



# A game-theoretic approach to sub-vertex registration<sup>☆</sup>



Rong Wang<sup>a,b,\*</sup>, Zheng Geng<sup>a</sup>, Xuan Cao<sup>a,b</sup>, Renjing Pei<sup>a,b</sup>, Xiangbing Meng<sup>a,b</sup>

<sup>a</sup> Institute of Automation, Chinese Academy of Sciences, No. 95, East Zhongguancun Road, Haidian District, Beijing 100190, China

<sup>b</sup> University of Chinese Academy of Sciences, China

## ARTICLE INFO

### Article history:

Received 6 January 2016

Available online 20 June 2016

### Keywords:

Rigid surface registration

Evolutionary game theory

Sub-vertex correspondence

High noise

Low resolution

## ABSTRACT

Surface registration is a fundamental technique in computer vision. Typically, it consists of two basic steps: a coarse registration, followed by a fine registration. A novel game-theoretic matching (GTM) algorithm was proposed recently to directly obtain a fine surface registration in a single step. The main idea of GTM is to cast the selection of point correspondences in an evolutionary game framework. However, GTM fails easily due to the lack of correct correspondences if model surface is in low resolution. To tackle this problem, in this paper, we propose a game-theoretic approach to establish sub-vertex correspondences. A new way to construct the payoff function to solve one-to-many matches is introduced. The weight population after evolving from replicator dynamics is used to compute the corresponding sub-vertex. The effectiveness of our proposed method is verified by extensive experiments. Though comparing with GTM and state-of-the-art Super4PCS, our method is accurate, efficient and especially robust in extreme situations of high noise and low resolutions. Finally, the sensitivity and the limitations of our method are discussed.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

Surface registration is one of the key problems in computer vision. It is a fundamental technique for 3D surface reconstruction, 3D object recognition and augmented reality [1]. Because rigid registration is the foundation of more advanced non-rigid registration, it has attracted considerable research interest in recent years.

The goal of surface registration is to estimate a 3D rigid transformation between two surfaces, so that they can be placed together under a minimal distance measure. The transformation is denoted as a  $3 \times 3$  rotation matrix  $R$  and a  $3 \times 1$  translation vector  $t$ . The two surfaces involving in registration can be simply represented as point sets. We name the matching target as model, while the matching source as data. An example of surface registration is shown in Fig. 1.

Surface registration is generally carried out in two steps: a coarse step and then a fine step. In both steps, correspondences are first found and then transformation is obtained in a closed-form solution. Since mature approaches such as the singular value decomposition (SVD) approach [2] or the quaternion-based approach

[3] can be used to get transformation, establishing good correspondences is critical for a successful surface registration.

A recently proposed game-theoretic matching (GTM) [4] does not need coarse matching result as initialization. This algorithm outperforms most coarse registration algorithms and works almost equally well with traditional fine registration methods. One major disadvantage of this algorithm, however, is that correct correspondences cannot be found on surfaces of different resolutions. To overcome this difficulty, we propose herein a modified game-theoretic approach to sub-vertex registration. The sub-vertices can be seen as the interpolation of the original vertices. The weights are obtained through evolutionary game. As detailed in Section 4, our method can work well in extreme situations of high noise and low resolutions.

The focus of this paper is on establishing sub-vertex correspondences. The rest of the paper is organized as follows. After this introduction, we briefly review previous work as well as GTM in Sections 2 and 3. Then, the motivation behind our method and the details of it are presented in Section 4. Next, experimental results are given and compared with GTM and state-of-the-art technique in Section 5. Finally, some conclusions are drawn in Section 6.

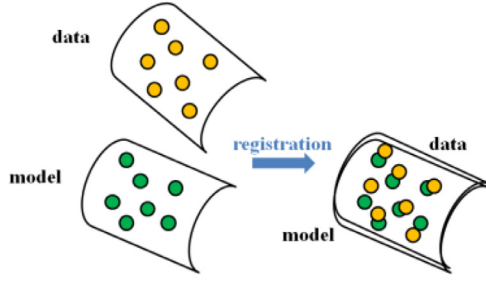
## 2. Literature review

The goal of rigid surface registration is to minimize an objective function  $\sum_{i=1}^N \|M_i - RD_i - t\|^2$  [3], where  $\{M_i, i = 1, \dots, m\}$  and  $\{D_i, i = 1, \dots, n\}$  are model and data vertices respectively and  $N$  is

<sup>☆</sup> This paper has been recommended for acceptance by Dr. D. Coeurjolly.

\* Corresponding author at: Institute of Automation, Chinese Academy of Sciences, No. 95, East Zhongguancun Road, Haidian District, Beijing 100190, China. Tel.: +86 15010793916.

