ICP. Most significant correspondences are shown as dotted links. **Bottom Row**: Sequential intermediate states of MPM. A dotted circle of radius $\sqrt{T}$ is drawn around each transformed template point to show the annealing process.

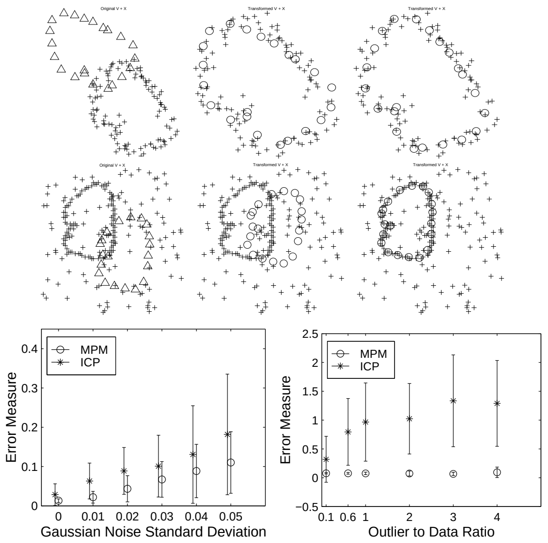

Figure 3: **Synthetic Experiments. Top Left:** The original position of two point-sets from a noise experiment. **Top Middle:** ICP result. **Top Right:** MPM result. **Middle Row:** Original position, ICP result and MPM result from an outlier experiment. **Bottom Row:** The error statistics. Notice that with large amounts of noise, the original alignment may no longer be optimal. That partially causes MPM's errors also to increase.

faster than MPM. The difference is most evident in the case of outliers.

4.2 Non-rigid Point Matching

The point matching problem becomes more interesting and more challenging when non-rigid transformations are involved. While much of the recent efforts have been focused on learning the *statistics* of non-rigid *shapes* [4, 6], the non-rigid matching algorithms are still mostly based on some *local* search heuristics like ICP. The problem of tracking correspondences over *long ranges* (especially in the case of large deformations), as first presented in [15], was largely neglected. In this work, we choose a relatively simple non-rigid parameterization model while direct more of our attention towards the general non-rigid matching problem.

We use the thin-plate spline (TPS) to parameterize the non-rigid mapping. TPS is chosen because it is the only spline that can be cleanly decomposed into affine (rigid) and non-affine (non-rigid) subspaces while enforcing a simple smoothness constraint based on the second derivative of the mapping. The transformation is parameterized by two matrices (d, w):

$$f(v_a; d, w) = v_a \cdot d + \phi(v_a) \cdot w \qquad (16)$$

where d is a $(D+1) \times (D+1)$ (D=2 in the case of 2D) matrix representing the affine transformation and w is a $K \times (D+1)$ warping coefficient matrix representing the non-affine deformation. The vector $\phi(v_a)$ is related to the thin-plate spline kernel with its entries $\phi_b(v_a)$ defined as $\phi_b(v_a) = ||v_a - v_b||^2 \log ||v_a - v_b||$ in 2D. The template point-set is used as *control points* in the TPS kernel. So loosely speaking, the TPS kernel contains the information about the point-set's internal structural relationships. When combined with the warping coefficients w, it generates a non-rigid warping. In the case of TPS, the abstract smoothness term in (9) now becomes:

$$\lambda T ||Lf||^2 \mapsto \lambda T \, \text{trace} \, (w^T \Phi w)$$

where the weight parameter λ is a constant and Φ is just a concatenated version of $\phi_b(v_a)$. The more detailed close form solution for (10) for TPS can be found in [13, 3].

A typical non-rigid point matching example is shown in Figure 4. The non-rigid matching process yields interesting insights into the algorithm's nature.

At a very high temperature T, the correspondence $\{m_{ai}\}$ is almost uniform. The estimated corresponding point-sets $\{y_a = \sum_{i=1}^{N} m_{ai} x_i\}$ are essentially very close to the *center*

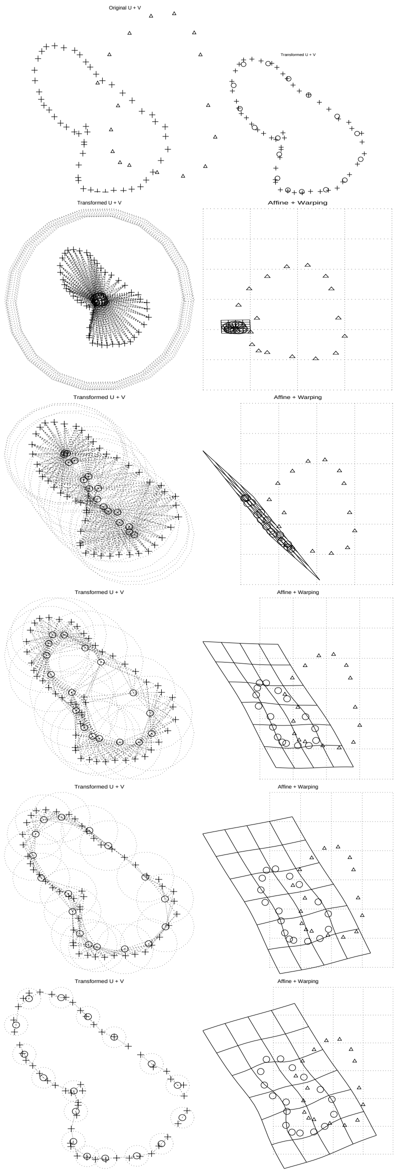

Figure 4: **Non-rigid Point Matching Example. Top Left:** Original position of the template$V$ (triangles) and the data X (crosses). There obviously is non-rigid deformation between the 2 shapes. **Top Right:** MPM result. The transformed template U (circles) are shown with the data.
Rows 2 to 6: MPM Matching Process. Left: The currently estimated correspondences between the transformed template U (circles) and the data (crosses). A dotted circle of radius $\sqrt{T}$ is also drawn around each point in U to show the annealing process. **Right:** Deformation of the space: dotted regular grid and the solid deformed grid, original template V (triangles) and the transformed template U (circles).

