A New Algorithm for Non-Rigid Point Matching

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

Abstract

We present a new robust point matching algorithm (RPM) that can jointly estimate the correspondence and non-rigid transformations between two point-sets that may be of different sizes. The algorithm utilizes the softassign for the correspondence and the thin-plate spline for the non-rigid mapping. Embedded within a deterministic annealing framework, the algorithm can automatically reject a fraction of the points as outliers. Experiments on both 2D synthetic point-sets with varying degrees of deformation, noise and outliers, and on real 3D sulcal point-sets (extracted from brain MRI) demonstrate the robustness of the algorithm.

1 Introduction

It is a fundamental yet still open problem in computer vision to match two point-sets, i.e, to find the geometric mapping and correspondence between two sets of points in 2D or in 3D [11]. There are various factors that make the point matching problem difficult. One such factor is the existence of outliers—many point features may exist in one point-set that have no corresponding point (homology) in the other. Secondly, the geometric transformations may need to incorporate high dimensional non-rigid mappings in order to account for deformation. Such deformable matching problems frequently arise in medical imaging [21] and in computer vision [17]. Consequently, non-rigid point matching requires us to solve for the correspondences between two point-sets, reject outliers that have no homologies and determine a non-rigid mapping that can warp one point-set onto the other.

Non-rigid mappings call for high-dimensional transformations of the point-sets. To make the problem more tractable, some existing point matching algorithms attempt to constrain the mapping or the set of correspondences or both. For example, the mapping can be approximated by articulated affine transformations. The set of correspondences can also be constrained by restricting the point-sets to lie along curves (ordering constraint). Also, it is quite common to use nearest-neighbor methods to assign correspondence (as in the iterated closest point (ICP) algorithm [2]). Unfortunately, these methods generate local minima and do not usually guarantee that correspondences are one-to-one. Motivated by these observations, we feel that there is a need for new point matching algorithms that can capture non-rigid deformations while solving for (one-to-one) correspondence and rejecting a fraction of points as outliers.

We should mention a different take on this problem which involves recasting point matching as inexact, weighted graph matching [20]. By considering points along with their inter-relationships and by using attributes and multiple link types, the search for correspondence can be alleviated [9]. However, there remains the problem of graph definition. Interrelationships, attribute and link type information can be notoriously brittle and context dependent. Consequently, in this work, we solely make use of point location information.

Our approach to non-rigid point matching closely follows our earlier work on joint estimation of pose and correspondence using the *softassign* [10] and *deterministic annealing* [8]. This earlier work resulted in the *robust point matching* (RPM) algorithm [16] which was restricted to using affine and piecewise-affine spatial mappings. Here, we extend the pose parameterization to include non-rigid deformations of the point-sets. The *thin-plate spline* (TPS) is used to parameterize the non-rigid mapping [22]. TPS is chosen because it is the only spline that can be cleanly decomposed into affine and non-affine subspaces while minimizing a bending energy based on the second derivative of the mapping. In this sense, the TPS map can be considered to be a natural non-rigid extension of the affine map. Bookstein [3] pioneered the usage of TPS for mapping landmarks (points with known correspondence). It has proven to be a very useful tool in shape

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

analysis. Such a tool would certainly be more powerful if the limitation of known correspondence can be lifted. Our new TPS-RPM algorithm simultaneously solves for the correspondences, rejects a fraction of the points as outliers and determines a thin-plate spline mapping between the point-sets.

2 Previous Work

Due to our interest in point matching, we restrict our focus to the class of feature-based non-rigid matching methods. One way of dividing the space of feature-based methods is in terms of sparse versus dense features. Labeled landmark points are the most popular kind of sparse features since (as mentioned previously) non-rigid matching of landmarks does not require a solution to the point-to-point correspondence problem [3]. These methods are quite sensitive to the number and choice of landmarks.

Dense feature-based matching methods run the gamut of matching points, lines, curves and surfaces [2]. These methods usually begin with an object parameterization. Then, the allowable ways in which the object can deform is specified [14, 17]. The methods that fall into this class differ in object parameterizations and in the specification of the allowed deformations. In most cases, curves and/or surfaces are first fitted to features extracted from the images and then matched [14, 21, 7]. These methods work well when the surfaces/curves to be matched are reasonably smooth. Also, even with good feature extraction, the surface/curve fitting step can be very difficult when the shapes involved become complex. The main advantage is that curve correspondence is easier than point correspondence.