E-mail address: [wangrong2013@ia.ac.cn](mailto:wangrong2013@ia.ac.cn) (R. Wang).



**Fig. 1.** An example of surface registration. Points on data and model surfaces are used for registration.

the number of established correspondences. The optimization is a two-stage problem of correspondence estimation and rigid transformation calculation.

As we have mentioned above, registration is usually implemented through the coarse and the fine step. Díez et al. [5] provided a good summary of coarse registration. Local feature based algorithms are often used [6–8] in the coarse step because of their convenience and low computational cost. Features specifically designed for GTM are presented in [9]. Guo et al. [10] gave a recent survey of existing local surface features for surface matching.

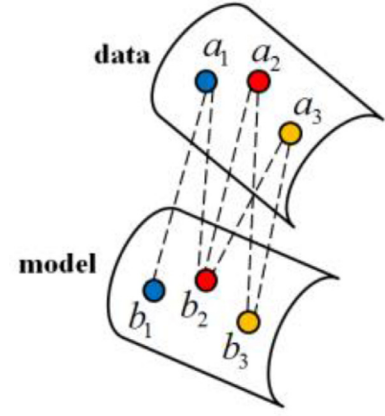
Fine registration is a refining process after the coarse step. The most popular algorithm in fine registration is Iterative Closest Point (ICP) [3] due to its simplicity and high accuracy. Bellekens et al. [11] offered a recent survey on ICP and its variants. ICP has been evolving over time and it has got improvements in speed, accuracy and robustness. Rusinkiewicz and Levoy [12] introduced a detailed classification of ICP variants. Since original objective function treats each correspondence equally, the influence of low-quality correspondences may result in poor registration. Therefore, assigning different weights for correspondences was proposed [12,13] and then the objective function changes into  $\sum_{i=1}^N w_i \|M_i - RD_i - t\|^2$ . The weight reflects the certainty of each established correspondence being a correct match. Recently, a sparse ICP [14] was proposed to robustly handle registration with outliers. Although ICP is popular, it heavily depends on the coarse registration results and easily converges to local minimum.

Global registration methods attract much attention recently since the two surfaces can be placed in arbitrary initial poses. Among them, game-theoretic matching (GTM) [4] is proposed to achieve the precision of fine registration without initial transformation estimation. It efficiently establishes robust one-to-one correspondences which mean that each data point takes at most one model point as its corresponding point. The state-of-the-art Super4PCS [15] finds transformation between two surfaces using coplanar sets of 4 points and can achieve outstanding results even when fine registration method fails.

Although GTM can achieve promising results in general occasions, the one-to-one matching limits its performance in the situation where surfaces have much different resolutions [16]. This is the main motivation for our method. To the best of our knowledge, this problem has not been explored before. In this paper, we proposed, implemented and verified the modified game-theoretic approach to sub-vertex registration in order to tackle the dilemma of GTM. Our method is also compared with both GTM and state-of-the-art Super4PCS.

### 3. A brief review of GTM

As our method is the modification of GTM, a brief review of GTM [4] is necessary.



**Fig. 2.** An illustration of game-theoretic matching (GTM).

#### 3.1. Game-theoretic matching

In the two-player's non-cooperative game,  $S = \{1, \dots, n\}$  is a set of available pure strategies and  $\Pi : S \times S \rightarrow R$  is a payoff matrix, where  $\pi_{ij} = \Pi(i, j)$  indicates the benefit that a player playing strategy  $i$  gains when another player adopting strategy  $j$ . A mixed strategy is a probability distribution  $\mathbf{x} \in \Delta$  denoted as a column vector over the strategies  $S$ , where  $\Delta$  is defined in Eq. (1).  $x_i$  is the distribution proportion of strategy  $i$  and it is also the probability that the player will choose that strategy.

$$\Delta = \left\{ \mathbf{x} : x_i \geq 0, \forall i \in 1 \dots n, \text{ and } \sum_{i=1}^n x_i = 1 \right\} \quad (1)$$

