LAMP-TR-117 December 2004
CAR-TR–1005
CS-TR-4633
UMIACS-TR-2004-75

ROBUST POINT MATCHING FOR NON-RIGID SHAPES: A RELAXATION LABELING BASED APPROACH

Yefeng Zheng
David Doermann

Language and Media Processing Laboratory
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742-3275
(zhengyf,doermann)@cfar.umd.edu

Abstract

Shape matching or image registration, which is often formulated as a point matching problem, is frequently encountered in image analysis, computer vision, and pattern recognition. Although the problem of registering rigid shapes was widely studied, non-rigid shape matching has recently received more and more attention. For non-rigid shapes, most neighboring points cannot move independently under deformation due to physical constraints. Therefore, though the absolute distance between two points may change significantly, the neighborhood of a point is well preserved in general. Based on this observation, we formulate point matching as a graph matching problem. Each point is a node in the graph, and two nodes are connected by an edge if their Euclidean distance is less than a threshold. The optimal match between two graphs is the one that maximizes the number of matched edges. The shape context distance is used to initialize the graph matching, and relaxation labeling (after enforcing one-to-one matching) is used to refine the matching results. Non-rigid deformation is overcome by bringing one shape closer to the other in each iteration using deformation parameters estimated from the current point correspondence. Experiments on real and synthesized data demonstrate the effectiveness of our approach: it outperforms shape context and TPS-RPM algorithms under non-rigid deformation and noise on a public data set.

Keywords: Point Matching, Shape Matching, Image Registration, Relaxation Labeling, Deformable Template, Thin Plate Spline (TPS)

The support of this research by the US Department of Defense under contract MDA-9040-2C-0406 is gratefully acknowledged.

1 Introduction

Shape matching or image registration is often encountered in image analysis, computer vision, and pattern recognition. A shape may be represented as a set of features at different levels, such as points, line segments, curves, or surfaces, and shape matching may be performed on these representations. The survey paper by Loncaric [1] covers the extraction and representation of a shape. Definitions of the distance between two features (i.e., point, lines, or curves) and their use in shape matching can be found in [2]. In general, the higher the level of a feature, the more difficult it is to extract reliably. The extraction of a set of point representation, for example, is easy, and it is more general since lines and surfaces can be discretized as a set of points. Although such discretization is by no means optimal, in many cases, reasonable matching results may be achieved [3]. Point matching, therefore, is often used in applications such as pose estimation [4, 5], medical image registration [6], surface registration [7, 8], object recognition [9], and handwriting recognition [10, 11]. In this paper, we focus on the point pattern based shape matching.

1.1 Overview of Our Approach

For non-rigid shapes, most neighboring points cannot move independently under deformation due to physical constraints. Such constraints may be represented as the ordering of points on a curve. Sebastian et al. [12] demonstrated the effectiveness of point ordering in matching curves, but for general shapes other than curves, local point ordering is hard to describe, and is ignored in many point matching algorithms. For example, in the shape context algorithm [9], neighboring points in one shape may be matched to two points far apart in the other shape. We observed that although the absolute distance between two points may change significantly under a non-rigid deformation, the neighborhood of a point is generally well preserved. As a primary contribution of this paper, we formulate point matching as a graph matching problem. Each point is a node in the graph, and two nodes are connected by an edge if their Euclidean distance is less than a threshold. The optimal match between two graphs is the one that maximizes the number of matched edges, so we explicitly search for an optimal match which preserves the point neighborhood best.

Graph matching is an NP-hard problem. Exhaustive or branch-and-bound search for a global optimal solution is only realistic for graphs with few nodes [13]. Local optimal search techniques are often used in real applications, whose performance depends on the initial solution. In this paper, we use the shape context distance to initialize the graph matching, followed by a relaxation labeling process to refine the match. A difference to the previous applications of relaxation labeling [14, 15] is that we use it to solve a constrained optimization problem. The relaxation labeling process is guaranteed to converge to a local optimal solution [16]. In the previous work, it is used in an ad hoc way without an objective function to be optimized, so there is no guarantee about the quality of the solution. Furthermore, unlike the previous work where many-to-one is allowed, we enforce one-to-one matching in our approach.

There are two unknown variables in a shape matching problem: the correspondence and the transformation [17]. Since solving for either without information regarding the

other is quite difficult, most approaches to non-rigid shape matching use iterated approaches. Given an estimate of the correspondence, the transformation may be estimated and used to update the correspondence. If these two steps are well designed, such an iterated process will converge and improve the initial result. In this paper, the common framework of iterated correspondence and transformation estimation is used [9]. In the first iteration, the affine transformation between two shapes is estimated and corrected. A robust Least Median Squares (LMS) estimator is exploited to estimate the affine transformation. In the following iterations, Thin Plate Spline (TPS) is used to bring a shape closer to the other based on the current point matching results.

Experiments on real and synthesized data demonstrate the effectiveness of our approach. It is more robust under deformation and noise than two state-of-the-art algorithms, the shape context [9] and TPS-RPM algorithms [17], on a public data set.

1.2 Previous Work

Shapes can be roughly categorized as rigid or non-rigid, and the realization of a shape may undergo various deformations in captured images. With small number of transformation parameters (six for a 2-D affine transformation), rigid shape matching is relatively easy. Rigid shape matching under the affine [3, 7] or projective transformation [5] has been widely studied with an extensive literature. Since it is impossible to cover them well in this section without omitting many excellent works, the reader is referred to other survey papers for a large bibliography [18,19]. Although many point matching algorithms developed for rigid shapes can tolerate some degree of noise or local distortions, large free-form deformation is a significant challenge. Recently, point matching for non-rigid shapes has received more and more attention. In the following literature review, we will focus on publications on non-rigid shape matching.

Point matching for non-rigid shapes is hard because both linear distortions (i.e., translation, rotation, scale change, and shear) and non-linear distortions must be compensated for. Therefore, the common framework of iterated correspondence and transformation estimation is widely used. The Iterated Closest Point (ICP) algorithm, a well-known heuristic approach proposed by Besl and McKay, is one example [3, 7]. Assuming two shapes are roughly aligned, for each point in one shape, the closest point in the other shape is taken as the current estimate of the correspondence. The affine transformation estimated with the current correspondence will then bring two shapes closer. ICP was later extended for non-rigid free-form surfaces [8]. The framework consists of three stages. First, the rigid displacement is estimated using surface curvatures. Second, the global affine transformation is estimated using the ICP algorithm. Third, a local affine transformation (LAT) is attached to each point to locally deform the surface. LAT was also used by Wakahara [10] to match and recognize handwritten characters. A dynamic window with a gradually decreasing size is used to estimate the local affine transformation for a point. This approach was improved by combining global and local affine transformations to increase the robustness [11].

Although LAT is flexible enough to model local non-rigid deformation, there is no standard way to define the neighborhood window size to estimate the parameters of LAT. How to combine the global and local affine transformations is an open problem as

well, so more flexible deformation models with closed-form representations are desired. In the literature on interpolation and approximation, radial basis functions (RBF) with different kernel functions, such as Thin Plate Spline (TPS) [20] and Gaussian RBF [21], are widely used. Recently, the TPS deformation model began to be applied in point matching [9, 17], because it can be formulated as an optimal solution of the bending of a thin plate [20]. Chui and Rangarajan proposed an optimization based approach – the TPS-RPM algorithm [17]. The bending energy of the TPS model and the average Euclidean distance between two point sets are combined in an objective function. The softassign technique and deterministic annealing algorithm are used to search for the optimal solution, which significantly outperforms the ICP algorithm on non-rigid point matching. Belongie et al. [9] proposed another method for non-rigid point matching. In this approach, a shape context is assigned to a given point, which describes the relative distribution of remaining points. After defining the similarity between two points based on their shape contexts, the Hungarian algorithm [22] is used to search for the optimal match between the two point sets. Similarly, the TPS model is used to bring two shapes closer in each iteration.

More recently, Glaunes et al. [23] proposed another point matching approach. Taking a point set as a sampling of the underlining distribution, they proposed a theory based on diffeomorphisms on distributions. Their formulation reduces to an optimization problem with a weighted summation of two parts: the energy associated with the deformation and the distance between two point sets under this deformation. This is similar to the objective function in [17], although no explicit deformation model is assumed. Instead, a variational method is used to search for the optimal deformation. Experimental results on synthesized data are encouraging, but more extensive tests should be performed to show the effectiveness of their approach.

Another interesting work is the matching of articulated objects [4]. An articulated object (such as a person) is composed with several rigid segments connected by pivot points. The deformation of rigid segments can be modeled with an affine transformation. A global hierarchical search strategy is used to search for and match pivot points, and local matching of rigid segments is used to prune the search tree, thus reducing the computational cost [4].

The remainder of this paper is organized as follows. In Section 2, we formulate point matching as a graph matching problem. Section 3 describes our relaxation labeling based graph matching approach. Shape deformation models, such as the affine transformation and TPS, are discussed in Section 4, followed by a brief summary of our approach in Section 5. We demonstrate the robustness of our approach with experiments in Section 6, and the paper concludes with a discussion of the future work in Section 7.

2 Formulation as a Graph Matching Problem

In this section, we formulate point matching as a graph matching problem. Suppose a template shape T is composed with M points, $S_T = \{T_1, T_2, \cdots, T_M\}$, and a deformed shape D is composed with N points, $S_D = \{D_1, D_2, \cdots, D_N\}$. We want to find a matching function $f : S_T \Leftrightarrow S_D$ between these two point sets, which is optimal for some metric. In many applications, one-to-one matching is desired, but in general, the number of points

in T and D may be different. Even if two shapes have the same number of points, not all points in one shape will have correspondence in the other shape. To solve this problem, the concept of a dummy or *nil* point, is introduced. The point sets S_T and S_D are augmented to $S'_T = \{T_1, T_2, \cdots, T_M, nil\}$ and $S'_D = \{D_1, D_2, \cdots, D_N, nil\}$ respectively. A match between shapes T and D is $f : S'_T \Leftrightarrow S'_D$, where the match of normal points is one-to-one, but multiple points may be matched to a dummy point.