In this paper, we use an integrated non-rigid mapping and correspondence formulation. A similar integrated approach, but with no accompanying algorithm, was first presented in [24]. While the correspondence problem has a long history in rigid, affine and projective point matching [11], there is relatively a dearth of literature on non-rigid point matching. Recently, there has been some interest in using point-based correspondence strategies in non-rigid matching [17, 7]. The modal matching approach in [17] relies on the point correspondence approach pioneered in [18] and further developed in [19]. The basic idea is to use two pairing matrices that are built up from the Gaussian of the pairwise distances between point features in either set. The major deformation modes [5] are obtained from these matrices and can be used to find the correspondence. In a way, the matrices indirectly model the points' spatial relationships. Such inter-relationships can be directly modeled as well through a graph representation. In [6], after building a graph representation from Delaunay triangulations, the search for correspondence is accomplished using inexact graph matching [20]. However, the spatial mappings are restricted to be affine or projective. In [1], decomposable graphs are hand-designed for deformable template matching and matched via dynamic programming. However, the graphs are not automatically generated and there is no direct relationship between the deformable model and the graphs that are used. In [13], a maximum clique approach is used to match relational sulcal graphs. Again, the graphs are hand designed and not related to spatial deformations.

3 A Non-Rigid Mapping Energy Function

First, we review thin-plate splines with an eye toward integrating it with our softassign correspondence engine. Later, we develop the full-blown non-rigid mapping and correspondence energy function from which we derive the TPS-RPM algorithm.

3.1 Thin-Plate Splines

Assume we have two sets of of *corresponding points* (in either 2D or 3D) X and V, consisting of points $\{x_a, a = 1, 2, \ldots, K\}$ and $\{v_a, a = 1, 2, \ldots, K\}$ respectively. For the sake of clarity, we present the 2D formulation alone. Each v_a represents the location of a point with the homogeneous coordinate notation, $v_a = (1, v_{ax}, v_{ay})$. If we denote the dimensionality as D ($D = 2$ for 2D here), v_a is a $(D+1)$ sized vector. The thin-plate spline fits a mapping function $f(v_a)$ between corresponding point-sets X and V by minimizing the following energy function:

$$E_{TPS}(f) = \sum_{a=1}^{K} ||x_a - f(v_a)||^2$$

$$+\lambda \int\int \left[(\frac{\partial^2 f}{\partial x^2})^2 + 2(\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2 \right] dxdy. \quad (1)$$

The point-set V or $\{v_a\}$ is mapped as closely as possible to the point-set X or $\{x_a\}$ by minimizing the first error measurement term. Since the mapping is non-rigid, there are an infinite number of such mappings f that can minimize the first term. The second term, which is essentially a smoothness constraint, comes into play to regularize the mapping. The extent of the warping depends on the regularization parameter λ. Only when λ is close to zero do we get exact matching of corresponding points.

For this energy function, at a fixed λ, there exists a unique minimizing function f (for any location v in $\Re^2$) specified by two parameter matrices d and w:

$$f(v) = v \cdot d + \phi(v) \cdot w \tag{2}$$

where d is a $(D+1) \times (D+1)$ *affine transformation matrix* and w is a $K \times (D+1)$ *non-affine warping coefficient matrix.* The vector $\phi(v)$ is related to the thin-plate spline *kernel.* It is a $1 \times K$ vector for any v, where each entry $\phi_a(v) = c||v - v_a||^2 \log ||v - v_a||$ with c being a constant [22]. Loosely speaking, the thin-plate spline kernel contains the information about the point-set's internal structural relationships and generates a non-rigid warp when combined with the warping coefficient matrix w.