The expected payoff obtained by a player playing mixed strategy  $\mathbf{y} \in \Delta$  against another player adopting  $\mathbf{x} \in \Delta$  is  $\mathbf{y}^T \Pi \mathbf{x}$ . The most important concept in game theory is Nash equilibria. A strategy pair  $(\mathbf{x}, \mathbf{y})$  is a Nash equilibrium if  $\mathbf{x}$  is the best reply to  $\mathbf{y}$ , and at the same time,  $\mathbf{y}$  is the best reply to  $\mathbf{x}$ . Since we only consider symmetric games in which the two players are indistinguishable, only  $(\mathbf{x}, \mathbf{x})$  are of interest [17]. Therefore, a strategy  $\mathbf{x}$  is a Nash equilibrium if it is the best reply to itself. This means  $\forall \mathbf{y} \in \Delta, \mathbf{x}^T \Pi \mathbf{x} \geq \mathbf{y}^T \Pi \mathbf{x}$ . Further, a strategy  $\mathbf{x}$  is called an evolutionary stable strategy (ESS) if it is first a Nash equilibrium and then robust to a small perturbation of itself [18]. As the expected payoff of the entire population is  $\mathbf{x}^T \Pi \mathbf{x}$  [17], evolutionary game can be seen as an optimization problem defined in Eq. (2).

$$\max_{\mathbf{x} \geq 0} \mathbf{x}^T \Pi \mathbf{x}, \text{ s.t. } \mathbf{x}^T \mathbf{1} = 1 \quad (2)$$

We use Fig. 2 to illustrate the process of GTM. The dashed lines represent the possible matching candidates. Imagine two players participating in the game and they pick up matching candidates  $(a_1, b_1)$  and  $(a_2, b_2)$  respectively. If the two candidates are compatible with each other, then both players receive a high score, otherwise the score will be low. The two candidates are compatible if they satisfy the rigidity constraint. In this example,  $(a_1, b_1)$  and  $(a_2, b_2)$  are compatible if the Euclidean distances  $\|a_1 - a_2\|$  and  $\|b_1 - b_2\|$  are equal. In order to gain high scores, each player tends to pick matching candidates that are compatible with the other player's choice. For more knowledge of GTM, please refer to [16] and [19].

#### 3.2. Matching candidates, payoff function and evolution

Matching candidates can be established between any two vertices on data and model surfaces, which are quite a lot. They will definitely take large memory size while finally most of them will

be dropped. Therefore, matching candidates in GTM are established between feature points through descriptors. For each data feature point, matching candidates are established between it and its several similar model feature points.

Payoff function takes a pair of matching candidates as input. The definition of payoff function follows the rigidity constraint. The value of payoff function is high if two matching candidates are compatible with each other. For example, given a pair of strategies  $(a_1, b_1)$  and  $(a_2, b_2)$ , where  $a_1, a_2$  are on data and  $b_1, b_2$  are on model, according to the principle explained above, we define the payoff function as in Eq. (3).  $\lambda$  is a scalar, which is set by the user. The value of payoff function corresponds to an element in the payoff matrix. Since GTM looks for one-to-one correspondences, the main diagonal of the payoff matrix should be zeros and matching candidates sharing the same data or model points should also be assigned zeros [4]. Then, establishing correspondences can be seen as optimizing problem defined in Eq. (2).

The replicator dynamics [17] shown in Eq. (4) is used to solve the optimization problem and evolve a mixed strategy  $\mathbf{x}$  to a stable state. Matching candidates with relatively high values in the final  $\mathbf{x}$  are more likely to be correct correspondences. GTM then uses these correspondences and their weights, which are the values in  $\mathbf{x}$ , to compute transformation. The replicator dynamics guarantees the evolutionary game to converge to a Nash equilibrium [18] which is a local maximum [4]. However, the construction of the payoff matrix  $\Pi$  contains global information which enables GTM to have better performance.

$$\pi((a_1, b_1), (a_2, b_2)) = \exp(-\lambda(\|a_1 - a_2\| - \|b_1 - b_2\|)) \quad (3)$$

$$x_i(t+1) = x_i(t) \frac{(\Pi \mathbf{x}(t))_i}{\mathbf{x}(t)^T \Pi \mathbf{x}(t)} \quad (4)$$

#### 4. The modified method

In this article, we propose a modification of GTM to sub-vertex registration. We will elaborate the motivation behind our method and provide details of it in the following subsections.

##### 4.1. Motivation

Traditional surface registration makes an assumption that each data point has an ideal corresponding model point. However, it is often difficult to obtain the ideal correspondences because the point sets for registration are acquired with different techniques in practice [20]. Some examples are presented below. 3D scanning operations cannot guarantee that the same physical point is captured in different views, which usually results in missing data. Point sets representing the same organ surface may be reconstructed from either 3D endoscope or preoperative CT segmentation in computer-aided surgery [21]. Or we need to align between a scan of a manufacturing product and its ground truth CAD model in digital quality inspection [22]. All the above registrations have the same problem that many data samples do not have an ideal corresponding model point.

In order to find the best match, interpolated points are used [23]. In this article, we apply a simple weighted combination strategy to get the interpolated points. In this way, the corresponding point of a data vertex is defined as the weighted combination of several model vertices rather than an individual model vertex. Apparently, the corresponding point obtained in this way is a “sub-vertex” which means that it may not be part of the original 3D vertices. The sub-vertex can provide more reliable corresponding relationship by weighting when the model surface is in high noise or low resolution.