Under a rigid transformation (i.e., translation and rotation), the distance between any point pair is preserved. Therefore, the optimal match $\hat{f}$ is

$$\hat{f} = arg\min_f C(T, D, f) \tag{1}$$

where

$$\begin{aligned} C(T, D, f) &= \sum_{m=1}^{M}\sum_{i=1}^{M}\Big(\|T_m - T_i\| - \|D_{f(m)} - D_{f(i)}\|\Big)^2 + \\ &\quad \sum_{n=1}^{N}\sum_{j=1}^{N}\Big(\|D_n - D_j\| - \|T_{f^{-1}(n)} - T_{f^{-1}(j)}\|\Big)^2 \end{aligned} \tag{2}$$

If $M = N$ and no points are matched to dummy points, the summation of the first term in (2) should be equal to the summation of the second term. Points, which are matched to a dummy point, need special treatment in the above cost function. To make the representation simple, we do not explicitly describe such treatment here. We will come back to this problem later.

If non-rigid deformation is present, the distance between a pair of points cannot be preserved. This is especially true for two points which are far apart. In many situations, however, the local neighborhood of a point may not change freely due to physical constraints. Therefore, we define a neighborhood for point m as N_m. The neighborhood relationship is symmetric, which means if $i \in N_m$ then $m \in N_i$. Then, (2) can be modified as

$$\begin{aligned} C(T, D, f) &= \sum_{m=1}^{M}\sum_{i\in N_m}\Big(\|T_m - T_i\| - \|D_{f(m)} - D_{f(i)}\|\Big)^2 + \\ &\quad \sum_{n=1}^{N}\sum_{j\in N_n}\Big(\|D_n - D_j\| - \|T_{f^{-1}(n)} - T_{f^{-1}(j)}\|\Big)^2 \end{aligned} \tag{3}$$

The absolute distance of a pair of points is not preserved well under scale changes. Therefore, we quantize the distance to two levels as

$$\|T_m - T_i\| = \begin{cases} 0 & i \in N_m \\ 1 & i \notin N_m \end{cases} \quad \text{and} \quad \|D_n - D_j\| = \begin{cases} 0 & j \in N_n \\ 1 & j \notin N_n \end{cases} \tag{4}$$

Then, (3) is simplified as

$$C(T, D, f) = \sum_{m=1}^{M}\sum_{i\in N_m} d(f(m), f(i)) + \sum_{n=1}^{N}\sum_{j\in N_n} d(f^{-1}(n), f^{-1}(j)) \tag{5}$$

where

$$d(m,i) = \begin{cases} 0 & i \in N_m \\ 1 & i \notin N_m \end{cases} \tag{6}$$

In order to deal with the case that a point may be matched to a dummy point, we let

$$d(.,nil) = d(nil,.) = d(nil,nil) = 1 \tag{7}$$

to discourage matching to dummy points.

In the following, we rewrite the objective function of (5), and interpret it as a graph matching problem. First, we subtract a constant term from $C(T,D,f)$.

$$\begin{aligned} C'(T,D,f) &= C(T,D,f) - \sum_{m=1}^{M} \sum_{i \in N_m} 1 - \sum_{n=1}^{N} \sum_{j \in N_n} 1 \\ &= \sum_{m=1}^{M} \sum_{i \in N_m} [d(f(m),f(i)) - 1] + \sum_{n=1}^{N} \sum_{j \in N_n} [d(f^{-1}(n),f^{-1}(j)) - 1] \\ &= -\sum_{m=1}^{M} \sum_{i \in N_m} \delta(f(m),f(i)) - \sum_{n=1}^{N} \sum_{j \in N_n} \delta(f^{-1}(n),f^{-1}(j)) \end{aligned} \tag{8}$$

where

$$\delta(i,j) = 1 - d(i,j) \tag{9}$$

Minimizing $C(T,D,f)$ is equivalent to minimizing $C'(T,D,f)$ since the difference between them is a constant. Therefore, the minimization problem of (1) is equivalent to the following maximization problem.

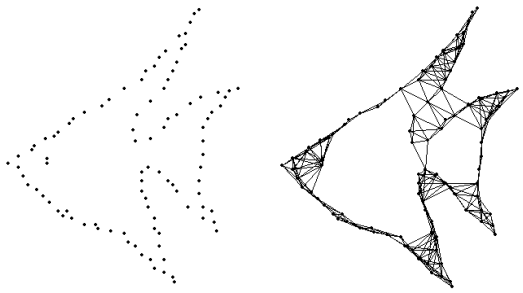
$$\hat{f} = arg\max_{f} S(T,D,f) \tag{10}$$

where

$$S(T,D,f) = \sum_{m=1}^{M} \sum_{i \in N_m} \delta(f(m),f(i)) + \sum_{n=1}^{N} \sum_{j \in N_n} \delta(f^{-1}(n),f^{-1}(j)) \tag{11}$$

This formulation can be interpreted as a graph matching problem. We can represent a point set as a graph, where each point is a node in the graph and two nodes are connected by an edge if they are neighbors. The dummy node is not connected to other nodes in the graph. If connected nodes m and i in one graph are matched to connected nodes $f(m)$ and $f(i)$ in the other graph, $\delta(f(m),f(i)) = 1$. Therefore, the optimal solution of (10) is the one which maximizes the number of matched edges of two graphs.

Our definition of neighborhood is as follows. Initially, the graph is fully connected, and we then remove long edges until a pre-defined number of edges are preserved. Suppose, there are M nodes in the graph, the number of preserved edges is $M \times E_{ave}$, where E_{ave} is a parameter in the range from five to nine in our experiments. With this neighborhood definition, the graph representation of a point set is translation, rotation, and scale change invariant. Fig. 1 shows a graph representation of a point set with $E_{ave} = 7$. We expect points connected with an edge move together under deformation, so the structure of the graph is preserved.



(a) (b)

Figure 1: A point set (a) and its graph representation (b).

Graph matching (or more generally, attributed relational graph matching) is used in [24] and [25] to match road maps extracted from aerial photographs. Their graph definition is different from ours, where road intersections are nodes in the graph and two nodes are connected by an edge if there is a road between two intersections. Such a graph definition is natural for a road map, but errors in road detection will change the graph structure. In our case, given an arbitrary set of points, there is no such natural definition of connections among points. Graph matching is widely used in many fields, such as computer vision and pattern recognition. There are various kinds of graph structures, and many different metrics are available to evaluate a match between two graphs in the literature [26]. Our graph representation and the corresponding matching metric are derived from the observation (or assumption) that non-rigid deformation will not change the neighborhood of a point significantly.

3 Relaxation Labeling for Graph Matching

As previously stated, graph matching is an NP-hard problem. Global optimal approaches, such as exhaustive or branch-and-bound search, are only applicable to graphs of a small size (for example, less than 20 nodes) [13]. Many local optimal graph matching algorithms have been proposed in the literature. Among them, the relaxation labeling technique is widely used [24,27]. Since it converges to a local optimal solution depending on the initial estimate, a good initialization is crucial to achieve a good result. In this paper, we use the shape context distance to initialize the matching of two graphs.

3.1 Matching Probability Matrix

We can represent the matching function f in (10) with a set of supplemental variables, which are organized as a matrix P with dimension $(M+1) \times (N+1)$.

$$P = \left[\begin{array}{ccc|c} p_{11} & \cdots & p_{1N} & p_{1,nil} \\ \vdots & & \vdots & \vdots \\ p_{M1} & \cdots & p_{MN} & p_{M,nil} \\ \hline p_{nil,1} & \cdots & p_{nil,N} & 0 \end{array}\right] \tag{12}$$

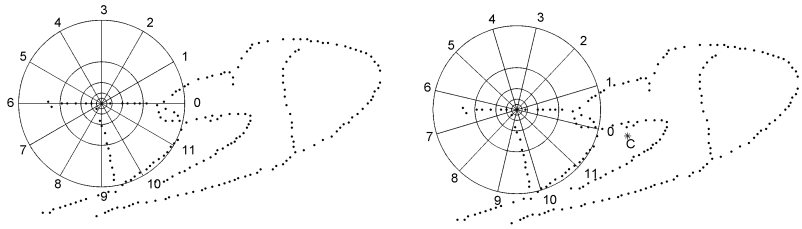


Figure 2: Shape context of a point. (a) Basic shape context. (b) Our rotation invariant shape context. The point labeled with * is the mass center of the point set.

If point T_m in the template shape T is matched to point D_n in the deformed shape D, then $P_{mn} = 1$, otherwise $P_{mn} = 0$. The last row and column of P represent the case that a point may be matched to a dummy point. Matrix P satisfies the following normalization conditions

$$\sum_{n=1}^{N+1} P_{mn} = 1 \quad \text{for } m = 1, 2, \cdots, M \tag{13}$$

$$\sum_{m=1}^{M+1} P_{mn} = 1 \quad \text{for } n = 1, 2, \cdots, N \tag{14}$$

Using matrix P, the objective function of (11) can be written as

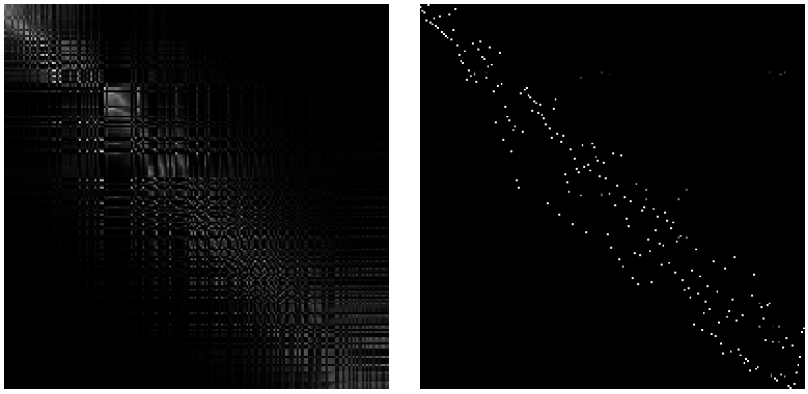
$$S(T, D, P) = 2 \sum_{m=1}^{M} \sum_{i \in N_m} \sum_{n=1}^{N} \sum_{j \in N_n} P_{mn} P_{ij} \tag{15}$$

Since $P_{mn} \in \{0, 1\}$, searching for an optimal P which maximizes $S(T, D, P)$ is a hard discrete combinatorial problem. In this paper, we use relaxation labeling to solve the optimization problem, where the condition $P_{mn} \in \{0, 1\}$ is relaxed as $P_{mn} \in [0, 1]$ [27]. After relaxation, P_{mn} is a real number, and the problem is converted to a constrained optimization problem with continuous variables.

