Proceedings of 2015 IEEE
International Conference on Mechatronics and Automation
August 2 - 5, Beijing, China

A New Algorithm for Non-rigid Point Matching Using Geodesic Graph Model

Deheng Qian
State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation,
Chinese Academy of Sciences,
Beijing 100190, China

Tianshi Chen
State Key Laboratory of Computer Architecture,
Institute of Computing Technology,
Chinese Academy of Sciences,
Beijing 100190, China

Hong Qiao
State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation,
Chinese Academy of Sciences,
Beijing 100190, China

***Abstract*—Point matching is a problem of finding the optimum matching between two sets of key points which are extracted from the surfaces of objects. A popular approach represents the features of a set of points with a graph model. Traditionally, the measurement applied in the graph model is the Euclidian distance, which is not suitable for objects with non-rigid deformations. In this paper, we propose a novel graph model called the geodesic graph model (GGM) which uses a geodesic-like distance as its measurement. GGM can better tackle non-rigid deformations because the geodesic-like distance is a kind of invariant structural feature during non-rigid deformations. The building process of the GGM is justified under the assumption that all the key points are spanning on a manifold. To further handle the deviations of key point locations, we come up with a feature weighting process to increase our algorithm's robustness. We conduct several experiments on different kinds of deformations over several widely used datasets. Experimental results demonstrate the effectiveness of our algorithm.**

***Index Terms*—Point matching, non-rigid deformation, manifold, geodesic distance.**

I. INTRODUCTION

Point matching, which finds appropriate correspondences between two different point sets, is an important and fundamental problem in the field of computer vision. The application scope of point matching is broad, including object recognition, 3D reconstruct, picture mosaicking, motion detection, robotic grasping and assembling. There have been a number of studies dedicated to this problem [1], which can be roughly categorized into two types: 1, rigid deformation matching; 2, non-rigid deformation matching [2].

Algorithms for rigid deformation matching problem demand the deformation type of the target object to be rigid. In spite of their strict assumptions, these algorithms can solve lots of matching problems including some simple non-rigid deformation ones. For the wide application of them, a lot of studies focus on the rigid deformation matching problem [3]. Random Sample Consensus (RANSAC) [4] and its different variations [5] are commonly used. In addition, this problem has been studied from the graph matching [6], [7] point of view, for example, the spectral methods [8], [9]. However, all these methods have difficulties to handle objects with complex deformations.

Algorithms for non-rigid deformation matching problem handle the situations when complex deformations can-not be well approximated by a rigid one. In this field, typical previous works include [2], [10]. Chui and Rangarajan proposed the TPS-RPM [2] which uses the thin-plate spline [11] to parameterize the non-rigid deformations and the soft assign for the correspondence. One popular approach [12], [13] makes use of the global features such as the spatial relations among points for matching. This kind of algorithms are called structural method [14].

Generally, structural methods cast the matching problem as an energy minimization task by defining a matching objective including a local loss term and a global one, which depend on the local features and global ones respectively. The matching objective calculates the dissimilarity of features of the whole matched pairs of points.

The features of points are represented by a graph model which is built up based on the points in the image. Each edge in the graph describes the features of the relation between its two end nodes. To build a graph model which stably represents the spatial relations among key points, the central problem is how to link all these nodes in a proper way. Extant literatures have proposed several methods to link nodes, including k-neighbors, ϵ-neighbors, Delaunay triangulation et al. They can build a relatively stable graph model while the object undergoing rigid deformations. However, they are unable to achieve high performance under the condition of non-rigid deformations. That is because the spatial relations built by these methods are based on classical distance measurements, such as Euclidian distances, which are very sensitive to the changing of appearance caused by non-rigid deformations. Points far apart on the surface of the target object may appear deceptively close as measured by straight-line Euclidean distance. As a result, the crux of structural methods for non-rigid deformations is that how we can describe the spatial relations by a graph model properly.