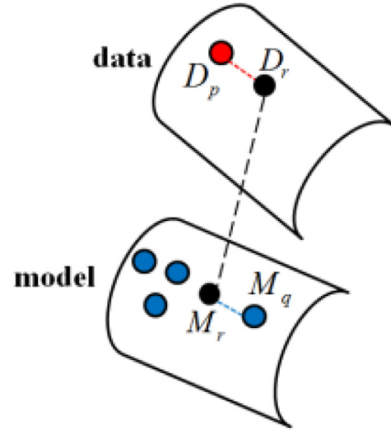


Fig. 3. An illustration of false combination.

An intuition to determine the combination weights is through the similarity between the model and the data points. There are two similarity measures we can use. As every point is assigned a feature descriptor, the similarity between two points is measured by the Euclidean distance between their descriptors. Another similarity measure is defined by rigidity constraint in which a reference point match is required. Combining the two similarity measures, the weight can be defined in Eq. (5).  $D_p$  is a data point and  $M_q$  is one of model points with similar descriptors to  $D_p$ .  $d_i$  is the descriptor vector associated to point  $i$ .  $(D_r, M_r)$  is the established reference point match which will be described later.

$$\text{weight}(M_q) = \exp(-\alpha \|d_{M_q} - d_{D_p}\|) \cdot \exp(-\beta (\|M_q - M_r\| - \|D_p - D_r\|)) \quad (5)$$

Although this definition of weight seems reasonable, it will bring some problems. For example, imagine the situation shown in Fig. 3. Black points correspond to the reference point match. Our aim is to find the best corresponding point to data point  $D_p$ . Model points with similar descriptors to  $D_p$  are denoted as blue points. If the weight is defined by Eq. (5), the point  $M_q$  will take a large weight in computing the sub-vertex due to the distance  $\|M_q - M_r\|$  is close to  $\|D_p - D_r\|$ . The large weight brought by  $M_q$  will make the sub-vertex deviate from the accurate corresponding point of  $D_p$  and result in poor correspondence.

To cope with the above problem, we use a game-theoretic approach to determine the weights. Our method can avoid the false combination shown in Fig. 3. The main idea is that the weight of every similar model point is not considered independently, but in pairwise. Taking Fig. 3 as an example,  $M_q$  can be seen as an outlier and eliminated through considering other blue points. More details of our method will be described in the next subsection.

##### 4.2. Implementation details

Our modified method aims at obtaining accurate sub-vertex correspondence. The entire pipeline of our method is shown in Fig. 4. It mainly consists of 4 steps: feature point and model candidates extraction, reference match establishment, sub-vertex calculation and robust correspondence revision. We will explain the details of our method step by step.

The first step is to detect feature points from the original surfaces and establish model candidates. Feature points are obtained by applying local feature detectors and then represented by feature descriptors. Model candidates for computing sub-vertex to each data feature point are selected as those model feature points that have similar descriptors to the data point.

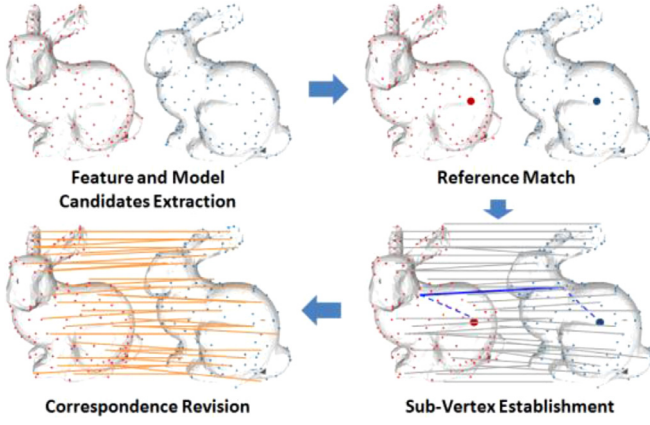


Fig. 4. The entire pipeline of our method.

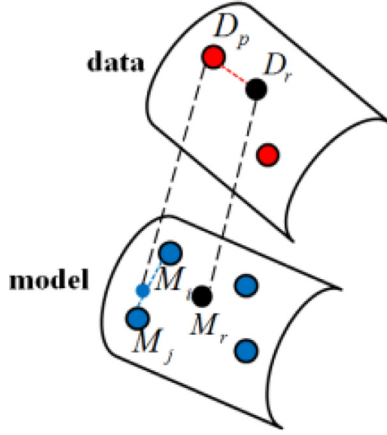


Fig. 5. An illustration of constructing payoff function in our method.

