Bi-Stage Large Point Set Registration Using Gaussian Mixture Models

Junfen Chen[1], Munir Zaman[2], Iman Yi Liao[3] and Bahari Belaton[1]
cj11_com079@student.usm.my, munir.zaman@nottingham.edu.my,
iman.liao@nottingham.edu.my, bahari@cs.usm.my

[1]School of Computer Sciences, Universiti Sains Malaysia, Malaysia
[2]Faculty of Engineering, University of Nottingham Malaysia Campus, Malaysia
[3]School of Computer Science, University of Nottingham Malaysia Campus, Malaysia

Abstract. Point set registration is to determine correspondences between two different point sets, then recover the spatial transformation between them. Many current methods, become extremely slow as the cardinality of the point set increases; making them impractical for large point sets. In this paper, we propose a bi-stage method called bi-GMM-TPS, based on Gaussian Mixture Models and Thin-Plate Splines (GMM-TPS). The first stage deals with global deformation. The two point sets are grouped into clusters independently using K-means clustering. The cluster centers of the two sets are then registered using a GMM based method. The point sets are subsequently aligned based on the transformation obtained from cluster center registration. At the second stage, the GMM based registration method is again applied, to fine tune the alignment between the two clusters to address local deformation. Experiments were conducted on eight publicly available datasets, including large point clouds. Comparative experimental results demonstrate that the proposed method, is much faster than state-of-the-art methods GMM-TPS and QPCCP (Quadratic Programming based Cluster Correspondence Projection); especially on large non-rigid point sets, such as the swiss roll, bunny and USF face datasets, and challenging datasets with topological ambiguity such as the banana dataset. Although the Coherent Point Drift (CPD) method has comparable computational speed, it is less robust than bi-GMM-TPS. Especially for large point sets, under conditions where the number of clusters is not extreme, a complexity analysis shows that bi-GMM-TPS is more efficient than GMM-TPS.

1 Introduction

Point set registration has become an active research topic due to its wide applications in object tracking, motion recovery, 3D image reconstruction, stereo matching, to name a few. Registration between two point sets is to find out the meaningful correspondences between the points among the two sets and to recover the underlying spatial transformation that warps one point set onto another [14]. A point set is a collection of the spatial coordinates (locations), while other information such as mesh structure and texture information may not be involved.

In recent years, several state-of-the-art algorithms have greatly influenced the field, including robust point matching algorithm based on thin-plate splines (RPM-TPS) [5], coherent point drift algorithm (CPD) [14] and Jian's method [9, 8], amongest others. For convenience, we call Jian's method GMM-TPS in this paper, which is an extension of a correlation-based point set registration proposed by Tsin and Kanade [17]. Jian and Vemuri represented two point sets as two separate Gaussian Mixture Models (GMMs), and formulated point sets registration as aligning two distribution functions via minimizing their dissimilarity. However, registration speed slows down dramatically as the number of points increases, especially in the case of non-rigid point sets registration. Another problem with GMM-TPS is that it is not reliable when topological ambiguity is exhibited in the data (e.g., see the banana dataset plotted in Fig. 5(a)).

In this paper, we propose a bi-stage point set registration framework to address the above problems. The first stage is a coarse alignment that deals with global deformation. A clustering method is applied to divide the two point sets into clusters separately. The two cluster centers are then registered using a GMM-based method, the transformation of which is subsequently extended to the entire point sets. At the fine registration stage, each pair of clusters between the two point sets is registered also using a GMM-based method to accommodate any local deformation. The alignment problem is solved by minimizing the dissimilarity of two GMMs with respect to thin-plate splines (TPS) transformation. As our proposed method applies GMM-TPS method at both stages in the implementation, we conveniently name it as bi-GMM-TPS.

The proposed bi-GMM-TPS method can deal with point set registration in following situations that are not well handled otherwise: (i) the deformation appears uneven in different parts of the data, and (ii) the two data sets present global misalignment caused by topological ambiguity. It is also worth noting that the proposed method presents a concept of a general top-down hierarchical framework for large point set registration, where the specific registration method in each stage may be methods other than GMM-TPS.

The rest of this paper is organized as follows. An overview of point sets registration is presented in Section 3. A hierarchical bi-stage point sets registration framework and implementation details are described in Section 4. In Section 5, we evaluate the performances of the proposed algorithm. Section 6 concludes the paper and recommends possible future works.

2 Related Works

Point sets registration involves three main topics: (i) the modelling of the point set registration, (ii) the correspondences between the points among the two sets, and (iii) the transformation that aligns one point set onto the other. As for the modelling, Joshi and Lee [10], Luo and Hancock [13] and Myronenko and Song [14, 15], formulated point set registration as a maximum likelihood (ML) estimation problem forcing the GMM centroids to approach the data points; where one point set denotes the GMM centroids and the other represents the data points in a Gaussian mixture. While Tsin and Kanade [17], Jian and Vemuri [9, 8] also considered registration as an alignment of two Gaussian mixtures, they represent the two point sets as two separate GMMs centroids. By considering structure information of a point set as a weighted neighborhood graph, the point matching problem can be formulated as a probabilistic graphical model that can be solved by maximizing its associated probability [6, 3]. A Riemannian framework of point cloud matching was first proposed by Deng et. al. [21]. They treated point clouds matching as a shape matching problem where the point cloud is represented by a Schrödinger Distance Transform (SDT) shape representation.