In fact, the surface of the target object can be regarded as a manifold which is embedded in 2D image. Hence, the distance measurement in manifold can be used in point matching problem for reference. The geodesic distance which is a popular measurement in manifold can preserve the intrinsic geometry of the manifold, which is particularly useful for providing a kind of invariant feature during non-rigid deformations. Inspired by such measurement, we introduce the geodesic-like distance in a weighted graph model. The geodesic-like distance can preserve the intrinsic geometry among key points as the geodesic distance do among data points in the manifold. For neighboring points, the Euclidean distance provides a good approximation to geodesic-like distance. For faraway points, geodesic-like distance can be approximated by adding up a sequence of short edges between neighboring points. As a result, the graph model which represents the geodesic-like distances among points tends to stay within the manifold which is the surface of the target object. Since the local structural features among key points are more stable than the global ones during non-rigid deformations [15], this graph model can more stably describe the features of key points on objects with non-rigid deformations. The approximations of geodesic-like distances can be efficiently computed in the graph model by finding shortest paths in it.

To build such a graph model, the edges of it should be as short as possible since shorter edges are more likely to change less during the non-rigid deformations. However, if discarding all edges that are longer than a threshold, the graph model may be partitioned into several sub-graphs without any edges connecting them to each other. Hence, some of the geodesic-like distances become infinity. To handle this problem, we establish the minimal spanning tree (MST) of the complete graph of all the key points as the basis of the graph model. The minimal spanning tree makes sure that all nodes are involved in a connected graph while keeping the edges of the graph model as short as possible. However, the structure of MST changes easily while the object undergoing deformations. In order to improve its robustness, we further extend the MST with more edges which are shorter than a threshold. This final graph model is referred to as geodesic graph model (GGM). Based on the GGM, geodesic-like features among points can be extracted for matching.

In order to give our algorithm a complete performance evaluation, we conduct several experiments on both non-rigid deformation matching tasks and rigid ones. Experimental results show the effectiveness of our algorithm compared with others, since our algorithm achieves the best performance in most cases.

The rest of this paper proceeds as follows: Section II discusses some of the related works. Section III details the procedure of getting the geodesic-like features proposed by our model. Section IV formulates the point matching as an energy minimization task. The experimental results are reported in Section V. We draw the conclusions in the final section.

II. Related Work

Non-rigid deformations are handled in various sub fields of computer vision. By regarding the surface of the targeted object as a manifold, algorithms proposed in manifold learning can provide useful insights for the non-rigid deformation matching problem.

Manifold learning [16]–[18] which has been a research focus for feature subspace learning, maintains the geometric relations among data points during the dimensional reduction process. To be more specific, it keeps the same (or as similar as possible) geometric relations among the projected data in the target low dimensional space with those in high dimensional space. Since the data with high dimension are often twisted in space while the projected data are flatten, the dimensional reduction process is actually non-rigid.

In point matching problem, the surface of the target object can be regard as a manifold which is embedded in 2D image. The key points in the image are analogical to the data points in the manifold. In the meanwhile, features of key points are represented by a graph model. With this observation, it is reasonable to use the ideas of manifold learning for reference in the point matching problem which also needs to describe the geometric information of data points. In fact, some of matching studies have similar insight with us in terms of taking advantage of algorithms proposed in manifold learning.

Li et al. proposed an object matching method [19] which is inspired by the Locally Linear Embedding (LLE) algorithm [17] and has a similar formulation. Based on the fact that one data point can be linearly reconstructed by its neighbor points, their method keeps the reconstruction weight to describe the geometric relations among points. However, this method is not suitable for the target non-rigid deformations since the reconstruction weight of each point possibly change too violently to be able to get a satisfied description of the geometry.

Zheng and Doermann proposed a point matching method [15] which has a similar idea with LPP algorithm [20]. Their method matches two points if the correspondence of one point's neighbor be a neighbor of its correspondence. This neighborhood relationship is much robust during non-rigid deformations. However, this method discards all the geometric information between points who are far (not neighbor) from each other.

Focus on the geodesic-like distance proposed in our algorithm, it is inspired by the geodesic distance used in the Isomap [18] which is a popular manifold learning algorithm.

Isomap presents the data points with a graph model which is commonly used in point matching problem. It maintains the spatial relations among data points by keeping the geodesic distances among points constant during the dimensional reduction process. Since the geodesic distance is insensitive to non-rigid deformations especially articulated deformations naturally, it is therefore a good choice for representing the geometric relations among key points in the matching problem.

Some matching algorithms take advantage of the geodesic distance in matching problems as we do. Elad and Kimmel proposed an algorithm [21] used in 3D object matching problem taking advantage of the geodesic distance among key points. It is a typical work using the geodesic distance to describe the spatial relations among key points. Since applied in 3D shapes, it is a direct application of the geodesic distance.