The second step is to find a reference point match. We use GTM to achieve this goal automatically. More specifically, we only establish one matching candidate in this GTM step between the most similar model feature point and each data feature point. The matching pair which has the largest weight after evolution is picked as reference. In this way, reference match can be obtained efficiently and less influenced by the limitations of GTM. Other strategies such as user specification can also be used to make a choice of a reference point match when GTM fails.

In the third step, we want to build corresponding sub-vertex to each data feature point. That is to say, for each data point, we need to compute weights for its previously established model candidates. This is seen as an optimization problem defined in Eq. (2), but with a different payoff matrix. The population  $\mathbf{x}$  in Eq. (2) is the degree of participation of each model candidate to construct the corresponding sub-vertex to a data feature point.

$$\pi(M_i, M_j) = \exp \left( -\lambda \left\| D_p - D_r \right\| - \left\| \frac{s_i M_i + s_j M_j}{s_i + s_j} - M_r \right\| \right) \quad (6)$$

$$s_i = \exp \left( -k \left\| d_{M_i} - d_{D_p} \right\| \right) \quad (7)$$

A novel payoff function is proposed to compute accurate sub-vertex to a data point. The construction of the payoff function complies with the rigidity constraint and is illustrated in Fig. 5. For a data point  $D_p$ , the payoff function is constructed among each pair of its model candidates. The payoff function for model candidates  $M_i$  and  $M_j$  is defined in Eq. (6), where the similarity function  $s_i$  is given in Eq. (7). The similarity function indicates that higher weight should be assigned to the model candidate

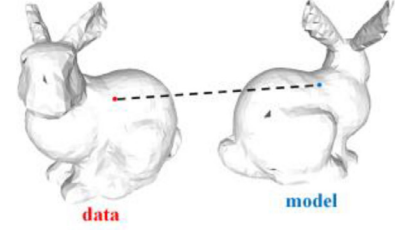


Fig. 6. An example of false correspondence.

which has a more similar feature descriptor to the data point when computing combined point  $(s_i M_i + s_j M_j) / (s_i + s_j)$ . When the distance between combined point and the reference model point  $M_r$  is close to the distance between a given data point  $D_p$  and the reference data point  $D_r$ , the payoff value will be high. Considering the case in Fig. 3 again, the low payoff values between  $M_q$  and other model candidates will make  $M_q$  disappear in the final population  $\mathbf{x}$ . Parameter  $\lambda$  in Eq. (6) is used to control the selectivity of the evolution. The shape of similarity function defined in Eq. (7) is controlled by the parameter  $k$ . It is worth noting that in our case the elements on the main diagonal of payoff matrix are not zeros, but defined the same as Eq. (6) with equal  $i$  and  $j$  indices. That is because we are trying to establish one-to-many matches rather than one-to-one matches in GTM. The final population vector  $\mathbf{x}$  after achieving equilibrium is used to compute the corresponding sub-vertex.

Finally, the previously obtained sub-vertex correspondences are taken as the input of GTM to discard false correspondences. This refining step cannot be omitted since the corresponding sub-vertex we obtained does not exist in original point set and may occasionally cause false correspondence as shown in Fig. 6.