3.2 Initialization with Shape Context Distance

The performance of relaxation labeling depends heavily on the initial value of the matching probability matrix P. We need a good initial measure of the matching probabilities. One option is to assign an attribute, such as the color or intensity gradients of the pixel, to a point if it is extracted as a pixel in an image [28]. We can then compute the similarity between a pair of points, and convert it to a measure of the matching probability. If a set of points is given without any additional information, the shape context provides an effective way to compute the similarity between two points [9]. In this paper, we use the shape context distance to initialize the point matching probabilities. If other attributes of a point are available, they can be easily incorporated into our framework.

To extract the shape context of a point, an array of bins is put around the point, as shown in Fig. 2a. The number of points inside each bin is calculated as the context of



(a) (b)

Figure 3: Point matching probability matrix P. The matching probabilities to a dummy point are not shown. (a) Initial probabilities using the shape context distance. (b) After 300 iterations of relaxation labeling updates.

this point. Therefore, the shape context of a point is a measure of the distribution of other points relative to it. Bins which are uniform in log-polar space are used to make the descriptor more sensitive to positions of nearby points than to those of points far away. Five bins for the radius and 12 bins for the rotation angle are used throughout our experiments. Consider two points, m in one shape, and n in the other shape. Their shape contexts are $h_m(k)$ and $h_n(k)$, for $k = 1, 2, \ldots, K$, respectively. Let C_{mn} denote that cost of matching these two points. As shape contexts are distributions represented as histograms, it is natural to use the χ^2 test statistic [9]

$$C_{mn} = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_m(k) - h_n(k)]^2}{h_m(k) + h_n(k)} \tag{16}$$

The Gibbs distribution is widely used in statistical physics and image analysis to relate the energy of a state to its probability [29]. Taking the cost C_{mn} as the energy of the state that points m and n are matched, the probability of the match is

$$P_{mn} \propto e^{-C_{mn}/T_{init}} \tag{17}$$

Parameter T_{init} is used to adjust the reliability of the initial probability measures, where $T_{init} \in [0.05, 0.1]$ is appropriate according to our experiments. We set the probability for a point matching to a dummy point, $P_{m,nil}$ or $P_{nil,n}$, to 0.2. Experiments show that our approach is not sensitive to this parameter. Fig. 3a shows the initial matching probability matrix P of two shapes.

It is obvious that the shape context is translation invariant. Using bin arrays with an adaptive size according to the mean point distance of a shape, the shape context is scale change invariant too [9], but it is sensitive to large rotations. In some applications, rotation invariance is required. Our graph representation is rotation invariant, so we need a rotation invariant initialization scheme. A complete rotation invariant shape context was proposed using the tangent direction at each point as the positive x-axis for the local coordinate system [9]. One drawback of this approach is that the tangent direction,

defined for gray-scale images, is not applicable for binary images. Furthermore, if only the point set is given without accessing the original image, we cannot estimate the tangent direction. Another drawback is that as a first-order derivative operation, the estimate of the tangent direction is sensitive to noise. Instead, in this paper, we use the mass center of a point set as a reference point, and use the direction from a point to the mass center as the positive x-axis for the local coordinate system. Our rotation invariant shape context is shown in Fig. 2b. If there is zero mean white noise in point position measurements, after averaging, the effect of noise to the mass center is reduced. Therefore, our approach is more robust than the tangent direction based approach under noise.

3.3 Relaxation Labeling

Relaxation labeling was first proposed in a seminal paper by Rosenfeld, Hummel, and Zucker in mid-1970s [27]. The basic idea is to use iterated local context updates to achieve a globally consistent result. Their updating rule is [1]

$$P_{mn} := \frac{P_{mn} S_{mn}}{\sum_{j=1}^{N} P_{mj} S_{mj}} \tag{18}$$

where S_{mn} is a support function of the match between points T_m and D_n. It represents how much support the current match gets from its neighbors. The denominator is used to enforce one normalization constraint.

In the original paper, S_{mn} is defined heuristically, although with ad hoc heuristic arguments, a variety papers later reported on the practical usefulness of the algorithm (see [30] for a review and an extensive bibliography). The success in real applications and the heuristic flavor of the algorithm motivated investigators to establish a theoretic foundation. There are two different approaches. Some have tried to set the labeling problem within a probabilistic framework using Bayesian analysis [24, 31]. The Bayesian theory, however, can only explain one iteration of the relaxation process. An alternative explicitly defines some quantitative measure of consistency to be maximized, and formulates the labeling problem as one of optimization [32, 33]. Projected gradient methods are often used to optimize the objective function. In these theories, the support function S_{mn} is defined as the derivative of the objective function with respect to P_{mn} [33]. The updating rule of the projected gradient methods is

$$P := P + \gamma Q(S) \tag{19}$$

where γ is the updating step and S is a matrix composed with elements S_{mn}. $Q(S)$ is a projection operation of S to limit the range of P_{mn} to $[0, 1]$ and enforce normalization constraints. In the case of boundary points (i.e., having at least one component of the probabilities equal to zero or one) the projection operation is much more complicated and the procedure becomes computationally expensive. Furthermore, the updating step γ is difficult to tune. An increase in the objective function is guaranteed only when

[1] In the original paper, the support function S is defined in a heuristic way in the range of $[-1, 1]$. In order to satisfy $P \geq 0$ after updating, $1 + S$ is used to substitute S in both the numerator and denominator in the updating rule [27].

infinitesimal steps are taken, and searching for the optimal step size in each iteration is computationally expensive. Recently, Pelillo [16] showed that the original updating rule in (18) does converge to a local minimum if (a) the objective function is a polynomial with nonnegative coefficients, and (b) S_{mn} is defined as a gradient of the objective function. The advantages of this updating rule, compared with the projected gradient methods, are (a) computationally expensive projection operations are avoided, and (b) it is parameter free. We tried several updating rules compared in [34] and found that the updating formula of (18) is robust and achieves better results. With our objective function of (15), S_{mn} takes the form of

$$S_{mn} = 4 \sum_{i \in N_m} \sum_{j \in N_n} P_{ij} \tag{20}$$

Since $S_{mn} \geq 0$, the constraint that $P_{mn} \in [0, 1]$ is satisfied after normalization.

In previous applications of the relaxation labeling technique, many-to-one match is allowed [14, 15, 35–37]. Only one-way normalization constraint, either (13) or (14), is enforced. Unfortunately, in many applications, one-to-one match is desired. Projected gradient methods may be modified to enforce one-to-one match. The projection operation, however, is computationally expensive and it is unclear how to find a projection satisfying two-way normalization constraints. In this paper, a different approach based on alternated row and column normalizations of the matching probability matrix P is used to enforce one-to-one match [17]. A nonnegative square matrix with each row and column summing to one is called a doubly stochastic matrix. Sinkhorn showed that the iterative process of alternated row and column normalizations will convert a matrix with positive elements to a doubly stochastic matrix [38]. The conclusion can be extended to a positive non-square matrix. We call a matrix where each row and column (except the last row and column) sums one a generalized doubly stochastic matrix. We can show that alternated row and column normalizations (except the last row and column) of a positive matrix will result in a generalized doubly stochastic matrix (refer to the Appendix for a proof). This technique is also used in the softassign point matching approach without proof [17].

Fig. 3a shows the initial value of the point matching probability matrix P of two shapes. After each relaxation labeling update, we perform alternated row and column normalizations to matrix P. Generally, a few rounds are enough to bring a matrix close to a generalized doubly stochastic matrix. After 300 iterations of relaxation labeling updates, the ambiguity of matches decreases. As shown in Fig. 3b, most elements of the matrix converge to zero or one.

After relaxation labeling updates, points with maximum matching probability less than P_{min} ($P_{min} = 0.95$) are labeled as outliers by matching them to dummy points. The matched point pairs are used to estimate the parameters of the affine or TPS deformation model, and the estimated parameters are used to transform the template shape to bring it closer to the deformed shape. In some application scenarios (e.g., the experiments in Section 6.1), we may want to find as many matches as possible. Unfortunately, the ratio of points matched to dummy points by the relaxation labeling updates cannot be controlled directly. After several iterations of correspondence and transformation estimations, two point sets may be close to each other. Therefore, in the last round, we find the optimal one-to-one match by minimizing the summation of Euclidean distances

from the transformed template shape to the deformed shape.

$$D = \sum_{m=1}^{M} \|T_m^* - D_{f(m)}\| \tag{21}$$

where T_m^* is a point from the template shape after transformation. The optimal match $\hat{f}$ can be found using the Hungarian algorithm [22].

3.4 Relationship to Previous Work

The relaxation labeling technique has been used for shape matching [14,15,35–37], shortly after it was proposed. Among them, those works on point matching [14, 15] are closely related to our approach. Several works on relaxation labeling based graph matching also appear in the literature [24, 25, 39]. A difference to the previous applications of relaxation labeling [14, 15] is that we use it to solve a constrained optimization problem. With support function defined as the derivative of the objective function, the relaxation labeling process is guaranteed to converge to a local optimal solution [16, 33]. In the previous work, relaxation labeling is used in an ad hoc way without an objective function to be optimized, so there is no guarantee about the quality of the solution. It was found in experiments that, after some point, further iterations of relaxation labeling updates may deteriorate the performance [34]. The second difference is that one-to-one match is enforced by alternated row and column normalizations of the matching probability matrix. Therefore, unlike the previous work where only one-way normalization was enforced, two-way normalization constraints are satisfied in our approach.

The relaxation labeling method used in this paper is similar to the well-known softassign technique [5, 17, 40]. Both of them convert the discrete combinatorial optimization problem to one with continuous variables by assigning a probability measure to a match. The procedure is called "relaxation" or "soft" in these two techniques respectively. In softassign, however, in order to achieve a firm or unambiguous solution (with matching probabilities be zero or one), a penalty term is added in the objective function to encourage an unambiguous solution. An appropriate weight of the penalty term is necessary to achieve good results [17]. On the other hand, for relaxation labeling, it has been shown that each unambiguous consistent solution is a fixed point. The relaxation labeling process will converge to it, starting from a nearby point [16,33]. Although there is no guarantee that the relaxation labeling process will converge to an unambiguous solution starting from an arbitrary initialization, our experiments show that most elements of matrix P do converge to zero or one. Therefore, a penalty term is unnecessary for our objective function.