However, the geodesic distance can hardly be applied in the situations of 2D images directly. Without the depth information, the direct counterpart of the geodesic distance between two key points on the surface of a 2D shape is the distance between them along the contour. This distance is not useful in the matching process [22].

To still make use of the intrinsic idea of geodesic distance, some studies come up with new measurements which discard the superficial representation of geodesic distance.

Ling's inner distance method [23] for shape classification is the most similar work with ours. They come up with the inner distance inspired by the geodesic distance as we do with the geodesic-like distance. The biggest difference between our algorithm and Ling's is that, our algorithm calculates the distances only depending on the locations of key points. In contrast, Ling's method calculates distances depending on not only the locations of key points but also the contour of the object. Though their method has a variety of applications [24], it is not appropriate for point matching problem because of its dependence of the contour. When there is no contour or it is too vague to locate its position, this algorithm is not suitable to be applied.

The good results achieved by these methods mentioned above in their own target problems show the useful connection between the matching problem and the manifold learning algorithms.

III. Geodesic-like Features

In this section, we detail the procedure of obtaining the geodesic-like features in our algorithm. First, the building mechanism of the geodesic graph model is introduced. After that, a weight setting process which makes the matching process more robust is presented.

A. Geodesic Graph Model

The minimal spanning tree (MST) is a minimum-cost connected graph which links nodes far from each other via relaying nodes. We select it as the skeleton of our graph model for its flexibility in non-rigid deformations, especially in the articulated deformations. In addition, MST contains the shortest edges which describe the local relations among neighboring points. However, the structure of the MST changes easily when the object undergoing violent deformations. In the meanwhile, the discriminative ability of it is not powerful enough since it contains a few edges.

In order to improve the robustness and the discriminative ability of our graph model, the MST is further extended by adding more edges which are restricted to be shorter than a threshold t. The extended graph model is referred to as the geodesic graph model (GGM). The specific procedure of building GGM are described as follows.

Since the long edges possibly change violently in non-rigid deformations, the edges between nodes in our graph model are designed be as short as possible so that all the edges longer than a threshold are removed in procedure 3.

Algorithm 1: Procedure of building the geodesic graph model.

Input: The coordinations of all the key points.
Output: The geodesic graph model GGM.

1. Calculate the Euclidian distances between every pair of key points to build the complete graph G of all the points.
2. Build the minimal spanning tree Tr of the complete graph G.
3. Remove all the edges longer than a certain threshold t in the graph G to obtain a new graph Gr.
4. Combine Tr and Gr together by adding all the edges of them into a single graph model which is the GGM.
5. **return** The single graph model which is the GGM.

Suppose the key points of an image are intensively located in two separate regions so that no short edges can connect these two regions to each other. As a result, the graph model Gr is partitioned into two sub-graphs gr_1 and gr_2 without any edges connecting these two to each other. Hence, the distance from one node in gr_1 to another node in gr_2 is infinite which is intractable for the solver. We handle this issue by adding edges which belong to Tr to make Gr a connected graph in procedure 4. The edges in Tr are the shortest ones that can link these sub-graphs. To only select edges belong to Tr is consistent with the statement that the edges in our graph model should be as short as possible.

We call the shortest path between two nodes in the graph a link. Than the lengths of links between every pair of nodes are the geodesic-like distances which are computed based on the GGM. Fig. 1 shows the procedure of building the GGM.

B. Weight Setting

While the object undergoing dramatic deformations, the geometric relations between two points who are far from each other are probably not as reliable as these who are closer to each other. So, for a specific node in the graph model, the reliabilities of the geodesic-like features decrease while the lengths of corresponding links getting longer. In order to describe the reliabilities, a reasonable approach is to set weights for different features. Someone uses the learning algorithm [25] to set the weights of features. But learning algorithm is much complicated and is not suitable when training samples are not available. We propose a concise method to set weights on links based on their variances.

Assuming the nodes are infected by Gaussian noise, a link with more nodes passed by has a larger variance.