Chui [4] considered the latter two components of registration as two variables. All registration methods can be categorized into two types, according to the methods handling these two variables. One type of registration method attempts to determine the transformation without needing to establish the explicit point correspondences. This category includes density-based alignment approaches and TPS based registration algorithms [9, 8]. Under a similarity transformation, instead of the complex non-rigid transformation in points matching, when the transformation variables are eliminated, the least squares optimization problem with respect to correspondences constraints decomposes into a concave optimization with global optimality [23]. In [24], the concave optimization problem is further studied to reduce the complexity of computing the lower bound by a k-cardinality linear assignment.

The other type of point set registration method attempts to solve correspondences and transformation simultaneously via some alternative updating scheme such as expectation maximization (EM) [1]. Examples include iterative closest point (ICP) algorithm and its variations [7, 12], robust point matching algorithm (RPM-TPS) [5], and coherent point drift algorithm (CPD) [14, 15].

Recently, clustering methods have also been incorporated in some approximation schemes to deal with large point sets registration problems. A quadratic programming based cluster correspondence projection (QPCCP) algorithm was proposed to pursue the approximate solution by relaxing the correspondences to a continuous value [11]. A farthest-point clustering was used to group point set X into N clusters and point set Y into M clusters [19]. Here, computational complexity decreases by considering the relatively fewer cluster correspondences instead of dense point correspondences. Subsequently, the recovery of point correspondences from cluster correspondences is carried out by a simple substitution.

A further approximation of the point registration algorithm is based on clusters and a generalized radial basis function [20]. This approach was a variant of RPM-TPS where Gaussian kernel function replaces TPS to construct a non-rigid mapping. Deterministic annealing was adopted to update the weight matrix and the correspondence matrix. K-means clustering [1] was used to group one set into a number of clusters, while the other set remains untouched. The number of clusters is gradually increased leading to a coarse-to-fine matching. The proposed method is efficient and beneficial for large and unevenly distributed data. However, it is sensitive to missing and rough correspondences.

Although the proposed method is also based on clustering, it is different from the aforementioned approximation methods in that (i) the result from registering the cluster centers is not the end but the input to a second fine alignment stage; (ii) all points in both point sets are involved in the registration process, and (iii) it is insensitive to missing or rough correspondences (inherited from GMM-TPS).

3 Gaussian Mixture Models-Thin Plate Splines (GMM-TPS)

This section provides a summary of GMM-TPS [9]. TPS is an effective radial basis function (RBF) for representing coordinate mappings $\mathbb{R}^d \to \mathbb{R}^d$ (d=2 or 3). Given a control point set $Q = \{q_1, q_2, \ldots, q_t\}$, The TPS mapping function is defined as:

$$T(p) = pA_1 + h + \sum_{i=1}^{t} w_i U(\|p - q_i\|) \tag{1}$$

where A_1 is a $d \times d$ rigid transformation parameter, $p, q_i \in \mathbb{R}^d$, and $U(r)$ is the radial basis function.

A homogeneous coordinates trick (a point denoting as $[1, p_x, p_y]$) is introduced to lead to the following form of the TPS mapping function

$$T(p) = pA + UW \tag{2}$$

where $A = \begin{bmatrix} h \\ A_1 \end{bmatrix}$ is a $(d+1) \times d$ global affine parameters, W is a $t \times d$ local non-rigid warping parameters and U is a row vector describing structure information between point p and control set Q.

Given two finite point sets $M = \{m_1, m_2, \ldots, m_a\}$, and $S = \{s_1, s_2, \ldots, s_b\}$, where $m_i, s_j \in \mathbb{R}^d$. Develop two Gaussian mixtures for M and S respectively:

$$f(x) = \sum_{i=1}^{a} \alpha_i \phi(x; T(m_i), \Sigma_i^2) \tag{3}$$

$$g(x) = \sum_{j=1}^{b} \beta_j \phi(x; s_j, \Pi_j^2) \tag{4}$$

where x is a spatial point (a vector); α_i, β_j are weights for Gaussian function ϕ; Σ_i^2, Π_j^2 are covariances, and T denotes the TPS transformation. The model can be simplified by assuming equal weights and isotropic covariances in (3) and (4).

Jian and Vemuri [9] have also pointed out that the final registration results were similar for most reasonable selections of covariance in their experiments.