4 Shape Deformation Models

It is difficult to achieve a good match for shapes under both rigid and non-rigid distortions with a single approach. The strategy of iterated point correspondence and transformation estimations is widely used for non-rigid shape matching. In our approach, for the first iteration, the affine transformation between two shapes is estimated and corrected.

Instead of using the Least Squares (LS) estimator to estimate parameters of the affine transformation [9], a more robust Least Median Squares (LMS) estimator is used. In the following iterations, the Thin Plate Spline (TPS) deformation model is exploited to bring two shapes closer. Our approach is similar to [9] except that a more robust LMS estimator is used to estimate the affine transformation, instead of the LS estimator.

4.1 Affine Transformation Estimation Based on LMS

The LS estimator is widely used to estimate transformation parameters. Suppose point (x_i, y_i) is matched to point (u_i, v_i), for $i = 1, 2, \cdots, n$, the optimal parameters of the affine transformation are those which minimize the summation of squares of the regression errors.

$$\hat{A}, \hat{T} = arg\min_{A,T} \sum_{i=1}^{n} \left\| \begin{pmatrix} u_i \\ v_i \end{pmatrix} - A \begin{pmatrix} x_i \\ y_i \end{pmatrix} - T \right\|^2 \tag{22}$$

where A is a 2×2 matrix representing the rotation and anisotropic scale changes, and T is a translation vector. One advantage of the LS estimator is that closed-form solutions are available [41]. It is, however, sensitive to outliers in the matching [42]. The breakdown point is often used to evaluate robustness of an estimator under outliers, which is defined as the smallest proportion of observations that must be replaced by arbitrary values in order to force the estimator to produce values arbitrarily far from the true values [43]. The breakdown point of the LS estimator is 0%. Furthermore, it is generally difficult to detect outliers based on the regression residual errors since they may spread over all of the points [42].

In general, the results of the first iteration of point matching may be noisy with many errors, so a more robust estimator is required. Several robust regression methods have been proposed in the statistics literature. Among them, the Least Median Squares (LMS) estimator achieves the highest possible break down point, 50% [42]. Instead of minimizing the summation of squares of regression errors, the LMS estimator minimizes the median of the regression errors.

$$\hat{A}, \hat{T} = arg\min_{A,T} \text{ median} \left\{ \left\| \begin{pmatrix} u_i \\ v_i \end{pmatrix} - A \begin{pmatrix} x_i \\ y_i \end{pmatrix} - T \right\|^2 \quad \text{for } i = 1, 2, \cdots, n \right\} \tag{23}$$

There are no closed-form solutions for (23). Normally, we randomly select a subset with three matched pairs, which can determine an affine transformation, and using the estimated parameters, we can calculate the median of the regression errors. Iterating the random selection procedure, the optimal solution of (23) can be achieved. Suppose, there are n matched pairs and about 50% of them are wrong (matching outliers). In the worse case, we must select at least $\begin{pmatrix} n \\ 3 \end{pmatrix} - \begin{pmatrix} n/2 \\ 3 \end{pmatrix} + 1$ different subsets to ensure at least one subset without outliers is selected. This is too pessimistic. In real applications, we only need to exam a small number of subsets. After examining k subsets, the probability of having at least one good subset is $1 - \left[1 - \begin{pmatrix} n/2 \\ 3 \end{pmatrix} / \begin{pmatrix} n \\ 3 \end{pmatrix}\right]^k$ (assuming sampling with replacement). For example, let $n = 200$, the probability of getting at least one good

subset in 50 random selections is 99.8%. The LMS estimator can be used to estimate the affine transformation without knowing the correspondence between two point sets [44]. Without rough correspondence, however, a large number of subsets need to be examined.

4.2 TPS Deformation Model

The TPS model is often used for representing flexible coordinate transformations, because it is parameter free with a physical explanation and closed-form representations [20]. It has been used in non-rigid shape matching in [9] and [17]. Suppose z_i is the target function value at location (x_i, y_i), for $i = 1, 2, \cdots, n$. Two TPS models are used for the 2-D coordinate transformation. Suppose point (x_i, y_i) is matched to (u_i, v_i), we set z_i equal to u_i and v_i in turn to obtain one continuous transformation for each coordinate. The TPS interpolant $f(x, y)$ minimizes the bending energy

$$I_f = \int\int_{\mathcal{R}^2} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 dxdy \tag{24}$$

and has the solution of the form

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{n} w_i U(\|(x_i, y_i) - (x, y)\|) \tag{25}$$

where $U(r)$ is the kernel function, taking the form of $U(r) = r^2 log r^2$. The parameters of the TPS models w and a are the solution of the following linear equation

$$\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} z \\ 0 \end{bmatrix} \tag{26}$$

where $K_{ij} = U(\|(x_i, y_i) - (x_j, y_j)\|)$, the ith row of P is $(1, x_i, y_i)$, w and z are column vectors formed from w_i and z_i respectively, and a is the column vector with elements a_1, a_x, a_y.

If there are errors in the matching results, we use regularization to trade off between exact interpolation and minimizing the bending energy as follows.

$$H_f = \sum_{i=1}^{n} [z_i - f(x_i, y_i)]^2 + \lambda I_f \tag{27}$$

where λ is the regularization parameter, controlling the amount of smoothing. The regularized TPS can be solved by replacing K in (26) with $K + \lambda I$, where I is the $n \times n$ identity matrix [45, 46]. We set $\lambda = 1$ in the following experiments.

5 Summary of Our Approach

Following is a brief summary of our approach.

Input: Two point sets, $T_1, T_2, \ldots, T_M$ from the template shape T, and $D_1, D_2, \ldots, D_N$ from the deformed shape D.
Output: The correspondence between two point sets.

1. Set the transformed template shape T^* as T.
2. Set iteration number to one.
3. Calculate the shape context for each point in T^* and D, and use (16) to calculate the distance between each point pair T_m^* and D_n.
4. Use (17) to initialize the matching probability matrix P, and convert it to a generalized doubly stochastic matrix by alternated row and column normalizations.
5. Use (18) to update the matching probability matrix R ($R = 300$) times. After each update, convert matrix P to a generalized doubly stochastic matrix.
6. If the iteration number is one, use LMS to estimate the affine transformation between T and D.
7. Otherwise, use (26) to estimate parameters of the TPS deformation model between T and D.
8. Transform template point set T to T^* using the estimated deformation parameters.
9. Increase the iteration number by one. If the iteration number is less than I_{max} ($I_{max} = 10$), go to step 3.

Suppose both shapes have N points, the computation cost of shape context distances is in the order of $O(N^2)$. Relaxation labeling updates will take $O(N^2)$ time. The computational complexity of the algorithm may be largely dependent on the implementation of the spline deformation, which can be $O(N^3)$ in the worst case. With our un-optimized C++ implementation, matching two shapes (each with 100 points) takes about 1.6 seconds on a PC with a 2.8GHZ CPU.

6 Experiments

6.1 Basic Examples

First, we test our algorithm on the test samples used in [17], and compare our results with two other algorithms: the shape context [9] and TPS-RPM algorithms [17]. The TPS-RPM algorithm and our relaxation labeling based approach may reject some points as outliers by matching them to a dummy point. There are no parameters available in either algorithm to adjust the ratio of rejected points explicitly. In these examples, the template and deformed shapes have the same number of points. In order to achieve a direct and fair comparison, we prefer to match as many point pairs as possible without rejection. The shape context algorithm can achieve this by setting the outlier ratio to zero. For the other two algorithms, after point matching and transformation are complete, we use the approach discussed in Section 3.3 to minimize the summation of Euclidean distances between the transformed template point set and the deformed point set (see, Eq. (21)).

Fig. 4 shows the point matching results of three algorithms on a pair of curves and two pairs of closed contours. As shown in the left column, all algorithms achieved

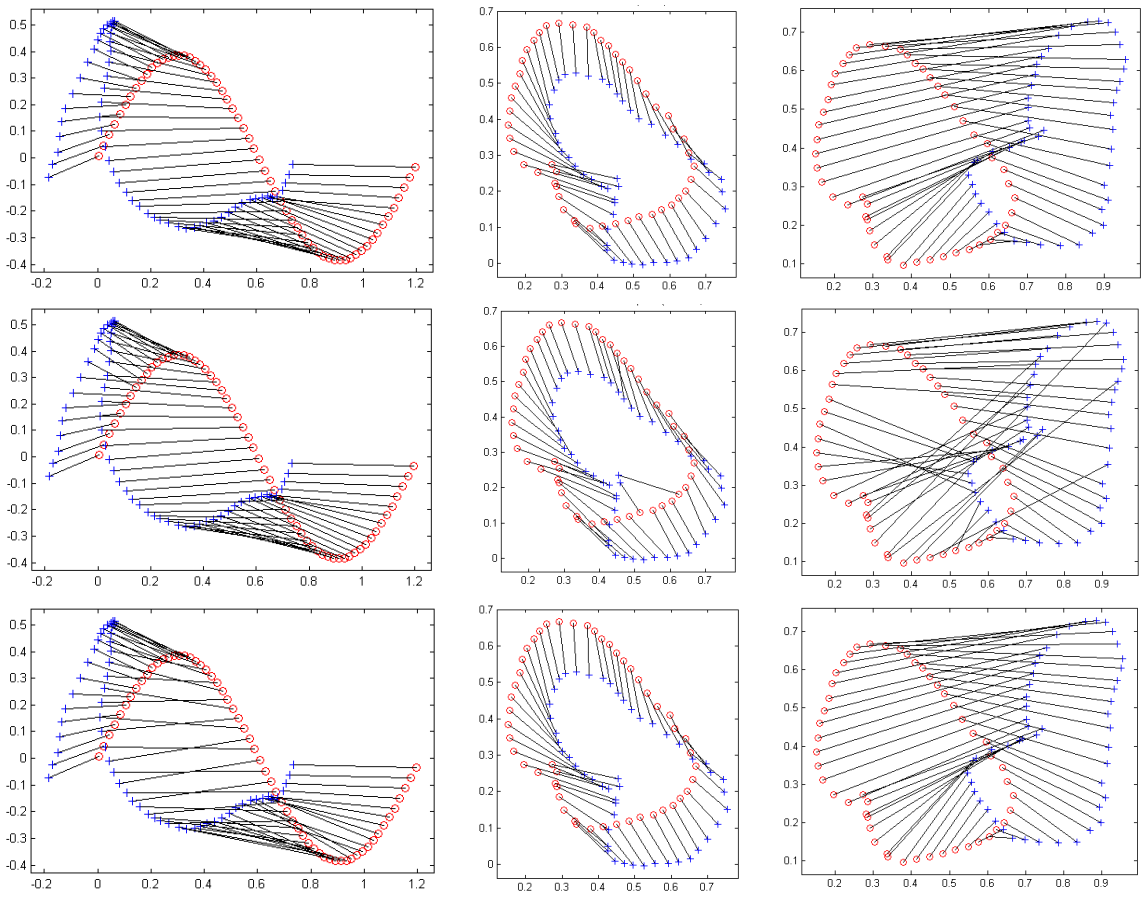


Figure 4: Point matching results on several test samples. Top row: our approach. Middle row: the shape context algorithm. Bottom row: the TPS-RPM algorithm.