Proof: there are three nodes: a, b, c and three edges $\overline{ab}$, $\overline{bc}$, $\overline{ac}$. We assume all the node locations P_a, P_b, P_c are infected by Gaussian noise $N(0, \sigma)$. The lengths of $\overline{ab} = \|P_a - P_b\|$, $\overline{bc} = \|P_b - P_c\|$ and $\overline{ac} = \|P_a - P_c\|$ are three random variables whose variances $D(\overline{ab})$, $D(\overline{bc})$, $D(\overline{ac})$ are equal to 2σ which is the sum of the variances of their corresponding end nodes. In another way, we link a and c by a dog-leg path through the node b. Then, the length of the new link from a to c

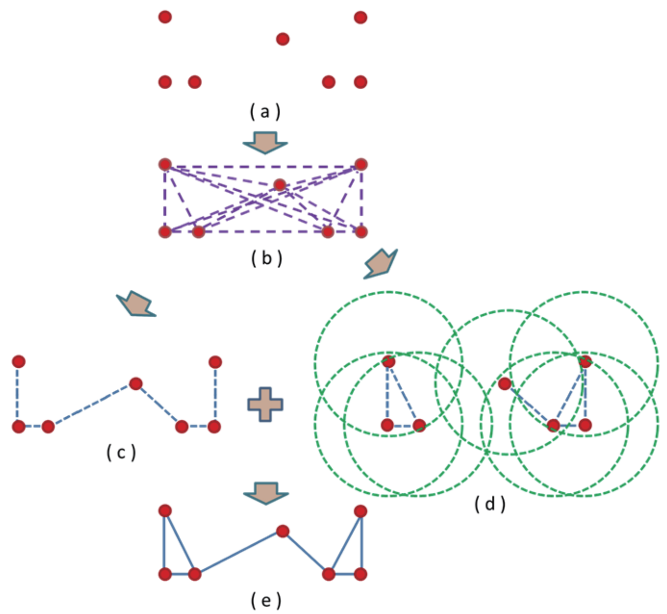

Fig. 1. The schematic diagram of our algorithm: (a) indicates the locations of nodes in a graph model; (b) is the complete graph of all these nodes; (c) shows the minimal spanning tree of (b); (d) is the graph model obtained by discarding edges which are longer than a threshold in (b); (e) is the GGM which is the combination of (c) and (d).

is $\overline{abc} = \|P_a - P_b\| + \|P_b - P_c\|$ with variance $D(\overline{abc}) = D(\overline{ab}) + D(\overline{bc}) + Cov(\overline{ab}, \overline{bc}) \geq \sqrt{D(\overline{ab})D(\overline{bc})} = 2\sigma$ where the $Cov(\overline{ab}, \overline{bc})$ is the covariance between $\overline{ab}$ and $\overline{bc}$. if b lies on the straight line between a and c, the equality hold.

Consequently, we know that the length of a link which consists of two parts have higher variance than connecting the end nodes directly. By mathematical induction, we can generalize this conclusion to regular situations. Finally, we know that the variance of the length of a link increase as the nodes passed by getting more.

We set the weight by function:

$$w_{p_h}^{p_i} = \alpha^{-\lambda_{p_h}^{p_i}} \tag{1}$$

where $\lambda_{p_h}^{p_i} \geq 0$ is the number of nodes passed by from P_i to P_h except the end points and α is the decrease rate.

IV. Problem Formulation

The matching problem is cast as an energy minimization task in our algorithm. The energy function used in our algorithm is introduced first in this section. Then, the optimization scheme of the energy function is presented.

A. Energy Function

The set of key points in an image is represented by a graph model whose nodes and edges describe the local features of points and the relations among points respectively. The graph model can be denoted as $\mathrm{g} = \{V, E, C\}$, where $V = [v_1, v_2, \dots v_n] \in R^{d_v \times n}$ and $E = [e_1, e_2, \dots e_m] \in R^{d_e \times m}$ are feature matrices computed for nodes and links respectively. C specifics the topology of g based on a node-node affinity matrix, in which $C_{ij} = 1$ when the ith and jth nodes are connected directly, and 0 otherwise.

The matching problem is cast as finding correspondences among nodes of two graphs that minimize a local loss term and a global one simultaneously. So, our energy function F has the form as below:

$$F(\widetilde{M}) = L(\widetilde{M}) + W \cdot G(\widetilde{M}) \tag{2}$$

where L stands for the local loss term. G stands for the global loss term. W is a weight adjusting the relative importance between L and G. The matching function $\widetilde{M}$ describes the estimated assignments from the nodes in the template graph to those in the scene graph. It is an estimation of the true matching function M . We use p_i to denote the ith node in the template, then the corresponding node of p_i in the scene can be represented as $\widetilde{M}(p_i)$. For the sake of brevity, we rewrite $\widetilde{M}(p_i)$ as q_j below.