The two point sets registration problem can be formulated as aligning two Gaussian mixtures. If two point sets are aligned sufficiently accurately, the corresponding two mixtures would be expected to be highly similar. A cost function for this is given by:

$$J = \int f^2 dx - 2 \int fg \, dx + \frac{\lambda}{2} \|LT\|^2 \tag{5}$$

The first two terms measure the similarity between two Gaussian mixtures via a L2 distance. The last term is a regularization term to control the smoothness of TPS mapping function. Parameter λ balances the regularization strength. The resulting bending energy is $trace(W^T KW)$, where the kernel matrix $K = K_{i,j} = U(|q_i - q_j|)$, describes the internal structure information among the control point set. An elegant closed-form solution is provided by a gradient-based numerical optimization algorithm L-BFGS-B in [18].

4 Bi-Stage Point Sets Registration Framework

4.1 The bi-stage registration framework

K-means clustering is a popular unsupervised learning algorithm. It groups similar spatial points to form a cluster, thus its center represents the spatial structure of the cluster to a certain degree. Motivated by it, the first stage alignment is designed to obtain a global and coarse TPS transformation on cluster centers. $X = \{x_1, x_2, \ldots, x_m\}$ is clustered into k clusters denoting as $X_i = \{x^i_1, x^i_2, \ldots, x^i_{n_i}\}$, $i = 1, 2, \ldots, k$, and the centers set is $C = \{c_1, c_2, \ldots, c_k\}$. Similarly, $Y = \{y_1, y_2, \ldots, y_n\}$ is clustered into k clusters as $Y_j = \{y^j_1, y^j_2, \ldots, y^j_{n_j}\}$, $j = 1, 2, \ldots, k$, and centers set is $S = \{s_1, s_2, \ldots, s_k\}$. For registration, there is no specific requirement on the number of clusters. Without loss of generality, we therefore set the number of clusters in both point sets to be the same.

Two Gaussian mixtures are generated for C and S respectively. A TPS mapping function $u(A^{(1)}, W^{(1)})$ is computed by minimizing the divergence of two Gaussian mixtures using gradient-based optimization L-BFGS-B. We update set X and its clusters $X_1, X_2, \ldots, X_k$ and denote them as $\tilde{X}$ and $\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_k$ respectively. The resultant alignment is capable of effectively removing the influence of global misalignment.

In addition, this coarse alignment provides a finer initialization for further local registration, as the resultant TPS function $u(A^{(1)}, W^{(1)})$ can be extended to the whole set X, by replacing U with a $m \times t$ matrix describing the structure information of set X and control point set Q. The correspondences between the cluster centers are recovered through a bijection function from center set C to S using a nearest neighbour scheme.

At the fine registration stage, dense point registration is conducted on each cluster pair $(\tilde{X}_i, Y_j)$. The resulting TPS function $u(A^{(2)}_i, W^{(2)}_i)$ is used to warp cluster $\tilde{X}_i$ onto Y_j. According to GMM-TPS implemented in the proposed framework, it is obvious that the registration result is influenced only by the points within a cluster, its corresponding cluster, and the control points. The method of aligning two distribution functions based on TPS reduces any possible interference between two clusters X_i and X_j, $i \neq j$. This inherent attribute greatly benefits the division-based alignment for large point sets. A larger number of control points yield a more flexible deformation, however, this will increase computation time. In this paper, less control points obtained from a sparse spacing are used at the coarse alignment stage, with more control points from a dense spacing introduced at the fine registration stage.

Our bi-stage point sets registration algorithm (bi-GMM-TPS) is summarized as follows:

Algorithm 1 Bi-Stage Point Set Registration with K-means Clustering

procedure Bi-GMM-TPS
    $X \leftarrow$ moving model point set X
    $Y \leftarrow$ fixed scene point set Y
    $Q \leftarrow$ control point set Q
    $k \leftarrow$ no. of clusters
    Clustering:
    $(C1, \mu 1) \leftarrow$ (indices, locations) of cluster centres X
    $(C2, \mu 2) \leftarrow$ (indices, locations) of cluster centres Y
    Global Alignment:
    $A^{(1)}, W^{(1)} \leftarrow$ align $\mu 1$ onto $\mu 2$ using GMM-TPS
    $\tilde{X} \leftarrow$ update X with TPS parameters $A^{(1)}, W^{(1)}$ using TPS warping
    $f \leftarrow$ bijection: assign each element of $\mu 1$ to its nearest neighbour in $\mu 2$
    Local Fine Registration:
    for i=1 to k **do**
        $x \leftarrow \tilde{X}_i$
        $y \leftarrow \tilde{Y}_{f(i)}$
        $A_i^{(2)}, W_i^{(2)} \leftarrow$ align x onto y using GMM-TPS

4.2 Computational Complexity Analysis

GMM-TPS In GMM-TPS, the computational cost is:

1. Computing two integral values $\int f^2 dx$ and $\int fg dx$: $\mathcal{O}(m^2) + \mathcal{O}(mn)$; and
2. Computing kernel matrices U and K: $\mathcal{O}(mt) + \mathcal{O}(t^2)$.