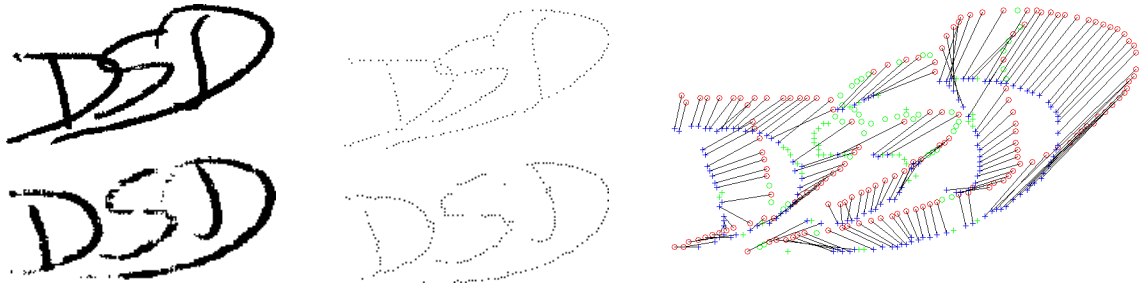
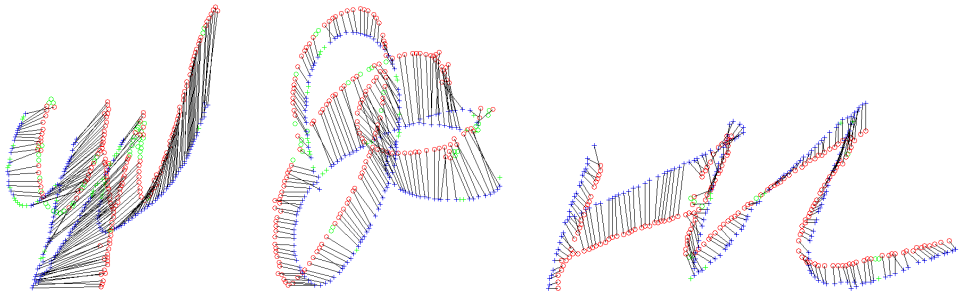


Figure 5: Handwriting matching. Left column: two handwritten initials from the same person. Middle column: points sampled from the skeletons (each with 200 points). Right column: point matching results using our approach.



(a) (b) (c)

Figure 6: More examples of handwriting matching using our approach.

good results for the pair of curves even though the deformation between them is large. Neighboring points may swap their matches, however, for the TPS-RPM algorithm. For the first pair of closed contours, all algorithms achieved reasonable results, but the shape context algorithm made a few matching errors as shown in the middle column of Fig. 4. Since the rotation between two shapes is large for the second pair of closed contours, the rotation invariance shape context is used for initialization in our approach and the shape context algorithm. Both our approach and the TPS-RPM algorithm achieve good results and preserve the sequential ordering of points. The result of the shape context algorithm is not as good: neighboring points in one shape may be matched to points far apart in the other shape.

Handwriting is a non-rigid shape that is of particular interest. The left column of Fig. 5 shows two samples of handwritten initials from the same person. We see that the structural change for handwriting is large: the characters overlap each other in the first sample, but they are well separated in the second sample. We randomly sample 200 points from the skeletons of the handwriting, as shown in the middle column of Fig. 5. The right column of Fig. 5 shows the point matching results using our approach. Points labeled with green color are outliers rejected by our algorithm. On the D's, most points are assigned with correct correspondence. The touching parts of the S are assigned with low matching probabilities, therefore rejected as outliers. More examples of handwriting matching are shown in Fig. 6.

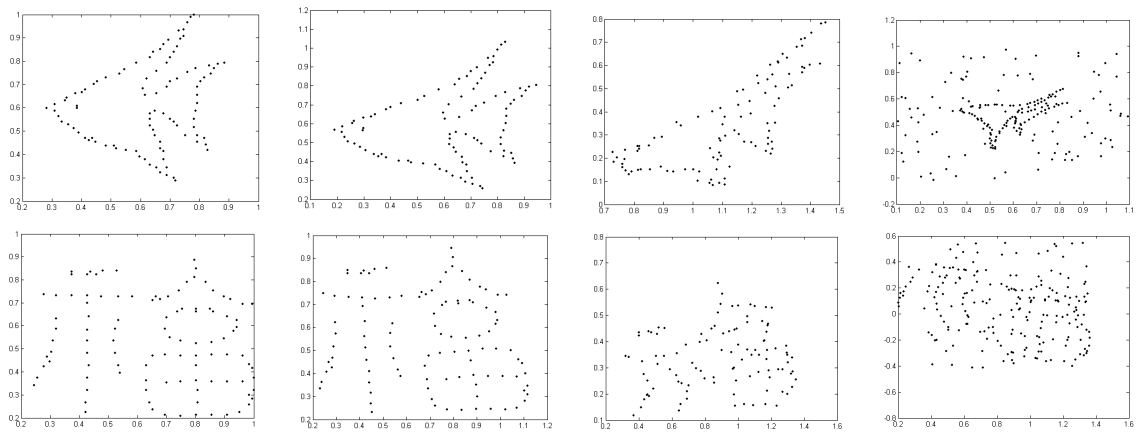


Figure 7: Chui-Rangarajan's synthesized data sets. The template point sets are shown in the first column. Column 2-4 show examples of target point sets for the deformation, noise, and outlier tests respectively.

6.2 Experiments on Chui-Rangarajan's Synthesized Data

Synthetic data is easy to obtain and can be designed to test a specified aspect of an algorithm. We test our algorithm on the same synthesized data as in [9] and [17]. There are three sets of data designed to measure the robustness of an algorithm under deformation, noise and outliers. In each test, the template point set is subjected to one of the above distortions to create a "target" point set (for the latter two test sets, a moderate amount of deformation is present). Two shapes (a fish and a Chinese character) are used, and 100 samples are generated for each degradation level. We then run our algorithm to find the correspondence between these two sets of points and use the estimated correspondence to warp the template shape. The accuracy of the matching is quantified as the average Euclidean distance between a point in the warped template and the corresponding point in the target. Alternative evaluation metrics are possible (e.g., the number of correctly matched point pairs), but in order to compare our results directly with two other algorithms, we use the above evaluation metric as in [9] and [17]. Several examples from the synthesized data sets are shown in Fig. 7.

The mean and variance of the performance of three algorithms (the TPS-RPM, shape context algorithms, and our approach) are shown in Fig. 8. Our algorithm performs best on the deformation and noise sets. For the outlier test set, however, there is no clear winner. The TPS-RPM algorithm outperforms our algorithm on the Chinese character shape under large outlier ratios. Since points are spread out on the Chinese character shape, when a large number of outliers are present, the neighborhood of a point changes significantly (as shown in Fig. 7), which violates our assumption. On the contrary, points on the fish shape are clustered, and the neighborhood of a point is preserved well even under a large outlier ratio, as shown in the last column of Fig. 7. Therefore, better results are achieved by our algorithm on this shape.

In Fig. 8, the variance of all algorithms is large. Therefore, a statistical analysis must be applied to ascertain whether the difference between these algorithms is significant. Mean and variance can only fully characterize a Gaussian distribution. Fig. 9a and

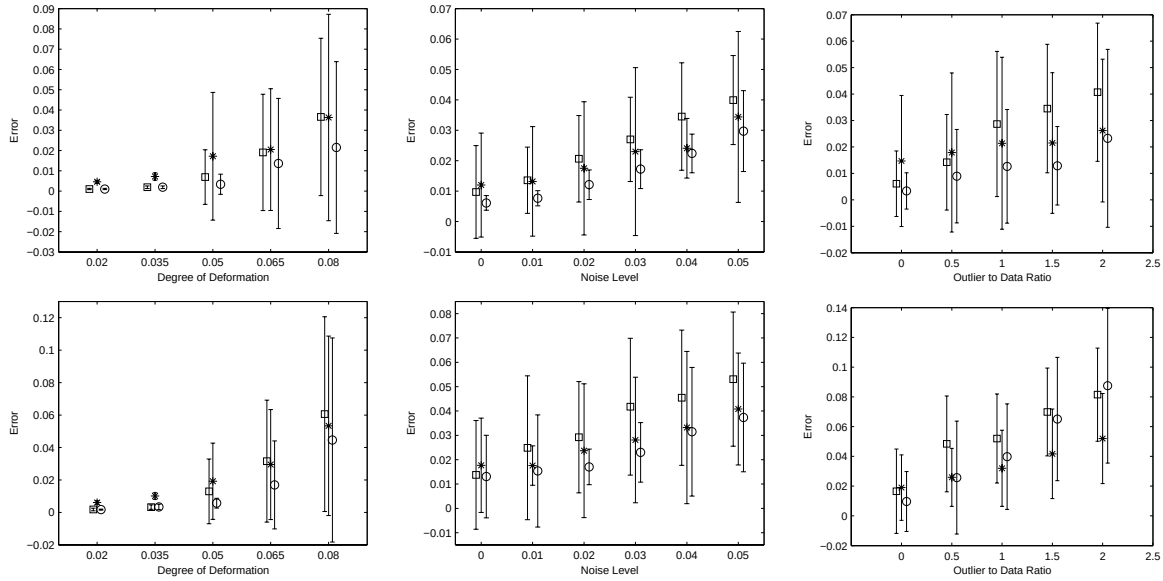
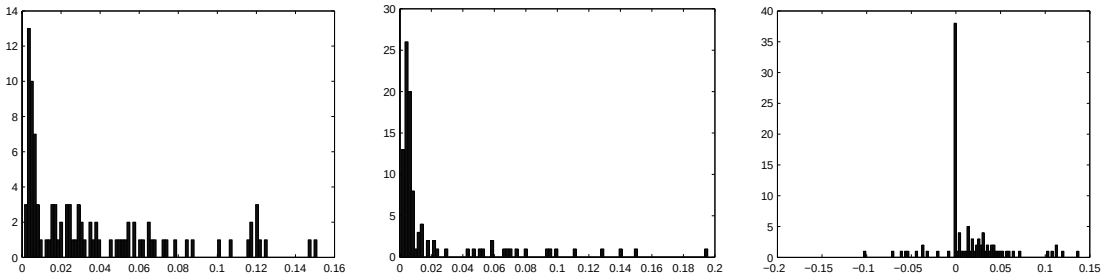