Focusing on the specific terms in Eq. (2), the local loss term L is defined as:

$$L(\widetilde{M}) = \sum D_L(p_i, q_j) \tag{3}$$

where D_L is the estimation of local loss between each correspondence which can be defined as:

$$D_L(p_i, q_j) = \min dist(v_{p_i}, v_{q_j}) \tag{4}$$

Here, v_{p_i} and v_{q_j} denote the local features of p_i and q_j respectively. Many kinds of local features can be applied such as SIFT, SPINE [26]. The distance between two features is defined according to the chosen feature descriptor.

The global loss term in Eq. (2) is defined as:

$$G(\widetilde{M}) = \sum D_G(d_G(E^{p_i}, E^{q_j}); w^{p_i}, w^{q_j}) \tag{5}$$

where E^{p_i} is the set of the links whose end nodes include p_i. $d_G(E^{p_i}, E^{q_j})$ is the evaluation of the global loss between p_i and q_j which can be rewritten as:

$$d_G(E^{p_i}, E^{q_j}) = \sum_{p_k \in V_1 \backslash p_i} \min dist(e_{p_k}^{p_i}, e_{\widetilde{M}(p_k)}^{q_j}) \tag{6}$$

in which the $e_{p_k}^{p_i}$ denotes the features of the link from p_i to p_k. $e_{\widetilde{M}(p_k)}^{q_j}$ is the corresponding features in the scene graph. w^{p_i} is a weight setting term which is calculated based on the variances of the global features of p_i. D_G calculates the global loss while taking the corresponding weights into consideration.

Let I_T and I_S denote the template and the scene image respectively. Suppose their graph models are given: $\mathrm{g}_1 = \{V_1, E_1, C_1\}$ and $\mathrm{g}_2 = \{V_2, E_2, C_2\}$. Two affinity matrices $K_V \in R^{n_1 \times n_2}$ and $K_E \in R^{m_1 \times m_2}$ can be calculated according to the similarities of every pair of nodes and links respectively. To be more specific, $k_{ij}^V = D_L(v_{p_i}, v_{q_j})$ measures the similarity of the local features of p_i and q_j, and $k_{ij}^E = d_G(E^{p_i}, E^{q_j})$ measures the similarity of the global features of these two nodes.

B. Calculation Details

This subsection details the calculation of these functions mentioned above, including the matching function, the local loss term and the global loss term.

1) Matching Function: The matching function $\widetilde{M}$ is usually modeled as a set of binary variables. Similarly, we define a binary variable matrix $X = \{0,1\}^{n_t \times n_s}$ to represent the matching result of $\widetilde{M}$. $X_{ij} = 1$ denotes that the ith template node and the jth scene node are corresponded, and 0 otherwise. In other words, X_{ij} is 1 if and only if $\widetilde{M}(p_i) = q_j$. We use the widely applied one-to-one constrain which makes each row of X contain exactly one 1, meaning every template node must be matched to exactly one node in the scene graph. As a result, n_s is larger than n_t in the matrix X.

2) Local Loss Term: In this paper, the shape context [27] is selected as our local feature. Shape context is a popular local feature which describes the distribution of all remaining points from one point's view. It gets a description about the global situation which provides it with good robustness and discriminative ability.

Intuitively, considering a point p_i in an image, we draw bins around it and all the other points must locate in one of these bins. Then, the numbers of key points located in every bin can form the shape context which is a distribution describing the local feature of p_i. According to the original method, we represent the shape contexts as histograms and use the χ^2 test statistic:

$$k_{ij}^V = D_L(p_i, q_j) = \frac{1}{2} \sum_{h=1}^{H} \frac{[b_{p_i}(h) - b_{q_j}(h)]^2}{b_{p_i}(h) + b_{q_j}(h)} \tag{7}$$

where $b_{p_i}(h)$ and $b_{q_j}(h)$ denote the H-bin normalized histogram at p_i and q_j respectively.