Thus, the total computational cost for GMM-TPS is

$$d1 = \mathcal{O}(m^2) + \mathcal{O}(mn) + \mathcal{O}(mt) + \mathcal{O}(t^2). \quad (6)$$

BI-GMM-TPS Assume the number of control points in our bi-GMM-TPS be t (the same as in GMM-TPS). The number of clusters is k. The computing cost is:

1. K-means clustering on point sets X and Y: $\mathcal{O}(km) + \mathcal{O}(kn)$;
2. Initial matching process: $\mathcal{O}(k^2) + \mathcal{O}(k^2) + \mathcal{O}(kt) + \mathcal{O}(t^2)$;
3. Building bijection function from $\mu 1$ to $\mu 2$: $\mathcal{O}(k^2)$, and
4. Local fine registration: $\mathcal{O}\left(\frac{m^2}{k}\right) + \mathcal{O}\left(\frac{mn}{k}\right) + \mathcal{O}(mt) + \mathcal{O}(kt^2)$.

The total cost for bi-GMM-TPS is therefore

$$\begin{aligned} d2 = &\mathcal{O}(km) + \mathcal{O}(kn) + \mathcal{O}(k^2) + \mathcal{O}(kt) + \mathcal{O}(t^2) \\ &+ \mathcal{O}\left(\frac{m^2}{k}\right) + \mathcal{O}\left(\frac{mn}{k}\right) + \mathcal{O}(mt) + \mathcal{O}(kt^2). \end{aligned} \tag{7}$$

Without loss of generality, let $m = \min(m, n)$, $d1 = (m^2 + mn + mt + t^2)$, $d2 = (km + kn + k^2 + kt + t^2 + \frac{m^2}{k} + \frac{mn}{k} + mt + kt^2)$. When $k = 1$ or $k = m$, we can obtain $d2 > d1$. That is to say, two extreme situations lead to a higher computational cost for bi-GMM-TPS.

Comparison between GMM-TPS and bi-GMM-TPS When the computation time of the initial matching process is negligible, and the number of clusters k, satisfies $\frac{m-\sqrt{m^2-4m}}{2} < k < \frac{m+\sqrt{m^2-4m}}{2} \quad (m > 4)$; $d2 < d1$ holds.

Furthermore, we consider the general case. With increasing value of k, $d2$ decreases initially and then increases. Suppose, when $k = k_0$, the minimum of $d2$ is reached, denoted as $\tilde{d2}$. We can see from (7), as t increases $\tilde{d2}$ increases, leading to $d2 > d1$. In practice, the control points will not be too great, otherwise this will increase the computational complexity and the risk of over-fitting. Thus the decision to disregard the computational time for the initial alignment stage can be justified.

5 Experimental Results and Discussion

In this section, extensive experiments are carried out to investigate the performance of the proposed method (bi-GMM-TPS) on various data sets. Quantitative and qualitative comparisons with state-of-the-art registration methods are also presented. All experiments were performed on Matlab, on a PC with 4GB of RAM and a 2.8 GHz Intel Xeon W3530 CPU running Windows 7 (64-bit).

5.1 Data Analysis

There are eight public data sets being used in the experiments. Four of them are obtained from cited literature in computer vision [14, 9]. Three are challenging data sets used in machine learning such as banana, Gaussian and swiss roll data. The last one is obtained from the USF 3D face database representing a very large point set [25]. The descriptions of the datasets are in Table 1.

Table 1. The eight datasets used in our experiments:

Dataset Name	**Size** (points x dimension)	**Description**
Dolphin	91×2	2D shape edge
Fish	98×2	2D shape edge
Synthetic face	392×3	3D surface
Gaussian	1800×2	2D points cloud
Banana	8000×2	2D shape edge points cloud
Swiss roll	8000×3	3D points cloud
Bunny	8171×3	3D laser range scan
USF face	75972×3	3D laser range scan

To quantitatively evaluate the performance of the registration methods, three metrics are defined in this paper. First, *registration error* is defined as a mean L2 distance between the transformed point set and the scene point set

$$\text{registration error} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\|x_i - y_j\|_{L2} \tag{8}$$

Another measurement is the *number of correspondences*. If the distance between a point in one set and its nearest neighbour in another set is less than a predefined threshold, it is considered that this point has found its correspondence. This metric is defined as:

$$\text{number of correspondences} = |C_X| \tag{9}$$

where $|\cdot|$ is the cardinality of a finite point set; and the correspondences set C_X is generated by

$$C_X = \{y_k, i = 1, 2, \ldots, m |\ d(x_i, y_k) < \tau;\ \text{where } d(x_i, y_k) = \min_j d(x_i, y_j)\} \tag{10}$$

The third metric is *recall*, that was first used in [16] and adopted in [9]. Recall is defined as the proportion of the number of the true-positive correspondences to the total number of ground truth correspondences. Under a reasonable assumption that one-to-one correspondence happens in two clean point sets with the same cardinality. The number of ground truth correspondences is equivalent to the cardinality of the point set. Thus, recall is computed as

$$\text{recall} = \frac{\text{The number of true-positive correspondences}}{\text{Total number of ground truth correspondences}}. \tag{11}$$

5.2 Performance Evaluation of Bi-GMM-TPS

Experiments on choice of clustering methods. Two clustering methods, K-means and K-medoids, are used in the experiments to compare the performance of bi-GMM-TPS on different choices of clustering methods. The results are shown in Table 2. One can choose other clustering methods according to the actual problem one is dealing with.