Figure 8: Comparison of our results ($\circ$) with the TPS-RPM ($*$) and shape context ($\square$) algorithms on the Chui-Rangarajan's synthesized data. The error bars indicate the standard deviation of the error over 100 random trials. Top row: the shape of fish. Bottom row: the shape of a Chinese character.

b show the error histograms of the shape context algorithm and our approach. The histograms are generated on 100 trials of the fish shape under the deformation level of 0.05. The distributions are far away from a Gaussian distribution. Some challenging samples significantly deteriorate the performance and increase the variance, and the performance of two algorithms on the same sample is not independent. Fig. 9c shows the histogram of paired differences between two algorithms (the error of the shape context algorithm minus that of our approach). The two algorithms have the same performance for about one third of the test samples, and our approach outperforms the shape context algorithm on most of the remaining samples.

Since the distribution of errors is not Gaussian, we use the Wilcoxon paired signed rank test, which is distribution free and powerful [47]. In the Wilcoxon test, paired differences are formed, and the absolute values are ranked. Where ties occur, the average of the corresponding ranks is used. If the difference between two measures is zero, this sample is excluded from the analysis. The sum of the ranks with a positive sign and the sum of the ranks with a negative sign are calculated. The test statistic is the smaller of these two sums. Table 1 shows the statistical analysis (with two-sided significance level of 0.01) of the performance of our approach compared with two other algorithms. Here, $+$ ($-$) means the improvement (deterioration) of our approach is statistically significant compared with the other algorithm. And $=$ means there is no significant difference between two algorithms. The statistical test verifies that the improvement of our approach on most data sets is significant.



(a) (b) (c)

Figure 9: Histogram of errors. (a) The shape context algorithm. (b) Our approach. (c) Paired differences between two methods (the error of the shape context algorithm minus that of our approach).

Table 1: Wilcoxon paired signed rank test. $+$, $-$ and $=$ mean the former algorithm is better, worse, or no difference than the latter respectively.

	Fish			Chinese Character		
	Deformation	Noise	Outlier	Deformation	Noise	Outlier
Ours vs. Shape Context	= = + + +	= + + + + +	+ + + + +	= = + + +	+ = + + + +	+ + + = =
Ours vs. TPS-RPM	+ + + + +	+ + + + = +	+ + + + +	+ + + + +	+ + + + + +	+ = = - -

6.3 Rotation Invariant Matching

In some applications, rotation invariance is a critical property of a shape matching algorithm. In the following experiments, we test our algorithm under rotations using synthesized data of the same fish and Chinese character shapes. A moderate amount of non-linear deformation is applied to a shape, and the ground-truthed correspondences are used to correct the rotation introduced in the deformation. We then rotate the deformed shape. The probability of selecting a clockwise or counterclockwise rotation is equal. Six rotation degrees are used: 0, 30, 60, 90, 120, and 180. One hundred samples are generated for each rotation. The top row of Fig. 10 shows two synthesized samples.

In the following experiments, for the first iteration, the rotation invariant shape context distance is used to initialize the graph matching in our approach. The rotation between two shapes is corrected by the affine transformation in the first iteration. After that, the normal shape context distance is used. Quantitative evaluation results are shown in the bottom row of Fig. 10. We can see that our method is truly rotation invariant, and it consistently outperforms the shape context algorithm. TPS-RPM, however, can only tolerate a rotation up to 60 degrees. The TPS-RPM algorithm often fails to converge to a useful solution if rotation with any degree is allowed [17], so a parameter λ_2 is used to penalize a large rotation in the TPS-RPM algorithm. If λ_2 is set to zero, its performance deteriorates significantly, much worse than our approach at any level of rotation. Therefore, the default setting of λ_2 ($\lambda_2 = 0.01$) is used in this comparison experiment for the TPS-RPM algorithm.

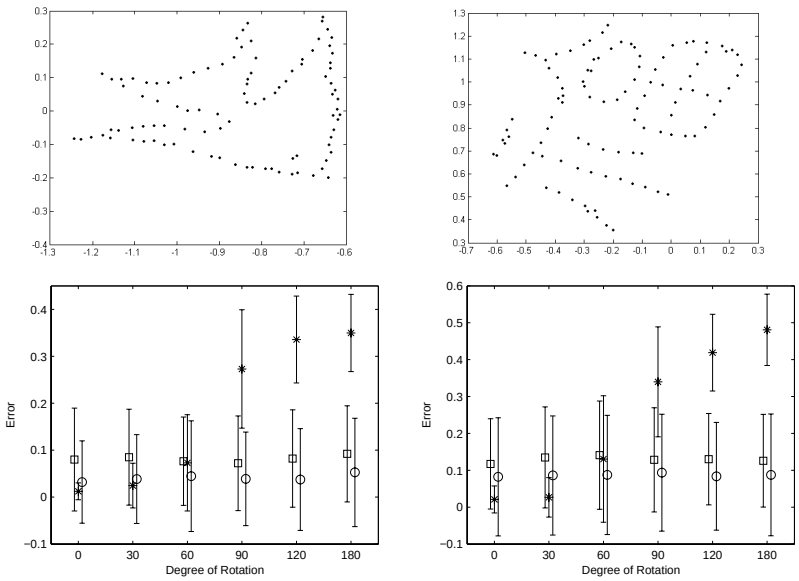


Figure 10: Comparison of our results ($\circ$) with the TPS-RPM ($*$) and shape context ($\square$) algorithms under rotation. Left column: the shape of fish. Right column: the shape of a Chinese character. Top row: synthesized samples. Bottom row: mean and variance of errors.

7 Conclusion and Future Work

In this paper, we have presented a relaxation labeling based point matching algorithm for non-rigid shapes. Based on the assumption that the neighborhood of a point does not change significantly after deformation, we formulate point matching as a graph matching problem. The shape context distance is used to initialize the matching of graphs, followed by relaxation labeling updates. Experiments on a public data set show that our approach clearly outperforms the shape context and TPS-RPM algorithms under non-rigid deformation and noise.

In this work, the relaxation labeling method is used to solve the constrained optimization problem. It is by no means the best approach. We are testing other optimization methods such as simulated annealing, genetic algorithms, and graduated non-convexity methods. Our graph matching formulation is applicable for both 2-D and 3-D shapes. Using the shape context distance for initialization, we only demonstrate it on 2-D shapes, since the original shape context is only defined for 2-D point sets. We will test the effectiveness of our approach for 3-D shape matching by extending the shape context to 3-D point sets. Other initialization methods are possible if more information is available.

A reference C++ implementation of our approach is available under the terms of the GNU General Public License (GPL) at http://www.enee.umd.edu/~zhengyf/PointMatching.htm.

Acknowledgments

The TPS-RPM algorithm used in the comparison experiments is based on the MATLAB® implementation by Dr. Haili Chui and Prof. Anand Rangarajan. The source code of the shape context algorithm comes from Prof. Serge Belongie and Prof. Jitendra Malik. The synthesized data sets used in Section 6.2 were generated by Chui and Rangarajan, and provided to us by Belongie. We want to thank their help for making the comparison experiments much easier.

Appendix

Sinkhorn showed that iterated alternative row and column normalization will convert an $N \times N$ matrix with positive elements to a doubly stochastic matrix [38]. In our relaxation labeling approach, we perform iterated alternative row and column normalization (except the last row and column) to a non-square $K \times N$ ($K \neq N$) matrix A. The purpose of this appendix is to show that this approach is mathematically sound: the process will converge to a unique matrix T_A, such that each row and column of T_A sums one (except the last row and column). The proof in this appendix follows the idea of Sinkhorn. In [38], several important steps are skipped and a few typoes exist. In this appendix more cases are discussed to make Sinkhorn's conclusion more general. First, we will give a formal definition of our generalized doubly stochastic matrix..

DEFINITION 1. A $K \times N$ matrix A is called a generalized doubly stochastic matrix if

$$\sum_{i=1}^{K} a_{ij} = 1 \quad \text{for } j = 1, 2, \ldots, N-1 \tag{28}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \text{for } i = 1, 2, \ldots, K-1 \tag{29}$$

The operation of row normalization can be represented as a left multiplication of A with a diagonal matrix, and the operation of column normalization can be represented as a right multiplication of A with another diagonal matrix. Multiple row (column) normalization matrices can be combined as D_1 (D_2). Therefore, the overall iterated row and column normalization can be represented as $T_A = D_1 A D_2$. The following theorem establishes the uniqueness of such representation.

Theorem 1 *To a given strictly positive $K \times N$ matrix A there corresponds exactly one generalized doubly stochastic matrix T_A which can be expressed in the form $T_A = D_1 A D_2$ where D_1 and D_2 are diagonal matrices with positive diagonals.*
$D_1 = diag\{d_{11}, d_{12}, \ldots, d_{1,K-1}, 1\}$ and $D_2 = diag\{d_{21}, d_{22}, \ldots, d_{2,N-1}, 1\}$. The matrices D_1 and D_2 are unique.