Given the K_V, the local loss term L can be calculated as:

$$L(X) = \sum D_L(p_i, q_j) = tr({K_V}^T X) = \sum_{i=1}^{n_t} \sum_{j=1}^{n_s} k_{ij}^V X_{ij} \tag{8}$$

where n_s and n_t are the number of points in the template and the scene graph respectively.

3) Global Loss Term: The global loss term can be directly calculated as the sum of the global loss of ever correspondence. The global loss of one correspondence is defined as the square sum of the differences between corresponding geodesic-like distances of these two nodes. Specifically, the $e_{p_k}^{p_i}$ is obtained according to the geodesic-like distance between p_i and p_h which is denoted as $l_{p_k}^{p_i}$. Hence, the global loss of a correspondence can be calculated as:

$$k_{ij}^E = d_G(E^{p_i}, E^{q_j}) = \sum_{p_h \in V_1 \backslash p_i} [l_{p_h}^{p_i} - l_{\widetilde{M}(p_h)}^{q_j}]^2 \tag{9}$$

Then, the function D_G is calculated as:

$$D_G = \sum_{p_h \in V_1 \backslash p_i} w_{p_h}^{p_i} w_{\widetilde{M}(p_h)}^{q_j} [l_{p_h}^{p_i} - l_{\widetilde{M}(p_h)}^{q_j}]^2 \tag{10}$$

where $w_{p_h}^{p_i}$ denotes the weight of our confidence on the geodesic-like distance between p_i and p_h in the template graph. $w_{\widetilde{M}(p_h)}^{q_j}$ is the corresponding weight in the scene graph.

Finally, the global loss term G which is defined in Eq. (5) is calculated as:

$$G(\widetilde{M}) = \sum_{i=1}^{n_t} \sum_{p_h \in V_1 \backslash p_i} w_{p_h}^{p_i} w_{\widetilde{M}(p_h)}^{q_j} [l_{p_h}^{p_i} - l_{\widetilde{M}(p_h)}^{q_j}]^2 \tag{11}$$

which can be rewritten as:

$$G(X) = \sum_{i=1}^{n_t} \sum_{j=1}^{n_s} X_{ij} \sum_{p_h \in V_1 \backslash p_i} w_{p_h}^{p_i} w_{\widetilde{M}(p_h)}^{q_j} [l_{p_h}^{p_i} - l_{\widetilde{M}(p_h)}^{q_j}]^2 \tag{12}$$

C. Optimization Scheme

The graph matching problem is usually a NP-hard problem [14]. Someone searches the exponential solution space using brute methods such as the branch and bound method whose worst and average complexities are exponential. But this method is only available for small scale problems, otherwise it is too inefficiency. To speed up the solving process, researchers usually use approximation methods to get an approximate solution.

We use the IPFP algorithm [28] which can approximately solve the corresponding problem. It is able to approximately solve problems with the form:

$$\begin{gathered} \min_x \left\| x^T H x \right\|_F \\ s.t.\ Ax = \mathbf{1},\ x^T A = \mathbf{1}^T,\ x \in \{0,1\}^{n_t n_s} \end{gathered} \tag{13}$$

where x is formed by concatenating column vectors of X, and H is the affine matrix. Let $H_{ij;ab}$ be the entry of H at the $(i-1)n_t + j$th row and $(a-1)n_t + b$th column which measures the total loss if we match p_i to q_j and p_a to q_b at the same time. In this paper, assuming a pair of points (i, a) in I_T and (j, b) in I_S, $H_{ij;ab}$ is calculated as:

$$H_{ij;ab} = D_L(i,j) + D_L(a,b) + w_a^i w_b^j \min dist\left(e_a^i, e_b^j\right) \tag{14}$$

Note that the function D_L and $\min dist\left(e_a^i, e_b^j\right)$ are both symmetrical. Therefore H is a symmetrical $n_t n_s \times n_t n_s$ matrix.

V. Experimental Results

In order to give our algorithm a complete performance evaluation, we test it in both articulated deformation matching tasks and rigid ones.