Table 2. Comparison of registration performance based on K-means and K-medoids.

Dataset	Registration Error		Number of Correspondences		Recall		Time(s)	
	k-means	k-medoids	k-means	k-medoids	k-means	k-medoids	k-means	k-medoids
Dolphin	0.0342	0.1588	79	39	0.7143	0.1538	0.2148	0.3291
Fish	0.0212	0.0731	93	71	0.4794	0.1327	0.3910	0.2612
Synthetic face	0.0926	0.3077	336	275	0.4388	0.0638	1.20	1.9939
Gaussian	1.0651	1.2007	1771	1780	0.0139	0.0100	1.7197	2.6628

Due to out of memory problem when computing the k-medoids, we do not have results for swiss roll, banana, bunny and USF 3D face data.

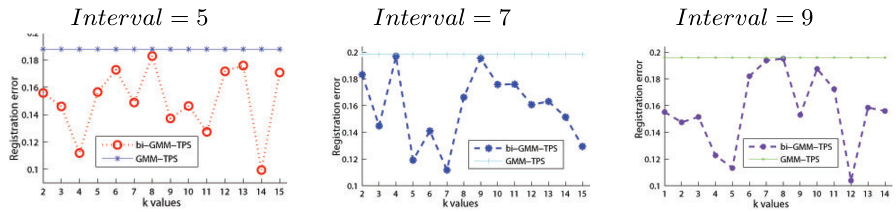
We can see from Table 2 that for bi-GMM-TPS method:

1. K-means can provide lower registration errors than K-medoids on the given data;
2. The recalls based on K-means are higher than those based on K-medoids.
3. For the given distance threshold, K-means found more correspondences on the first three data sets. However, K-medoids obtained higher number of correspondences on Gaussian points cloud;
4. K-means based algorithm runs faster than K-medoids based algorithm on dolphin and Gaussian data, while it is slightly slower on the synthetic face data.

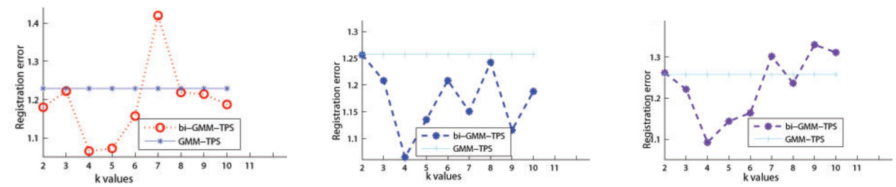
The recall for points cloud data (i.e. Gaussian data) is lower than the recalls for shapes and surface data (i.e., dolphin, fish, and synthetic face data), although the number of correspondences of the former is higher. Furthermore, note that K-medoids finds cluster centers in $\mathcal{O}(N^2)$ time while K-means takes $\mathcal{O}(N)$ time. Based on the above, we find that bi-GMM-TPS using K-means is better than using K-medoids, and therefore K-means is used in all of the following experiments.

Experiments on number of clusters, number of control points. Without loss of generality, face data and Gaussian data are chosen for the two groups experiments that are carried out here. The registration results with varying the number of clusters and the number of control points are shown in Fig. 1. The size of the control points is computed by $t = (\text{interval})^d \times d,\ d = 2 \,\text{or}\, d = 3$. The higher the interval value, the larger the number of control points.

Fig. 1(a) shows the registration errors (i.e. mean L2 distance) of bi-GMM-TPS on synthetic face data. Fig. 1(b) displays the registration errors on Gaussian data. The x-axis is the number of clusters (k) varying from 2 to 15 for (a) and from 2 to 10 for (b) respectively. For each k value, we have conducted 10 runs



(a) Synthetic face dataset



(b) Gaussian dataset

Fig. 1. Registration error of bi-GMM-TPS with respect to k and control points. The horizontal line represents the registration error of GMM-TPS, while the scatter plot represents the registration error of bi-GMM-TPS.

due to the random initialization for K-means, and the optimal results are shown here. The reason is that our main objective is to evaluate the bi-stage registration framework with a valid clustering result. There may be other clustering methods that can produce more stable cluster centres than K-means, however, it is out of the scope of this paper and can be addressed in our future work.

As for synthetic face data (see Fig. 1(a)), registration errors for all interval values and all k values in this experiment are less or no greater than the registration errors of the original GMM-TPS. The average registration error with interval value 5 is the smallest. There is no apparent trend between the registration error and the k values. However, as for Gaussian data (see Fig. (1(b)), it is observed that the registration error has an approximately increasing trend when k is larger than 4, especially when interval is 9. Furthermore, when k is 4, the optimal alignment is arrived at for all three intervals. This is due to the fact that Gaussian data has four peaks. When the k value is larger than 4, it is more likely to result in larger registration errors than the GMM-TPS method as evident in k=7 with interval=5; and k=7, 9, 10 with interval=9. It is easy to understand as Gaussian data has a clear clustered pattern and any incorrect clustering result would result in larger registration errors. In comparison with Gaussian data, one may see that for synthetic face center of the clusters, greater stability is shown to be concurrent with structure in the data. It is also evident that the computation complexity of our method rises sharply with an increase in the intervals (shown in Fig. 2).