of mass of V. This helps us recover most of the translation needed to align the two point-sets. This can be seen from the second row in Figure 4. At slightly lower temperatures, as shown in the third row, the algorithm starts to behave like a principal axis method. The model points are rotated and aligned with the major axis along which the points in the data are most densely distributed. Lowering the temperature even more, we observe that the algorithm finds more localized correspondences which makes it possible to capture the more detailed structures within the target point-set. The last row shows that algorithm almost converges to a nearly binary correspondence at a very low temperature and the correct non-rigid transformation is fully recovered.

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

4.3 3D Non-rigid Sulcal Point Matching

The registration of cortical 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 3D sulcal feature points [2]. It is a difficult 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.

We use MPM to solve for the TPS on 5 MRI sulcal point-sets (acquired from 3D MRI volumes [2]). 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 [13].

Overlays of all sulcal point-sets before and after the matching are shown in Figure 5. For comparison, we also include the results for both the TPS and the affine by our previous RPM algorithm [3, 2]. Denser, tighter packing of the sulci indicates better registration. Though the result of TPS-MPM clearly shows an improvement over affine-RPM alignment, it is actually slightly worse than the result of TPS-RPM [3]. We will discuss the difference between the two methods (MPM and RPM) in the next section.

5 Discussion and Conclusion

The mixture point matching (MPM) framework combines the merits of the deterministic annealing technique and the mixture modeling. Deterministic annealing enables us to have direct control over the fuzziness of the correspondence and, as demonstrated in this work, endows our

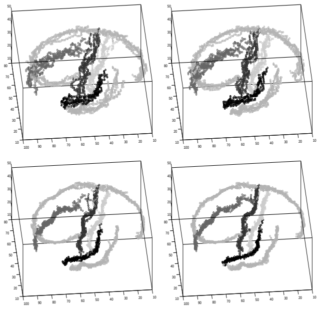

Figure 5: **Top Left:** Original overlay of the 5 sulcal point-sets. Only 6 sulci on one half of the brain are shown here. **Top Right:** Result of affine alignment by RPM. **Bottom Left:** Result of TPS alignment using MPM. **Bottom Right:** Result of TPS alignment using RPM.

algorithm with an *emergent*, scale-space property. Mixture modeling adds a probabilistic treatment of the outliers and of the regularization parameters. The resulting MPM algorithm is shown to be fast, robust, and easy to implement.

In our previous work [12, 2, 3], we have used a robust point matching algorithm (RPM) which does not have such a clear probabilistic modeling framework. Despite this, when it comes to the actual algorithms, the only major difference between RPM and MPM is that RPM enforces a double normalization over the membership/correspondence matrix M or $\{m_{ai}\}$, thereby enforcing a one-to-one rather than many-to-one correspondence between the points.

Note that the Gaussian mixture model treats the model and data in a symmetrical manner. The only thing that differentiates them is the way we use the one way constraint. If we use the constraint $\sum_i m_{ai} = 1$ instead of $\sum_a m_{ai} = 1$, the roles of the model and the data are reversed. The differentiation between the sparse template and the dense data, which is implicit in the mixture model, is not always appropriate. When there are roughly equal numbers of points in the data and the model with outliers in both, the many-to-one notion of the mixture model is not the right model.

We believe that the double normalization used in the RPM would be useful in this case. With the double normalization (with 2 outlier clusters for both sides), we can think of it as fitting a mutual-mixture model where each point-set is acting as both data and template. Such a model would still enjoy the benefits of probabilistic modeling as in the case of a standard mixture model thereby enabling an EM-like algorithm to overcome local minima. We plan to carefully investigate these issues in our future work.

Acknowledgements

This work was partially supported by an NSF grant to A.R. We thank Jim Rambo and Bob Schultz for assistance with the anatomical sulcal brain MRI data and Hemant Tagare for helpful discussions.

References

[1] 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.

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

[3] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE Press, 2000. (in press).

[4] 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.

[5] 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.

[6] N. Duta, A. K. Jain, and M. Dubuisson-Jolly. Learning 2d shape models. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 8–14, 1999.

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

[8] 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.

[9] S. Gold, C. P. Lu, A. Rangarajan, S. Pappu, and E. Mjolsness. New algorithms for 2-D and 3-D point matching: pose estimation and correspondence. Technical Report YALEU/DCS/RR-1035, Department of Computer Science, Yale University, May 1994. To appear in *Advances in Neural Information Processing Systems 7*.

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

[11] 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.

[12] 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.

[13] G. Wahba. *Spline models for observational data*. SIAM, Philadelphia, PA, 1990.

[14] W. Wells. Statistical approaches to feature-based object recognition. *Intl. J. Computer Vision*, 21(1/2):63–98, 1997.

[15] 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.