A special characteristic of the thin-plate spline is that the resulting transformation is always decomposed into a global affine transformation and a local non-affine warping component. The smoothness term above is solely a function of the warping components. This is a desirable property since the rotation, translation and global shear components (which are included in the affine transformation) are then not penalized. When the solution for f in (2) is substituted into the TPS energy function in (1), we get:

$$E_{TPS}(d, w) = ||X - Vd - \Phi w||^2 + \lambda \,\mathrm{trace}\,(w^T \Phi w) \tag{3}$$

where X and V are just concatenated versions of the point coordinates x_a and v_a, and Φ is a $(K \times K)$ matrix formed from the $\phi(v_a)$. Each row of each newly formed matrix comes from the original vectors. The progression from (1) to (3) mirrors a transition from the standard TPS functional to an energy function defined on the affine (d) and warping (w) parameters.

3.2 Joint Optimization of the Mapping and Correspondence

We have shown above that once we have two sets of corresponding points, we can fit a thin-plate spline interpolant between them. We now take up the question of what happens when we do not know the correspondence between the two point-sets.

An interesting feature of the point matching problem (as formulated here) is that even though the simultaneous solution of mapping and correspondence is quite difficult, solving for one factor while keeping the other fixed is trivial. Given the correspondence, finding the optimal transformation is a simple

m_{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 1: **An Example of the Correspondence Matrix.** Points v_1 and v_2 correspond to $x_1$and x_2 respectively, and the rest of the points are outliers.

least-squares problem as seen from (3) above. Given the non-rigid mapping, finding the correspondence between the point-sets is accomplished by solving a linear assignment problem [10, 12]. Inspired by this fact, a lot of algorithms have tried to incorporate an update scheme wherein the mappings and the correspondences are alternately optimized while keeping the other fixed [2, 6, 7, 23]. Our algorithm follows this basic idea. By alternating the update of the mapping and correspondence parameters, it is expected that the two solutions mutually improve one another during the process and finally converge to a reasonable, albeit sub-optimal solution.

We denote the correspondence parameter by a matrix M consisting of m_{ai}. Given two point-sets $\{x_i, i = 1, 2, \ldots, N\}$ and $\{v_a, a = 1, 2, \ldots, K\}$, the inner $K \times N$ part of M carries the correspondence information between the point-sets. When a point x_i matches a point v_a, m_{ai} is equal to 1. Otherwise it is zero. To ensure one-to-one correspondence, each row and each column of M should sum to one. We also put in an extra row and an extra column in M to take care of the outliers so that the row and column summation constraints still hold. An example of the correspondence matrix is shown in Figure 1. The energy function with both the correspondence M and the thin-plate spline (d, w) is the following:

$$E(M, d, w) = \sum_{a=1}^{K} \sum_{i=1}^{N} m_{ai} ||x_i - v_a d - \phi(v_a) w)||^2 + \lambda \,\mathrm{trace}\,(w^T \Phi w) \tag{4}$$

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

We now briefly describe the gist of our approach for the sake of intuition. Rather than solve (4) using a binary valued correspondence matrix, we assume that the entries of M can take continuous, non-negative values. (We later show that binary correspondences are recovered at the end.) Our alternating update scheme is very simple. We start with a uniform initial value of M, with d and w set to zero. For the first step, we keep M fixed and fit a thin-plate spline (d, w) between $\{v_a\}$ and their currently corresponding points $\{y_a = \sum_{i=1}^{N} m_{ai} x_i\}$. (Note that we are not assuming correspondence. y_a is obtained based on the current estimate of M.) At the second step, we keep d and w fixed and calculate the match matrix M between the transformed $\{v'_a = f(v_a|d, w)\}$ and $\{x_i\}$ while making sure that M always satisfies the row and column summation constraints. Steps one and two are repeated until the algorithm converges.

An obvious question can be raised at this juncture. Our initial values of the parameters M, d and w can be far away from their optimal values. How can we be sure that this alternating update scheme actually improves the solution rather than getting worse? It turns out that two techniques, namely the softassign and deterministic annealing, which have also been used in our previous point matching algorithms, provides a solution. The details are in [9, 10]. Here we just focus on the main idea.