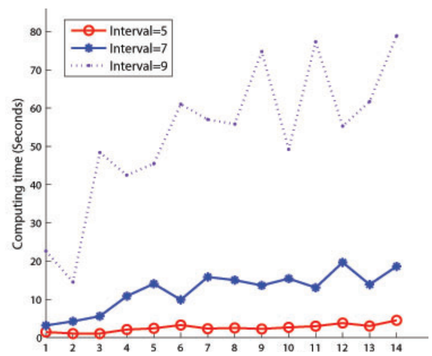


Fig. 2. Computing time on synthetic face data.

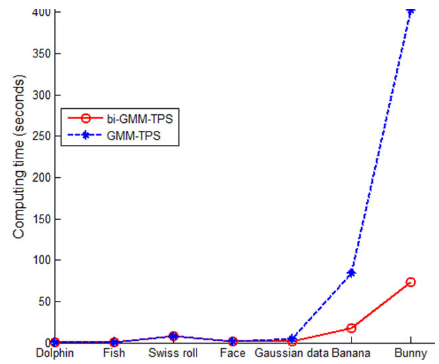
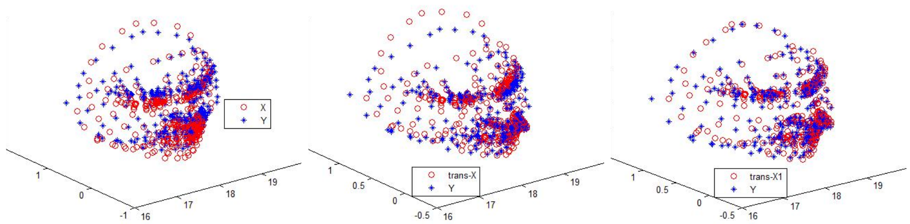


Fig. 3. Comparison of computing times on seven datasets.

Fluctuation of the registration errors is due to the randomness of the initialization of K-means, that may lead to different center locations each time the bi-GMM-TPS algorithm runs. Deeper studies on this shall be carried out in the future.

Experiments on choice of initial alignments Bi-GMM-TPS is a general hierarchical framework for large point set registration, the registration methods in each stage may be other choices than GMM-TPS. ICP-GMM-TPS algorithm refers to the alignment method being iterative closest point (ICP) at the first stage on two clustered centers sets. ICP-GMM-TPS method is tested on synthetic face and banana dataset without any preference and the only intention is to validate the registration performances.



(a) (b) (c)

Fig. 4. Point sets registration on synthetic face data. (a) The synthetic face dataset (input data); (b) ICP-GMM-TPS registration; (c) Bi-GMM-TPS registration.

For synthetic face dataset, comparison results between bi-GMM-TPS and ICP-GMM-TPS are shown in Fig. 4 and Table 3. For banana dataset, we have

Table 3. Comparison of registration performance on synthetic face data.

Algorithm	Registration error	Time (s)
Bi-GMM-TPS	0.0933	8.24
ICP-GMM-TPS	0.1229	9.02

compared ICP and GMM-TPS by registering two clustered center sets. The registration error of GMM-TPS is 8.1924 and the registration error of ICP is 8.8993. More importantly, GMM-TPS remedies the topological ambiguity of banana data, while ICP failed to do so.

Both quantitative performance and visualization of bi-GMM-TPS outperform those of ICP-GMM-TPS. The comparisons on these two datasets demonstrate that bi-GMM-TPS is more reliable and flexible than ICP-GMM-TPS when dealing with non-rigid registration.

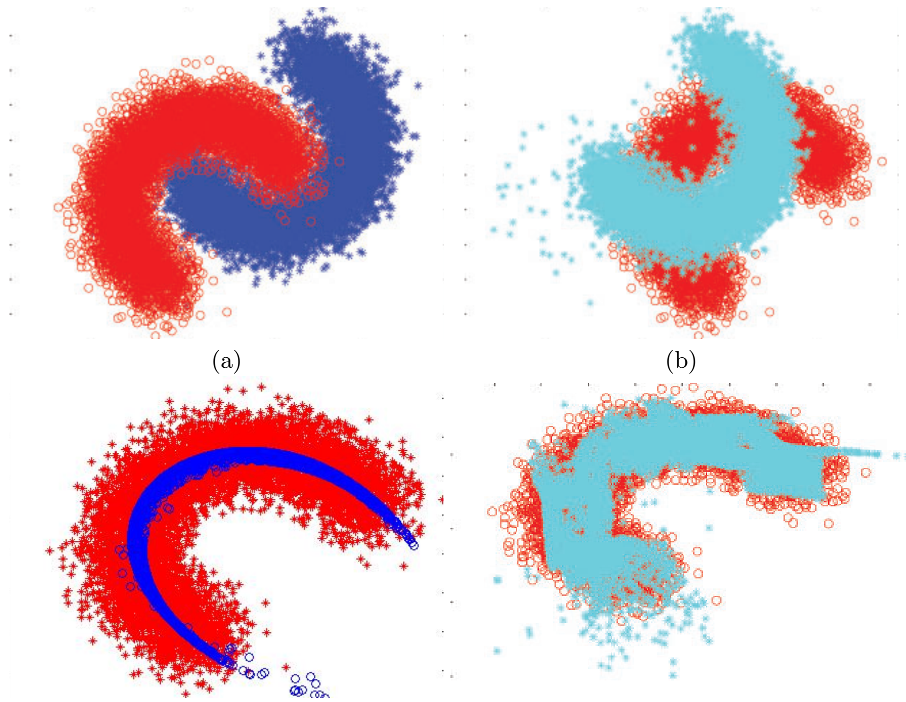
5.3 Comparisons with GMM-TPS, CPD and QPCCP