Since the way to construct the graph model plays a central roll in our algorithm, we compare our algorithm with graph matching algorithms where the graph models are constructed by conventional methods. The compared methods include the Delaunay triangulation (Del), the k nearest neighbors (Knei) and the ϵ neighbors (Enei). The Del triangulates the point set such that no point is inside the circumcircle of any triangle. In the meanwhile, the Del maximizes the minimum angle of all the angles of the triangles. The Knei links each point with its k nearest points. The Enei links every pair of points whose distance is smaller than ϵ. On these graph models, the distances between points are calculated as the lengths of the shortest path connecting points. The distances are used in the matching problem as features. Besides, all the methods apply the IPFP [28] to solve the quadratic assignment problem.

All the methods are evaluated according to their matching precisions. The matching precision is defined as the ratio between the number of correct correspondences of inliers and the total number of inliers.

A. Articulated Deformation

Articulated deformation is a kind of non-rigid deformations commonly observed. Articulated objects include human bodies, multiple tools and lots of robots. They change their shapes by rotating around their joints.

Since the geodesic-like feature is invariant under the articulated deformations, the geodesic graph model built by our algorithm is suitable for this kind of deformations naturally. To show the advantage of our algorithm in articulated deformations, we choose the tools dataset to test on. In this dataset, there are 7 kinds of different tools with 5 different deformed shapes in each kind.

To quantitatively analysis our model, we manually select 18 to 30 key points from each image and label their corresponding ground truth. Fig. 2 shows instances of the key points manually labeled on different kinds of tools.

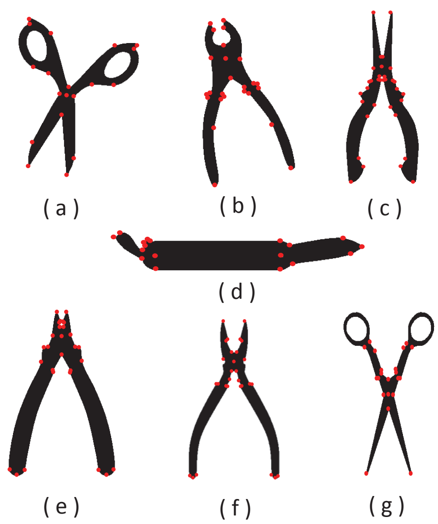

Fig. 2. Key points manually labeled in our experiment on the tools dataset.

Hence, we can calculate the matching precision of our algorithm by matching every pair of images which contain the same tool, since every key point has a unique correspondence in each of the rest images. Table. I shows the matching results, where the tool indexes are indicated in Fig. 2.

In this experiment, our algorithm obtains the best precisions in 6 out of 7 subset tests among all the algorithms tested.

In summary, this experiment shows obvious advantage of GGM in representing the spatial relations among key points on object with articulated deformations.

TABLE I
MATCHING PRECISIONS ON THE TOOLS DATASET (UNIT: PERCENT)

Tool indexes	a	b	c	d	e	f	g
Del	**87.9**	77.2	59.7	87.9	44.3	42.1	56.1
Knei	75.3	69.6	66.4	87.4	27.9	50.3	55.5
Enei	82.6	76.0	57.3	87.4	37.5	44.1	54.5
Ours	87.4	**83.6**	**83.3**	**88.7**	**82.9**	**89.0**	**82.1**

B. Rigid Deformation

Rigid deformation matching problem is the fundamental of all the other matching problems. Rigid deformation is commonly used to approximate more complex deformations.

To show the ability of our method in rigid deformations, we choose the CMU house/hotel motion data as our test dataset. This dataset concludes 111 frames of house and 101 frames of hotel with 30 key points labeled in each frame. We create test pairs using two frames separated by a specific in-between frames. As the separation between frames increases, the degree of deformations increases. Fig. 3 shows a pair of matched images in the house subset.

In this experiment, we also compare our algorithm with the PLNS algorithm. [15] which tackles the non-rigid point matching problem using an idea proposed in a manifold learning algorithm [15]. We use the code of PLNS algorithm which is implemented by its authors and apply the default parameters. For a fair comparison, the PLNS algorithm runs for one iteration. The matching results under different separation between frames are shown in Fig. 4.

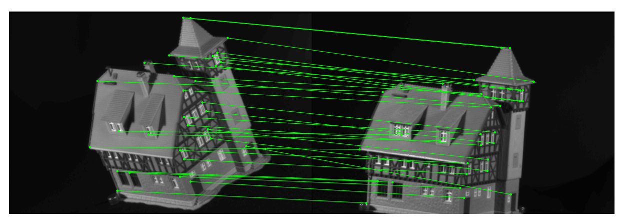

Fig. 3. Typical matching instance of our algorithm on house dataset.