The basic idea of the softassign is to relax the correspondence variable to be continuous in the interval of $[0, 1]$, rather than being binary-valued, while ensuring that the row and column constraints are still satisfied. By making the correspondence continuous, partial matches between the points are allowed. In other words, the correspondence can be "soft" instead of "hard." The energy function becomes better behaved 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). The correspondences are always constrained to be fuzzy and doubly substochastic.

The other important technique is deterministic annealing [8]. Deterministic annealing is closely related to simulated annealing except that all operations are deterministic. An additional temperature parameter T is introduced which specifies the degree of fuzziness of the correspondence matrix: higher the temperature, greater the fuzziness. This is done by adding an entropy term $T \sum_{ai} m_{ai} \log m_{ai}$ to the energy function. At each temperature, the initial condition from the previous temperature is used and a straightforward deterministic descent is performed on the energy function. As the temperature is gradually reduced, the doubly substochastic correspondence matrix approaches a permutation matrix (with binary outlier rejection). In more formal terms, the energy function in (4) is a linear assignment or weighted bipartite matching problem [15] with respect to the correspondences. Softassign used within deterministic annealing has been shown to find the optimal solution to linear assignment [12]. However, while softassign and the TPS solution are independently optimal, the combination is not. Our approach is only guaranteed to find a local minimum for *both* the mapping and the correspondence.

3.3 The TPS-RPM Algorithm

Putting everything together, the final energy function which is actually minimized by our algorithm is as follows:

$$\begin{aligned} E(M, d, w) = & \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} ||x_i - v_a d - \phi(v_a) w)||^2 \\ & -\zeta \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} + T \sum_{i=1}^{N} \sum_{a=1}^{K} m_{ai} \log m_{ai} \\ & + \lambda_1 \, \mathrm{trace}\,(w^T \Phi w) + \lambda_2 \, \mathrm{trace}\,[(d - I)^T (d - I)] \end{aligned} \quad (5)$$

where $m_{ai} \in [0, 1]$ satisfies:

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

and

$$\sum_{a=1}^{K+1} m_{ai} = 1, \; for \; i = 1, 2, \ldots, N.$$

We briefly go through all the components of the energy function. The first term is just the error measure term. The second term with a parameter ζ is used to guard against null matches. We usually set it to a value close to zero. The third term is an $x \log x$ entropy barrier function with the temperature parameter T. The entropy barrier function ensures positivity of M. For more details regarding the choice of the barrier function, please see [10]. The fourth term is the standard thin-plate spline regularization term which penalizes the local warping coefficients w. We also use the fifth term which is not usually employed in the thin-plate spline literature, to constrain the affine mapping d by penalizing the residual part of d which is different from an identity matrix I. The main reason for this term is to constrain the affine from unphysical reflection mappings which can flip the entire plane. The behavior of these two terms is controlled by two parameters λ_1 and λ_2. Inspired by the annealing schedule on the correspondence, we also use a linear annealing schedule ($\lambda_i = \lambda_i^{init} T$, $i = 1, 2$) for these two parameters instead of using a fixed value. To provide more freedom for the affine transformation, λ_2^{init} is set to be much smaller than λ_1^{init}.

There are two major steps involved in our algorithm. With the currently estimated transformation (d, w), the first step is to update the correspondence. The closed form solution for M can be obtained just by differentiating the energy function in (5) w.r.t. M and setting the result to zero:

$$m_{ai} = e^{-\frac{||x_i - v_a d - \phi(v_a) w||^2 - \zeta}{T}} \tag{6}$$

The Sinkhorn balancing technique of alternating row and column normalizations is used to satisfy the row and column constraints [10]. When the correspondences are held fixed, we use a least-squares approach to solve for the TPS parameters (d, w) as our second step. This is implemented using a QR decomposition [22]. The two steps are repeated while gradually reducing the temperature T and the regularization parameters λ_1 and λ_2.

TPS-RPM Algorithm pseudo-code:

Initialize parameters M, d and w.
Initialize parameters λ_1, λ_2 and T.
Begin A: Deterministic Annealing.

Begin B: Alternating Update.

Update M: Softassign.

Update (d, w): Least-squares solution for TPS.