Proof: Suppose there exist two different pairs of diagonal matrices $D1$, D_2 and C_1, C_2 such that $P = C_1 A C_2$ and $Q = D_1 A D_2$ are both generalized doubly stochastic. Then, we can write Q as $Q = D_1 C_1^{-1} P C_2^{-1} D_2$. Let $E = D_1 C_1^{-1}$ and $F = C_2^{-1} D_2$, then

$Q = EPF$. Suppose $E = \text{diag}\{e_1, e_2, \ldots, e_{K-1}, 1\}$ and $F = \text{diag}\{f_1, f_2, \ldots, f_{N-1}, 1\}$, Q can be expanded as

$$Q = \begin{bmatrix} e_1 f_1 P_{11} & e_1 f_2 P_{12} & \ldots & e_1 f_{N-1} P_{1,N-1} & e_1 P_{1N} \\ e_2 f_1 P_{21} & e_2 f_2 P_{22} & \ldots & e_2 f_{N-1} P_{2,N-1} & e_2 P_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ e_{K-1} f_1 P_{K-1,1} & e_{K-1} f_2 P_{K-1,2} & \ldots & e_{K-1} f_{N-1} P_{K-1,N-1} & e_{K-1} P_{K-1,N} \\ f_1 P_{K1} & f_2 P_{K2} & \ldots & f_{N-1} P_{K,N-1} & P_{K,N} \end{bmatrix} \tag{30}$$

The summation of the ith row of Q equals 1, for $1 \leq i \leq K-1$.

$$e_i(f_1 P_{i1} + f_2 P_{i2} + \ldots + f_{N-1} P_{i,N-1} + P_{iN}) = 1 \tag{31}$$

Since $\sum_{j=1}^{N} P_{ij} = 1$ and $P_{ij} > 0$, $1/e_i$ is a convex combination of $\{f_j, 1\}$. Therefore,

$$\min_j\{1, f_j\} \leq \frac{1}{e_i} \leq \max_j\{1, f_j\} \quad \text{for } i = 1, 2, \ldots, K-1 \tag{32}$$

Similarly, we can get

$$\min_i\{1, e_i\} \leq \frac{1}{f_j} \leq \max_i\{1, e_i\} \quad \text{for } j = 1, 2, \ldots, N-1 \tag{33}$$

There are three cases: 1) $\max_i e_i \leq 1$; 2) $\min_i e_i \geq 1$; and 3) $\min_i e_i \leq 1 \leq \max_i e_i$. Let's discuss the first case that $\max_i e_i \leq 1$. Using the second inequality in Eq. (33), we get $f_j \geq 1$. Then second inequality in Eq. (32) becomes $1 \leq e_i \max_j f_j$. It follows that

$$1 \leq \min_i e_i \max_j f_j \tag{34}$$

Similarly, the first inequality in Eq. (33) becomes $f_j \min_i e_i \leq 1$. Therefore,

$$\min_i e_i \max_j f_j \leq 1. \tag{35}$$

Combining the above two inequalities, we get

$$\min_i e_i \max_j f_j = 1 \tag{36}$$

Let consider the summation of the row of Q corresponding to the minimum e_i. Suppose $e_1 = \min_i e_i$

$$\begin{aligned} 1 &= e_1(f_1 P_{11} + f_2 P_{12} + \ldots + f_{N-1} P_{1,N-1} + P_{1N}) \\ &\leq e_1[\max_j f_j(P_{11} + P_{12} + \ldots + P_{1,N-1}) + P_{1N}] \\ &\leq e_1 \max_j f_j(P_{11} + P_{12} + \ldots + P_{1N}) \\ &\leq e_1 \max_j f_j \\ &= 1 \end{aligned} \tag{37}$$

The equality holds if and only if $f_1 = f_2 = \cdots = f_{N-1} = 1$. And considering the column with the maximum f_j, we get $e_1 = e_2 = \cdots = e_{K-1} = 1$.

For the second case, $\min_i e_i \geq 1$, it is easy to verify that

$$\max_i e_i \min_j f_j = 1 \tag{38}$$

And for the last case, $\min_i e_i \leq 1 \leq \max_i e_i$, we can get both equalities (36) and (38). Following similar arguments, we can show that the equalities $f_1 = f_2 = \cdots = f_{N-1} = 1$ and $e_1 = e_2 = \cdots = e_{K-1} = 1$ hold for all cases. It follows that $D_1 = C_1$ and $D_2 = C_2$, and $P = Q$. That means such factorization is unique and the resulted generalized doubly stochastic matrix is unique too.

Theorem 2 *The iterative process of alternately normalizing the rows and columns (except the last row and column) of a strictly positive $K \times N$ matrix is convergent to a strictly positive generalized doubly stochastic matrix.*

Proof: The iteration produces a sequence of positive matrices which alternately have row (except the last row) and column (except the last column) sums one. We will show that the two subsequences which are composed respectively of the matrices with row sums one and the matrices with column sums one each converge to a positive generalized doubly stochastic limit of the form $D_1 A D_2$. The uniqueness part of Theorem 1 will guarantee two limits are actually the same. In the following, we only show the convergence of the subsequence of the matrices with column sums one. The convergence of the other subsequence is easy to show following similar arguments.

Let $\{A_n\} = \{(a_{nij})\}$ be the sequence with column sums one (except the last column), and A_n have row sums $\lambda_{n1}, \lambda_{n2}, \ldots, \lambda_{n,K-1}$. After row normalization, we calculate the column sums δ_{nj} (for $1 \leq j \leq N-1$)

$$\delta_{nj} = \sum_{i=1}^{K-1} a_{nij}/\lambda_{nj} + a_{nKj} \tag{39}$$

Since $\sum_{i=1}^{K} a_{nij} = 1$, δ_{nj} is a convex combination of $\{1/\lambda_{nj}, 1\}$. It follows

$$\frac{1}{\max\{1, \lambda_n(M)\}} \leq \delta_{nj} \leq \frac{1}{\min\{1, \lambda_n(m)\}} \quad \text{for } j = 1, 2, \ldots, N-1 \tag{40}$$

where the m and M respectively label minimal and maximal quantities relative to a given A_n. Similarly, since $\lambda_{n+1,i}$ of matrix A_{n+1} is a convex combination of $\{1/\delta_{nj}, 1\}$, it follows that

$$\frac{1}{\max\{1, \delta_n(M)\}} \leq \lambda_{n+1,i} \leq \frac{1}{\min\{1, \delta_n(m)\}} \quad \text{for } i = 1, 2, \ldots, K-1 \tag{41}$$

There are three cases: 1) $\lambda_n(m) \geq 1$; 2) $\lambda_n(M) \leq 1$; and 3) $\lambda_n(m) \leq 1 \leq \lambda_n(M)$. For the first case $\lambda_n(m) \geq 1$, from Eq. (40) we get $1/\lambda_n(M) \leq \delta_{nj} \leq 1$. Using Eq. (41), we get

$$1 \leq \lambda_{n+1,i} \leq \lambda_n(M) \tag{42}$$

Therefore,

$$\text{case 1:}\quad \lambda_n(m) \geq 1 \quad \Rightarrow \quad 1 \leq \lambda_{n+1}(m) \quad \text{and} \qquad 1 \leq \lambda_{n+1}(M) \leq \lambda_n(M) \tag{43}$$

Similarly

$$\text{case 2:}\qquad \lambda_n(M) \leq 1 \;\Rightarrow\; \lambda_{n+1}(M) \leq 1 \quad \text{and} \quad \lambda_n(m) \leq \lambda_{n+1}(m) \leq 1 \tag{44}$$

$$\text{case 3:}\quad \lambda_n(m) \leq 1 \leq \lambda_n(M) \;\Rightarrow\; \lambda_n(m) \leq \lambda_{n+1}(m) \leq 1 \leq \lambda_{n+1}(M) \leq \lambda_n(M) \tag{45}$$

In the following, we want to show that for case 1 and 3, $\lambda_n(M)$ left converges to 1 (from a value larger than 1); and for case 2 and 3, $\lambda_n(m)$ right converges to 1 (from a value smaller than 1). If the convergence holds, using Eq. (40), it follows that δ_{nj} converges to 1 too. Therefore, the sequence of matrices A_n converges to a generalized doubly stochastic matrix.

Let a_n be the minimal element of A_n (excluding the last row and column), we want to show that $a_n > 0$ for all n. Starting from $A_1 = \{a_{1ij}\}$, we can combine all row normalizations of row i ($i < K$) up to nth iteration as $x_{ni} = [\lambda_{1i}\lambda_{2i}\cdots\lambda_{ni}]^{-1}$. For the last row $x_{nK} = 1$. All column normalization of column j ($j < n$) up to nth iteration is combined as $y_{nj} = [\delta_{1j}\delta_{2j}\cdots\delta_{nj}]^{-1}$. For the last column $y_{nN} = 1$. Since summation of column j of A_n equals one, $\sum_{i=1}^{K} x_{ni}a_{ij}y_{nj} = 1$, for $j = 1, 2, \ldots, N-1$, we get

$$y_{nj} = \frac{1}{\sum_i a_{1ij}x_{ni}} \leq \frac{1}{a_{1ij}x_{ni}} \leq \frac{1}{a_1 x_{ni}} \tag{46}$$

In particular $y_{nj} \leq 1/[a_1 x_n(M)]$. Since

$$\sum_{j=1}^{N} x_{ni}a_{1ij}y_{nj} = \lambda_{n+1,i} \tag{47}$$

As we can see from (43), (44) and (45), for all three cases, $\lambda_{n+1,i}$ is bounded away from 0. Let $\lambda_{n+1,i} \geq \lambda$, it follows that

$$x_{ni} \geq \frac{\lambda}{\sum_j a_{1ij}y_{nj}} \geq a_1\lambda x_n(M)/N. \tag{48}$$

The last inequality is derived from the fact that $a_{1ij} \leq 1$. Also $y_{nj} = 1/\sum_i a_{1ij}x_{ni} \geq 1/[Nx_n(M)]$ and we see that $a_{n+1,i,j} = x_{ni}a_{1ij}y_{nj} \geq a_1\lambda/N^2 = a > 0$. Therefore, $a_n > 0$ for all n.