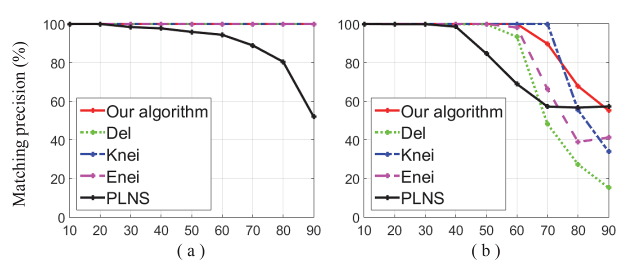

Fig. 4. (a) shows the matching precisions on the house sub-dataset; (b) shows the matching precisions on the hotel sub-dataset.

As shown in the results, our algorithm is the best or the second best in different separations between frames.

VI. Conclusion

In this paper, we introduce a new kind of graph model which is referred to as the GGM. The geometric relations among key points are maintained in the matching process by keeping the same geodesic-like distances in the template GGM with those in the scene GGM. Different from common methods, the GGM is built on the basis of the minimal spanning tree of points which is more flexible to the non-rigid, especially articulated, deformations. In addition, we introduce a discretization process and a weight setting process which can handle the drifting of points effectively. By comparative experiments, we demonstrate that this algorithm can provide robust structural features which are invariant to non-rigid deformations.

This algorithm is easy to implement. And the computational complexity mainly derives from the procedure of graph building. The complexity of this procedure is $O\left({n_t}^3 + {n_s}^3\right)$.

The parameters of this algorithm are set manually by experience. We want to build a mechanism to automatically set these parameters depending on the different conditions of each key point and the task handled. Meanwhile, this algorithm is easily infected by outliers. So, our next researching goal includes making it more robust to outliers.

Acknowledgment

The authors would like to thank Tang Tang for useful discussions.

References

[1] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *Int. J. Pattern Recogn.*, vol. 18, no. 03, pp. 265–298, 2004.

[2] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2000, pp. 44–51.

[3] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv. (CSUR)*, vol. 24, no. 4, pp. 325–376, 1992.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[5] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus," in *Eur. Conf. Comput. Vis.* Springer, 2008, pp. 500–513.

[6] A. D. Cross and E. R. Hancock, "Graph matching with a dual-step em algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1236–1253, 1998.

[7] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 377–388, 1996.

[8] S. Umeyama, "An eigendecomposition approach to weighted graph matching problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 5, pp. 695–703, 1988.

[9] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, 2005, pp. 1482–1489.

[10] J. A. Schnabel, D. Rueckert, M. Quist, J. M. Blackall, A. D. Castellano-Smith, T. Hartkens, G. P. Penney, W. A. Hall, H. Liu, C. L. Truwit *et al.*, "A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations," in *Medical Image Computing and Computer-Assisted Intervention*, 2001, pp. 573–581.

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

[12] H. Jiang, M. S. Drew, and Z.-N. Li, "Matching by linear programming and successive convexification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 959–975, 2007.

[13] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2383–2395, 2011.

[14] L. Torresani, V. Kolmogorov, and C. Rother, "A dual decomposition approach to feature correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 259–271, 2013.

[15] Y. Zheng and D. Doermann, "Robust point matching for nonrigid shapes by preserving local neighborhood structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 643–649, 2006.

[16] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[17] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[18] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[19] H. Li, X. Huang, and L. He, "Object matching using a locally affine invariant and linear programming techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 411–424, 2013.

[20] X. Niyogi, "Locality preserving projections," in *Proc. Conf. Neural Information Processing Systems*, vol. 16, 2004, p. 153.

[21] A. Elad and R. Kimmel, "On bending invariant signatures for surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1285–1295, 2003.

[22] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, 2007.

[23] ——, "Using the inner-distance for classification of articulated shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2005, pp. 719–726.

[24] S. Biswas, G. Aggarwal, and R. Chellappa, "Efficient indexing for articulation invariant shape matching and retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[25] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, "Learning graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1048–1058, 2009.

[26] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

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

[28] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and map inference," in *Proc. Conf. Neural Information Processing Systems*, 2009, pp. 1114–1122.