End B

Decrease T, λ_1 and λ_2.
End A

4 Experiments

We tested our algorithm on 2D synthetic and 3D real point-sets. The TPS-RPM algorithm is set up in the following manner. The initial value for T is set such that it is slightly bigger than the largest square distance of all point pairs to allow all possible matchings at first. It is gradually reduced by a factor of 0.93 (annealing rate). The correspondence variable M is initialized such that the inner $K \times N$ entries are all $\frac{1}{K}$ and the outlier row and column are equal to $\frac{1}{100K}$. We start our transformation variables d to be a unit matrix and w to be all zeros. The parameter λ_1^{init} is set to 1 and λ_2^{init} is set to $0.01\lambda_1^{init}$. The alternating update of correspondence M and transformation d, w is repeated 5 times which is usually sufficient for convergence after which we decrease the temperature T. The thin-plate spline kernels in 2D and 3D are

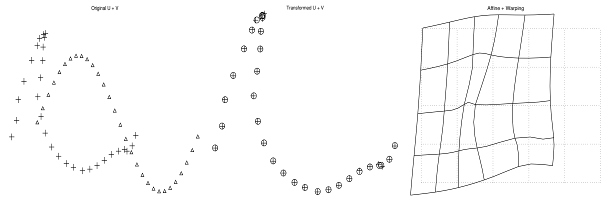

Figure 2: **Left:** Two point-sets V (triangles) and X (crosses). **Middle:** Final position V' (circles, transformed V) and X. **Right:** Deformation of the space is shown by comparing the original regular square grid (dotted lines) with its transformed version (solid lines).

not the same. In 3D, a different kernel of the form $\phi_a(v) = c||v - v_a||$ is used [22].

4.1 A 2D Example

We noticed that the algorithm demonstrates a very interesting scale-space like behavior. Such behavior can be seen from a simple 2D point matching example demonstrated here in Figures 2 and 3.

In the beginning, at a very high temperature T, the correspondence M is almost uniform. Then the estimated corresponding point-set $\{y_a = \sum_{i=1}^{N} m_{ai} x_i\}$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 first row in Figure 3.

With slightly lowered temperatures as shown in the 2nd and 3rd rows of Figure 3, the algorithm starts behaving like a principal axis method. 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 finds more localized correspondences which makes it possible to capture the more detailed structures within the target point-set. Progressively more refined matching is shown in the 4th, 5th and 6th rows. The last row shows that the algorithm 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 a coarse to fine approach. 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.

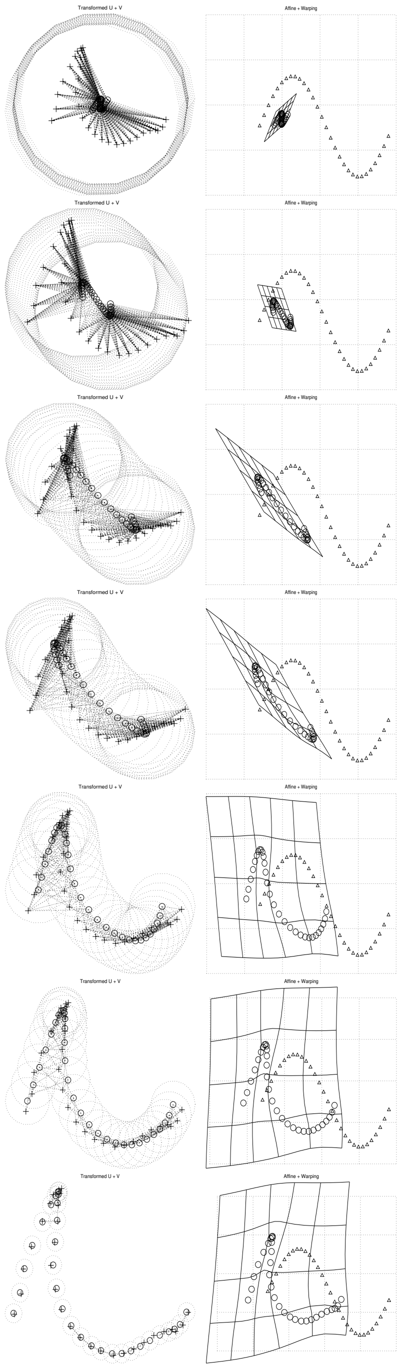

Figure 3: **Matching process.** Each row shows the state of the algorithm at a certain temperature. **Left:** The current correspondence between V' (circles, transformed V) and X (crosses). The most significant correspondences ($m_{ai} > \frac{1}{K}$) are shown as dotted links. A dotted circle of radius $\sqrt{T}$ is drawn around each point in V' to show the annealing process. **Right:** Deformation of the space: dotted regular grid with the solid deformed grid. Original V (triangles) and V' (circles, transformed V) are shown.