For case 1 and 3, we want to show that $\lambda_n(M)$ right converge to 1. It is clear that $\lambda_n(M) \to 1 + c$ where $c \geq 0$. For convenience set $\lambda_n(M) = 1 + c_n$.

$$\begin{aligned}
\delta_{nj} &= \sum_{i=1}^{K-1} \frac{a_{nij}}{\lambda_{ni}} + a_{nKj} = \sum_{i:\lambda_{ni}\leq 1} \frac{a_{nij}}{\lambda_{ni}} + \sum_{i:\lambda_{ni}>1} \frac{a_{nij}}{\lambda_{ni}} + a_{nKj} \\
&\geq \sum_{i:\lambda_{ni}\leq 1} a_{nij} + \frac{1}{1+c_n} \sum_{i:\lambda_{ni}>1} a_{nij} + \frac{1}{1+c_n} a_{nKj} = \frac{\sum_{i=1}^{K} a_{nij} + c_n \sum_{i:\lambda_{ni}>1} a_{nij}}{1+c_n}
\end{aligned} \tag{49}$$

Using the fact that $\sum_i a_{nij} = 1$,

$$\delta_{nj} \geq \frac{1 + c_n \sum_{i:\lambda_{ni}>1} a_{nij}}{1 + c_n} \geq \frac{1 + c_n a_n}{1 + c_n} \tag{50}$$

It follows that

$$\lambda_{n+1,i} = \sum_{j=1}^{N-1} \frac{a_{nij}}{\lambda_{ni}\delta_{nj}} + \frac{a_{niN}}{\lambda_{ni}} \leq \frac{1 + c_n}{1 + c_n a_n} \sum_{j=1}^{N-1} \frac{a_{nij}}{\lambda_{ni}} + \frac{a_{niN}}{\lambda_{ni}} \tag{51}$$

Since $0 < a_n < 1$, therefore $(1 + c_n)/(1 + c_n a_n) > 1$, thus

$$\lambda_{n+1,i} \leq \frac{1 + c_n}{1 + c_n a_n} \left(\sum_{j=1}^{N-1} \frac{a_{nij}}{\lambda_{ni}} + \frac{a_{nNj}}{\lambda_{ni}} \right) \tag{52}$$

Because $\sum_{j=1}^{N} a_{nij}/\lambda_{ni} = 1$ (the row summation after row normalization), therefore,

$$\lambda_{n+1,i} \leq \frac{1 + c_n}{1 + c_n a_n} < \frac{1 + c_n}{1 + c_n a} \tag{53}$$

The above inequality holds for all i, particularly,

$$1 + c \leq \lambda_{n+1}(M) < \frac{1 + c_n}{1 + c_n a} \tag{54}$$

Since $c_n \to c$, the above condition holds if and only if $c = 0$. Therefore $\lambda_n(M) \to 1$.

For case 2 and 3, we need to show that $\lambda_n(m)$ left converge to 1. Let $\lambda_n(m) \to 1 - d$ where $d \geq 0$, and $\lambda_n(m) = 1 - d_n$, then

$$\delta_{nj} = \sum_{i:\lambda_{ni}\leq 1} \frac{a_{nij}}{\lambda_{ni}} + \sum_{i:\lambda_{ni}>1} \frac{a_{nij}}{\lambda_{ni}} + a_{nMj} \leq \frac{1}{1 - d_n} \sum_{i:\lambda_{ni}\leq 1} a_{nij} + \sum_{i:\lambda_{ni}>1} a_{nij} + a_{nMj} = \frac{1 - d_n a_n}{1 - d_n} \tag{55}$$

And

$$1 - d \geq \lambda_{n+1}(m) \geq \frac{1 - d_n}{1 - d_n a_n} > \frac{1 - d_n}{1 - d_n a} \tag{56}$$

Since $d_n \to d$, the above condition holds if and only if $d = 0$. It follows $\lambda_n(m) \to 1$. This completes the proof.

References

[1] S. Loncaric, "A survey of shape analysis techniques," *Pattern Recognition*, vol. 31, no. 8, pp. 983–1001, 1998.

[2] R.C. Velkamp and M. Hagedoorn, "State of the art in shape matching," Utrecht University, Netherlands, Tech. Rep. UU-CS-1999-27, 1999.

[3] P.J. Besl and N.D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[4] H. Holstein and B. Li, "Low density feature point matching for articulated pose identification," in *Proc. British Machine Vision Conference*, 2002, pp. 678–687.

[5] P. David, D. DeMenthon, R. Duraiswami, and H.J. Samet, "SoftPOSIT: Simultaneous pose and correspondence determination," *Int. J. Computer Vision*, vol. 59, no. 3, pp. 259–284, 2004.

[6] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P.S. Goldman-Rakic, and J.S. Duncan, "A robust point matching algorithm for autoradiograph alignment," *Medical Image Analysis*, vol. 4, no. 1, pp. 379–398, 1997.

[7] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.

[8] J. Feldmar and N. Anyche, "Rigid, affine and locally affine registration of free-form surfaces," *Int. J. Computer Vision*, vol. 18, no. 2, pp. 99–119, 1996.

[9] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 4, pp. 509–522, 2002.

[10] T. Wakahara, "Shape matching using LAT and its application to handwritten numeral recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 6, pp. 618–629, 1994.

[11] T. Wakahara and K. Odaka, "Adaptive normalization of handwritten characters using global/local affine transform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 12, pp. 1332–1341, 1998.

[12] T.B. Sebastian, P.N. Klein, and B.B. Kimia, "On aligning curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 1, pp. 116–124, 2003.

[13] A.K.C. Wong, M. You, and S.C. Chan, "An algorithm for graph optimal monomorphism," *IEEE Trans. System, Man and Cybernetics*, vol. 20, no. 3, pp. 628–636, 1990.

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

[15] J. Ton and A.K. Jain, "Registering Landsat images by point matching," *IEEE Trans. Geoscience and Remote Sensing*, vol. 27, no. 5, pp. 642–651, 1989.

[16] M. Pelillo, "The dynamics of nonlinear relaxation labeling processes," *Journal of Mathematical Imaging and Vision*, vol. 7, no. 4, pp. 309–323, 1997.

[17] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003.

[18] L. G. Brown, "A survey of image registration techniques," *ACM Computing Survey*, vol. 24, no. 4, pp. 325–376, 1992.

[19] B. Li, Q. Meng, and H. Holstein, "Point pattern matching and applications – a review," in *Proc. Int'l Conf. Systems, Man and Cybernetics*, 2003, pp. 729–736.

[20] F.L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 6, pp. 567–585, 1989.

[21] A.L. Yuille and N.M. Grzywacz, "A mathematical analysis of the motion coherence theory," *Int. J. Computer Vision*, vol. 3, no. 2, pp. 155–175, 1989.

[22] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987.

[23] J. Glaunes, A. Trouvé, and L. Younes, "Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004, pp. 712–718.

[24] W.J. Christmas, J. Kittler, and M. Petrou, "Structural matching in computer vision using probabilistic relaxation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 8, pp. 749–764, 1995.

[25] R.C. Wilson and E.R. Hancock, "Structural matching by discrete relaxation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 6, pp. 634–648, 1997.

[26] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.

[27] A. Rosenfeld, R.A. Hummel, and S.W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. System, Man and Cybernetics*, vol. 6, no. 6, pp. 420–433, 1976.

[28] C.S. Kenney, B.S. Manjunath, M. Zuliani, G.A. Hewer, and A.V. Nevel, "A condition number for point matching with application to registration and postregistration error estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 11, pp. 1437–1454, 2003.

[29] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, 1984.

[30] J. Kittler and J. Illingworth, "Relaxation labeling algorithms – a review," *Image and Vision Computing*, vol. 3, no. 4, pp. 206–216, 1985.

[31] J. Kittler and E.R. Hancock, "Combining evidence in probabilistic relaxation," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 3, no. 1, pp. 29–51, 1989.

[32] O.D. Faugeras and M. Berthod, "Improving consistency and reducing ambiguity in stochastic labeling: An optimization approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 3, no. 4, pp. 412–424, 1981.

[33] R.A. Hummel and S.W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, no. 3, pp. 267–287, 1983.

[34] K.E. Price, "Relaxation matching techniques – a comparison," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 7, no. 5, pp. 617–623, 1985.

[35] L.S. Davis, "Shape matching using relaxation techniques," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 1, no. 1, pp. 60–72, 1979.

[36] O.D. Faugeras and K.E. Price, "Semantic description of aerial images using stochastic labeling," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 3, no. 6, pp. 633–642, 1981.

[37] S.Z. Li, "Matching: Invariant to translations, rotations, and scale changes," *Pattern Recognition*, vol. 25, no. 6, pp. 583–594, 1992.

[38] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The Annals of Mathematical Statistics*, vol. 35, no. 2, pp. 876–879, 1964.

[39] R.C. Wilson and E.R. Hancock, "Bayesian compatibility model for graph matching," *Pattern Recognition Letters*, vol. 17, no. 3, pp. 263–276, 1996.

[40] S. Gold, A. Rangarajan, C. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," *Pattern Recognition*, vol. 31, no. 8, pp. 1019–1031, 1998.

[41] I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis.* John Wiley, Chichester, 1998.

[42] P.J. Rousseeuw and A.M. Leroy, *Robust Regression and Outlier Detection.* John Wiley and Sons, New York, 1987.

[43] D.L. Donoho and P.J. Huber, "The notion of breakdown point," in *A Festschrift for Erich L. Lehmann Belmont*, P.J. Bickel, K.A. Doksum, and J.L. Hodges, Eds. Wadsworth, CA, 1983, pp. 157–184.

[44] N. Duta, A.K. Jain, and K.V. Mardia, "Matching of palmprint," *Pattern Recognition Letters*, vol. 23, no. 4, pp. 477–485, 2002.

[45] G. Wabha, *Spline Models for Observational Data.* Soc. Industrial and Applied Math., 1990.

[46] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, no. 2, pp. 219–269, 1995.

[47] G.K. Kanji, *100 Statistical Tests*. Sage Publications, 1999.