The above three comparators are chosen for comparing with the proposed bi-GMM-TPS since

1. GMM-TPS is the basis of the specific implementation in the framework, and
2. CPD is an alternative comparable non-rigid registration method especially for large point sets, and
3. QPCCP has used a clustering method for approximate registration, and is therefore a suitable comparator.

Comparison results between bi-GMM-TPS and GMM-TPS are shown in Figs. 1 and 3. The computational time of two algorithms on seven data sets are plotted in Fig. 3. When the size of the input point set is small, bi-GMM-TPS is slower than GMM-TPS. However, bi-GMM-TPS outperforms GMM-TPS on large data sets such as the Gaussian, banana and bunny data. A comparison of registration errors is carried out on face and Gaussian data in Fig. 1. The horizontal lines are the registration errors of GMM-TPS. All registration errors of bi-GMM-TPS are less than those of GMM-TPS on face data, and most of the registration errors of bi-GMM-TPS are smaller on Gaussian data.

Registration results on banana data shown in Fig. 5, where we set $k = 12$ and interval as 5, clearly demonstrate that bi-GMM-TPS is able to rectify any global misalignment caused by topological ambiguity in the data (Fig. 5(d)), when GMM-TPS failed to do so (Fig. 5(b)). Although we notice that CPD can also remedy topological ambiguity in the data, the transformed point set has been unevenly warped (Fig. 5(c)). Quantitative comparison is shown in Table 4. Although the difference in the registration error between bi-GMM-TPS and GMM-TPS is negligible, the alignment by GMM-TPS is clearly wrong. Furthermore, the bi-GMM-TPS is much faster than GMM-TPS on registering the



(a) (b) (c) (d)

Fig. 5. Point sets registration on banana data. (a) The banana dataset (input data); (b) GMM-TPS registration; (c) CPD registration; (d) bi-GMM-TPS registration

banana data, while CPD provides the worst registration error and longest computing time among these three methods.

Two 3D face scans chosen at random from the USF 3D face database [25] are used to evaluate the efficiency of bi-GMM-TPS algorithm on large point sets registration. An interval value 5 is used to set the size of the control points. The registration error is 6.3569 and 4.5885 for bi-GMM-TPS and GMM-TPS respectively. There is not much visual difference in terms of registration accuracy. However, the computational time for bi-GMM-TPS is 4.55×10^3 seconds (∼1.26 hours), which is much shorter than that of GMM-TPS 67.11×10^3 (∼18.64 hours). GMM-TPS becomes impractical in dealing very large point sets. CPD method is also carried out on this large dataset and the registration error is 7.7654 with a computational time of 4.42×10^3 seconds(∼1.22 hours). CPD is marginally quicker but has the lowest registration accuracy.

QPCCP is compared with the proposed bi-GMM-TPS, for QPCCP has also used clustering methods for approximate registration. As it has been reported that QPCCP demonstrates less registration accuracy in terms of distance-based metrics as compared to GMM-TPS [20], we compared our method with QPCCP

Table 4. Comparison of registration performance on banana data.

Algorithm	Registration error	Time (s)
Bi-GMM-TPS	5.8493	17.4
GMM-TPS	5.8450	84.4
CPD	7.8130	97.3

using not only the mean L2 distance but also the number of correspondences (refer to section 5.1) on fish data. From Table 5, we can see that bi-GMM-TPS can provide a lower registration error and more correspondences than QPCCP.

Table 5. Comparison of registration performance on fish data.

Algorithm	Registration error	No. of Correspondences
Bi-GMM-TPS	0.0212	93
QPCCP	0.0457	81

6 Conclusion

In this paper, a hierarchical bi-stage registration framework bi-GMM-TPS for large point sets has been proposed. K-means clustering is employed to group large point sets into the same number of clusters respectively. First stage alignment was done on two cluster center sets, where the resultant thin-plate splines (TPS) transformation is applied to the entire point set. Before further registration is conducted on the cluster pairs, the points of one center set find their nearest neighbors in the other center set. Thus, fine registration is performed within the clusters and their corresponding clusters. Extensive experimental results have demonstrated the computational efficiency and robustness of our proposed method. Results have shown that our bi-GMM-TPS method outperforms some of the state-of-the-art point set registration methods GMM-TPS and CPD and QPCCP.