4.2 Synthetic Examples with Noise and Outliers

To provide a comparison, an improved (and symmetric) version of the iterated closest point (ICP) algorithm [2, 7] (with TPS) was implemented. It includes a dynamic thresholding mechanism. Distances between nearest point pairs are fit to a Gaussian distribution. Points with values larger than the mean plus 3 times the standard deviation are rejected as outliers. Since we have an annealing scheme for the regularization parameters λ_1 and λ_2, we also used it for ICP. We ran a lot of experiments on synthetic data with different degrees of warping, different amounts of noise and different amounts of outliers to test RPM and ICP's performance.

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. Instead of TPS, we use a different non-rigid mapping, namely Gaussian radial basis functions (RBF) [24] for the random transformation. The coefficients of the RBF 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 RBF coefficients 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 find the best TPS to map 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 RBF.

We conducted three series of experiments. In the first 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 on 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. 100 random experiments were repeated for each setting within each series.

The template that we chose comes from a Chinese character (blessing) which is a rather complex pattern. Some examples of these experiments are shown in Figure 4. The error means and standard deviations for each setting are shown in Figure 5. ICP's performance deteriorates much faster when the examples become harder due to any of three factors—degree of deformation, amount of noise or amount of outliers.

4.3 3D Sulcal Point Matching

An important problem in medical imaging is the registration of anatomical brain MRI, which is required for human brain mapping. One way to accomplish this is through the matching of sulcal feature points [4]. It is a difficult task because of the highly convoluted nature the human brain cortex. Non-rigid registration is necessary to bring different subjects' brain MRI into a common coordinate frame. We tested RPM on 5 MRI sulcal point-sets acquired from 3D MRI volumes [4]. RPM is used to solve for the affine as well as the TPS mappings using the point-sets. Overlays of all 5 subject sulcal point-sets before and after the matching are shown in Figure 6. Denser, tighter packing of the sulci indicates better registration.

5 Conclusions

The point matching problem becomes much harder when high dimensional non-rigid transformations and outliers are involved. From the experiments, we demonstrated that TPS-RPM performs quite well even when moderate to large numbers of outliers are present while other algorithms like ICP may no longer be suitable to solve such problems. Our algorithm is able to solve non-rigid point matching problems with up to 100 points in less than 1 minute on an Pentium$^{\mathrm{TM}}$II 400MHz Linux$^{\mathrm{TM}}$ machine (with the code implemented in MATLAB$^{\mathrm{TM}}$.) TPS-RPM is a fast and robust algorithm.

We are currently working on a new maximum *a posteriori* (MAP) statistical framework for point matching in order to be able to better understand the annealing, regularization and outlier rejection properties. We are also planning to compare and relate our algorithm with classical graph matching algorithms that attempt to solve the same problem.

Acknowledgements

This work was partially supported by an NSF grant to A.R. We thank Jim Rambo and Bob Schultz for