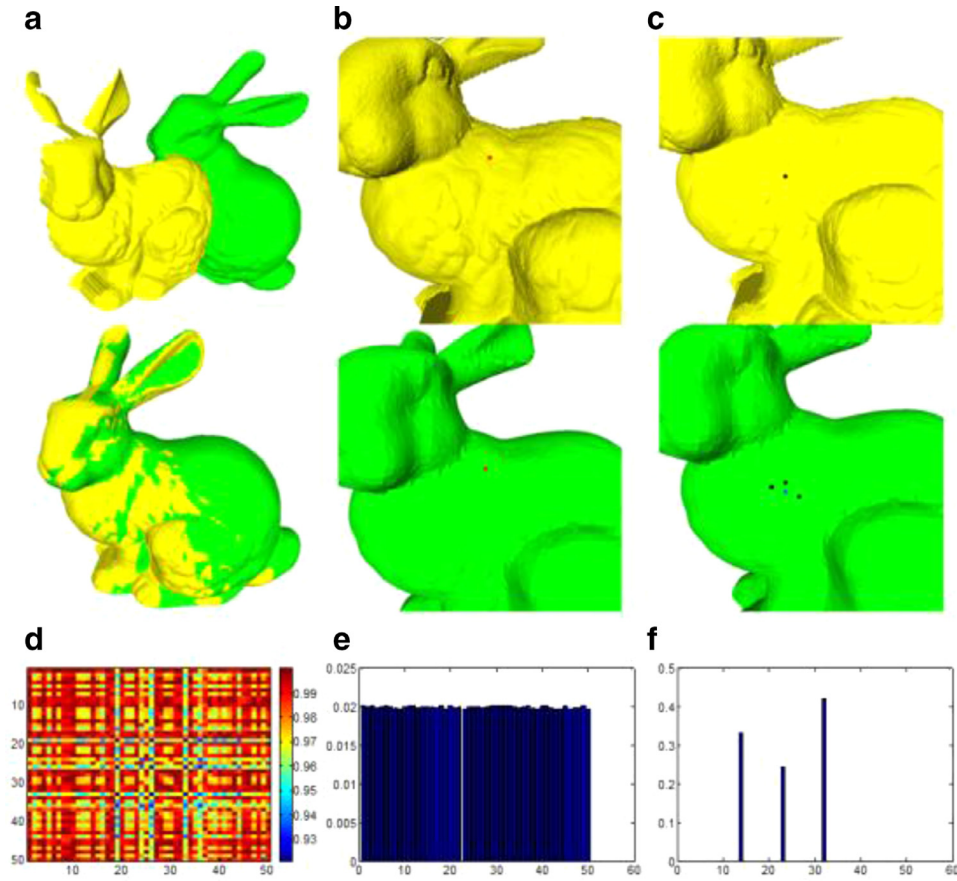
The main contributions of our method are summarized as follows: (1) we propose the weighted combination to obtain sub-vertex correspondences; (2) we compute the weights for constructing sub-vertex in the framework of evolutionary game.

#### 4.3. Registration example

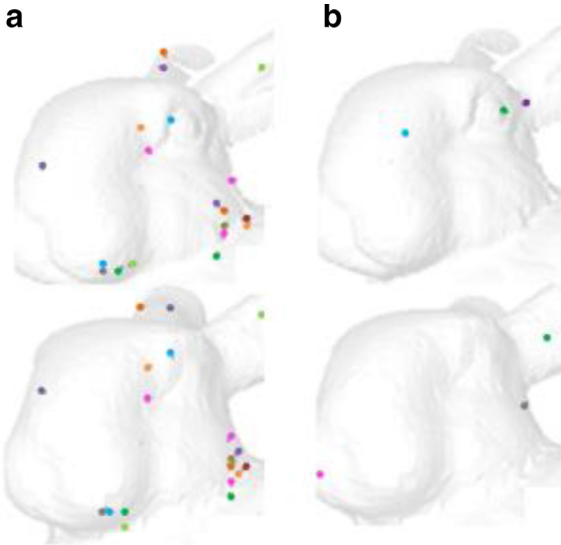
An example of our complete workflow is shown in Fig. 7. The data surface is a range scan in high resolution, while the model surface is a low resolution reconstruction result. Fig. 7(a) shows the initial and final poses of registration respectively. Although only points are used in our method, we draw the triangular meshes just for convenient visualization. Fig. 7(b) shows a reference match as red points located in data and model surfaces. In order to establish corresponding sub-vertex in evolutionary game, model candidates are the 50 most similar model points to the given data point. The data point is the black one in the top of Fig. 7(c). In Fig. 7(d), we visualize the payoff matrix constructed by Eqs. (6) and (7). The dimension of the payoff matrix is  $50 \times 50$ . In Fig. 7(e) and (f), we show weight population before and after evolution of the constructed game. We can see that after the game, there left only three model candidates which are the three black points in the bottom of Fig. 7(c). The corresponding sub-vertex is the weighted combination of the three black points and is labeled in blue. We can see that the synthetic blue point is more accurate as the corresponding point.

A part of the sub-vertex correspondences in our method are shown in Fig. 8(a). The points in the same color mean that they are correspondences. GTM is also applied to the same dataset and the correspondences are shown in Fig. 8(b). The results show that our method can find more accurate correspondences. More quantitative and qualitative study by extensive experiments and comparisons will be given in the experiment section.





**Fig. 7.** An example of our complete workflow. (a) The top is the initial surfaces for registration. The yellow surface denotes data and the green surface represents model. The bottom is the final registration result. (b) Reference points located on the two surfaces. (c) The top is a given data feature point; the bottom shows the corresponding sub-vertex (in blue) which is obtained as the weighted combination of the three model candidates (in black). (d) The payoff matrix in the process of computing corresponding sub-vertex to the given data point. (e) The initial weight population. (f) The weight population after convergence of evolutionary game. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).



**Fig. 8.** The visualization of the corresponding points on data (top) and model (bottom) surfaces. (a) The result of our method. (b) The result of GTM.