However, registration accuracy of the proposed method depends on the accuracy and reliability of clustering. A clustering method that is robust to point set deformation, and more stable to random initialization will be studied in the future.

Acknowledgement. This work is supported by Postgraduate Research Grant Scheme of Universiti Sains Malaysia (1001-PKOMP-846050) and National Natural Science Foundation of China (61473111).

References

1. Bishop, Christopher M.: Pattern recognition and machine learning. Springer, New York (2006)
2. Bookstein, Fred L.: Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence **11**(6) (1989) 567–585
3. Caetano, T.S., Caelli, T., Schuurmans, D., Barone, D. A.C.: Graphical Models and Point Pattern Matching. IEEE Transaction Pattern Analysis and Machine Intelligence **28**(10) (2004) 1646–1663
4. Chui, H.L.: Non-rigid Point Matching: Algorithm, Extensions and Applications. Yale University, (2001)
5. Chui, H.L., Rangarajan, A.: A New Point Matching Algorithm for Non-Rigid Registration. J. of Computer Vision and Image Understanding **89**(3) (2003) 114–141
6. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty Years of Graph Matching in Pattern Recognition. International Journal of Pattern Recognition and Articial Intelligence **18**(3) (2004) 265–298
7. Granger, S., Pennec, X.: Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration. European Conference on Computer Vision (ECCV02) **LNCS2353** (2002) 418-432
8. Jian, B., Vemuri, B. C.: A Robust Algorithm for Point Set Registration Using Mixture of Gaussians. 10th IEEE International Conference on Computer Vision (ICCV05) (2005) 1246-1251
9. Jian, B.,Vemuri, B.C.: Robust Point Set Registration Using Gaussian Mixture Models. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(8) (2011) 1633–1645
10. Joshi, A., Lee, Chia-H.: On the Problem of Correspondence in Range Data and Some Inelastic Uses for Elastic Nets. IEEE Transactions on Neural Networks **6**(3) (1995) 716–726
11. Lian, W., Zhang, L., Liang, Y., Pan, Q.: A quadratic programming based cluster correspondence projection algorithm for fast point matching. Computer Vision and Image Understanding **114** (2010) 322–333
12. Liu, Y.: Automatic registration of overlapping 3D point clouds using closest points. Image and Vision Computing **24**(7) (2006) 762-781
13. Luo, B., Hancock, E.R.: A Unified Framework for Alignment and Correspondence. Computer Vision and Image Understanding **92**(1) (2003) 26–55
14. Myronenko, A., Song, X.: Point Set Registration: Coherent Point Drift. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(12) (2010) 2262–2275
15. Myronenko, A., Song, X., Carreira-Perpinan, M.A.: Non-Rigid Point Set Registration: Coherent Point Drift. Advances in Neural Information Processing Systems (NIPS), The MIT Press, **19** (2007) 1009–1016
16. Starck, J., Hilton, A.: Correspondence labelling for wide-timeframe free-form surface matching. 11th International Conference on Computer Vision (ICCV07) (2007) 1–8
17. Tsin, Y., Kanade, T.: A Correlation-based Approach to Robust Point Set Registration. Eighth European Conference Computer Vision **3** (2004) 558–569
18. Zhu, C., Byrd, R.H., Lu, P., Nocedal, J.: Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software **23**(4) (1997) 550-560

19. Gonzalez,T.F.: Clustering to minimize the maximum intercluster distance. Theoretical Computer Science **38** (1985) 293–306
20. Xu, H., Liu, J., Smith, C.D.: Robust and Efficient Point Registration Based on Clusters and Generalized Radial Basis Functions (C-GRBF). IEEE International Conference on Image Processing (2012) 1669–1672
21. Deng, Y., Rangarajan, A., Eisenschenk, S., Vemuri, B.C.: A Riemannian framework for matching point clouds represented by the Schrdinger Distance Transform. IEEE Conference on Computer Vision and Pattern Recognition (2014) 1–8
22. Lou, L., Zeng, H., Xiong, J., Li, L., Gao, W.: Schrdinger Transform of Image: A New Tool for Image Analysis. Measurements in Quantum Mechanics, Mohammad Reza Pahlavani, **11** (2012) 269–292
23. Lian, W., Zhang, L.: Robust Point Matching Revisited: A Concave Optimization Approach. European Conference on Computer Vision (2012) 1–8
24. Lian, W., Zhang, L.: Point Matching in the Presence of Outliers in Both Point Sets: A Concave Optimization Approach. IEEE Conference on Computer Vision and Pattern Recognition (2014) 1–8
25. Blanz, Volker, Vetter, Thomas: Face Recognition Based on Fitting a 3D Morphable Model. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(9) (2003) 1063-1074