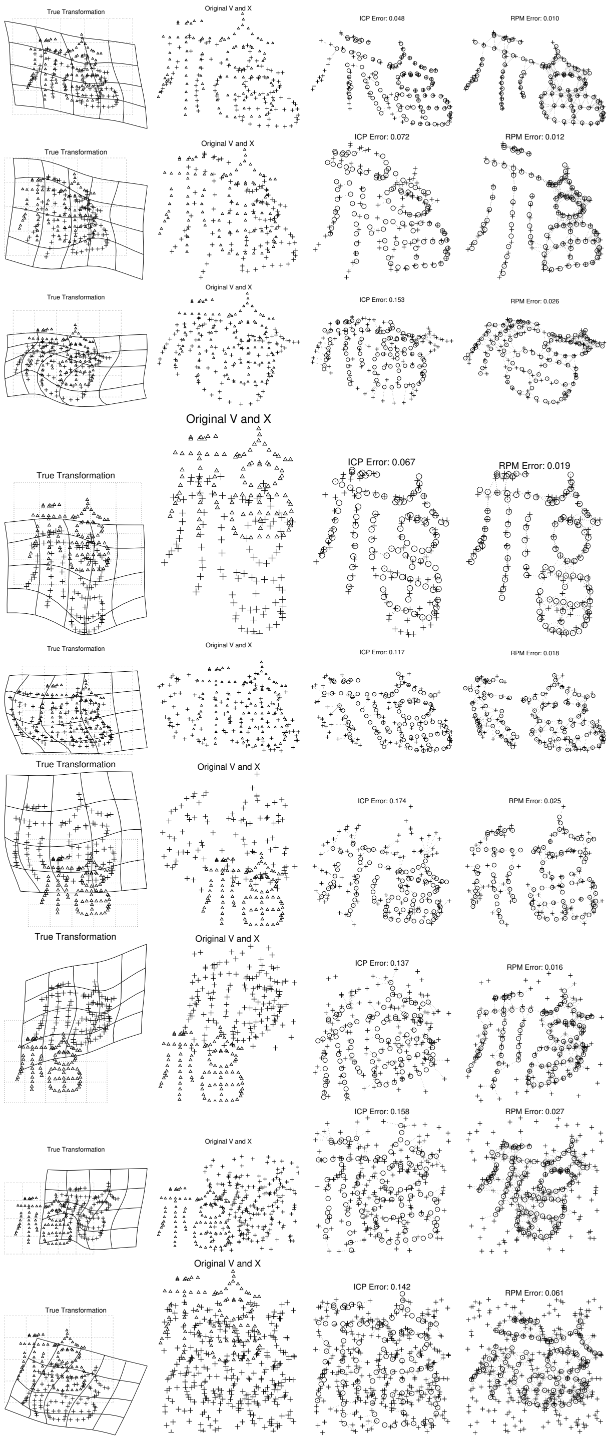

Figure 4: **Synthetic Experiment Examples.** Each row shows one example. Three examples from each of the three series are shown. Each row includes four parts arranged from left to right. **First:** Warped template. **Second:** Template and target (the warped template possibly with noise/outliers). **Third:** ICP result. **Fourth:** RPM result.

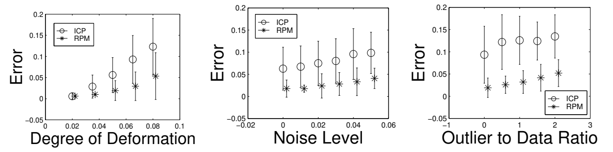

Figure 5: **Statistics of the Synthetic Experiments.** Note especially how ICP's errors increase much more faster than RPM's errors.

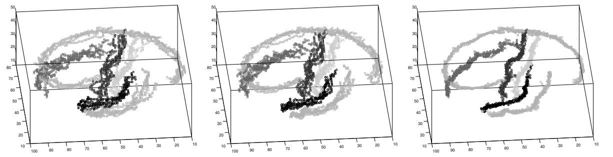

Figure 6: **Left:** Original position of the 5 sulcal point-sets overlayed. Only 6 sulci on one half of the brain are shown here. **Middle:** The overlay after affine RPM registration. **Right:** The overlay after TPS-RPM registration.

assistance with the anatomical sulcal brain MRI data and Hemant Tagare for helpful discussions.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[10] 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.

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

[12] 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.

[13] G. Lohmann and D. von Cramon. Automatic labelling of the human cortical surface using sulcal basins. *Medical Image Analysis.* (in press).

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

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

[16] 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.

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

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

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

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

[21] R. Szeliski and S. Lavallee. Matching 3D anatomical surfaces with non-rigid deformations using octree splines. *Intl. J. Computer Vision*, 18:171–186, 1996.

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

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

[24] 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.