## 5. Experimental validations

To validate the efficacy of our method, we designed extensive experiments. First, we compared our method with GTM and state-

of-the-art Super4PCS under different levels of noise and subsampling rate. Then, some visual results are given. Last, sensitivity analysis is made and limitations are discussed.

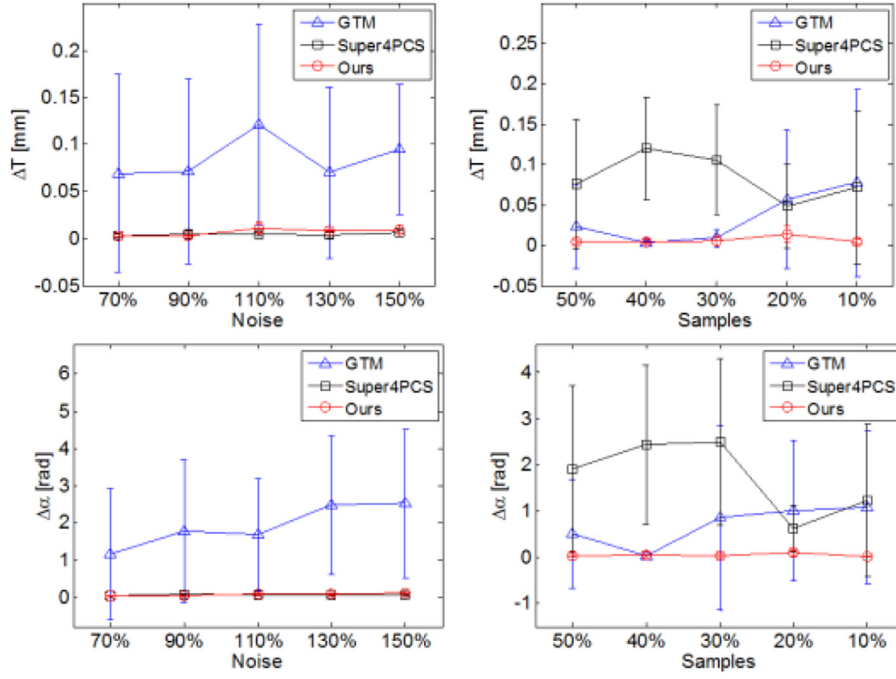
### 5.1. Experimental setup

In GTM and our method, a robust game-theoretic tool [17] was used to solve the problem defined in Eq. (2). We used the MeshDOG [24] detector to find feature points and the Spin Image descriptor [25] to represent each feature point. These methods were implemented in MATLAB on Windows. The state-of-the-art Super4PCS [15] is also compared in our experiments. The algorithm was implemented in C++ on Linux.

The Stanford 3D scanning repository [26] was used for our validations. We chose the Bunny, the Armadillo and the Dragon models which come from either 3D scanning or reconstruction. A partial scan is a range image in high resolution. Fully-reconstructed models are obtained in different resolutions.

### 5.2. Comparisons under different noise and subsampling rates

In this part, the model points were the known rigidly transformed copies of the data points. Different levels of noise and subsampling rates were performed on the model points. The noise obeyed Gaussian distribution, and its level was the ratio between standard deviation and the average edge length of the model. Sub-sampling was randomly carried out, and it was represented by the ratio between the point number of the subsampled model and the



**Fig. 9.** Comparison of the translation (row 1) and rotation (row 2) error of GTM, Super4PCS and our method under different noise (column 1) and subsampling rates (column 2).

original one. Tests for each different condition were repeated 6 times over the range images.

In Fig. 9 we report the comparison of the translation and rotation error of the three methods. Matching candidates in GTM were established by 3 model points that have the most similar descriptors to each data point. The translation error is defined as the Euclidean distance between the estimated translation vector and the ground truth. Rotation is represented as a  $3 \times 1$  vector whose elements are 3 Euler angles. Rotation error is the Euclidean distance between the Euler angles [27].

The results reflect the accuracy and robustness of each method. Small mean values and standard deviations indicate that the registration quality is high and stable. Super4PCS and our method are comparable in the noise condition, while GTM has bad performance. GTM performs worse because the reliability of local descriptors is heavily weakened by the noise condition. But our method can rectify this bad effect by sub-vertex. As for the conditions of different subsampling rates, our method outperforms both GTM and Super4PCS in accuracy and robustness.

Fig. 10 shows the comparison of the Root-Mean-Square (RMS) error, number of matches and elapsed time of GTM and our method. The RMS error is defined in Eq. (8), where  $D_i$  is a data point,  $C_i$  is its corresponding point and  $N$  is the total number of correspondences. The RMS error represents the closeness of correspondences after registration. In the condition of establishing a similar number of correspondences, the smaller RMS error means that the correspondences are more accurate. The matching pairs are deemed to be correspondences if their weights are higher than a fixed percentage of the largest weight. The execution time includes finding feature points and candidates, constructing payoff matrix and evolving via replicator dynamics.

$$\text{error} = \sqrt{\frac{\sum_{i=1}^N \|RD_i + t - C_i\|^2}{N}} \quad (8)$$

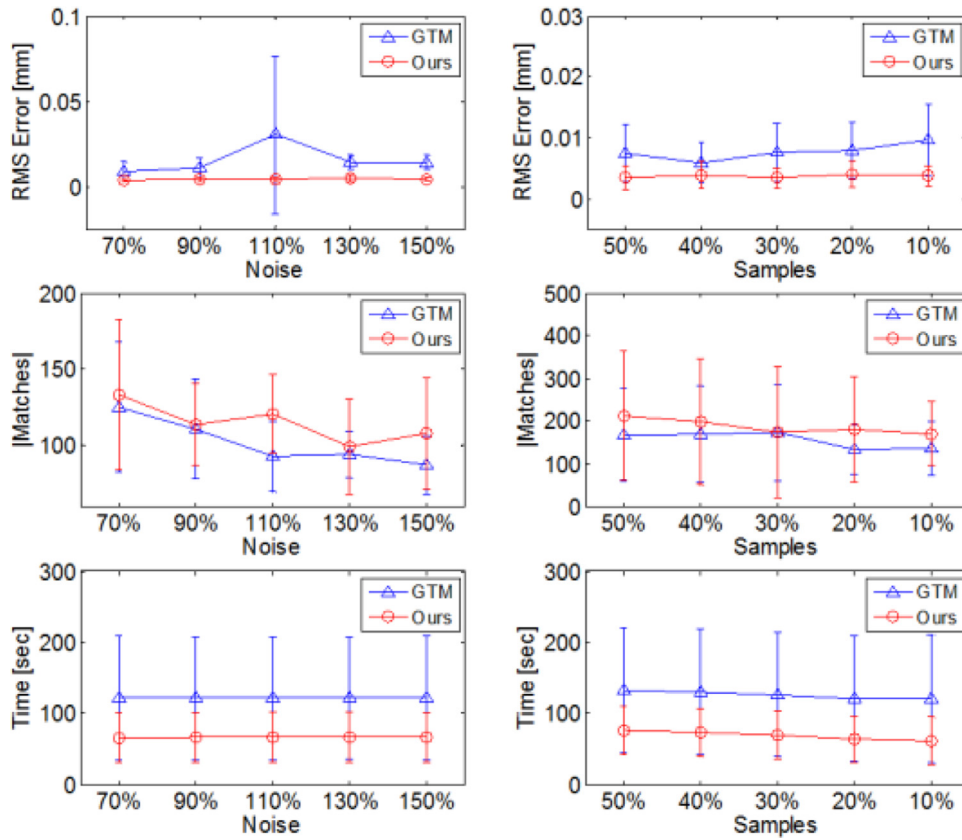
As the amount of established correspondences is slightly larger, we could conclude from the results that our method can find more

accurate correspondences. Our method also shows faster runtime which has a close relation with the computational complexity. Assume that we have  $n_D$  data feature points and  $n_M$  model feature points. As for GTM, matching candidates are established by 3 most similar model points toward each data feature point. Then computing the payoff matrix takes  $O((3n_D)^2)$  operations. Using replicator dynamics to evolve consumes  $O(3n_D)$  steps. Thus the overall complexity is  $O((3n_D)^2)$ . Suppose in our method, we use  $n_C$  model candidates to compute corresponding sub-vertex for a data point. Constructing payoff matrix needs  $O(n_C^2)$  steps and optimization takes  $O(n_C)$  steps. Since the number of data feature point is  $n_D$ , the complexity of the third step in our method is  $O(n_D \cdot n_C^2)$ . Both finding reference and the refining process apply GTM, which only establishes one matching candidate toward each data feature point, and take  $O(n_D^2)$  respectively. Therefore, the overall complexity is  $O(n_D \cdot n_C^2 + 2n_D^2)$ . Note that  $n_C \ll n_D$ , hence our method is faster. Since sub-vertex of one data point is computed independently, using GPU will further speed up our method.

Super4PCS [15] only establishes 4 correspondences to compute transformation and then evaluates the transformation by all the points rather than feature points. Therefore, we decide not to compare RMS error and the number of correspondences in it. Super4PCS has a linear time complexity in the number of data points, while from the above analysis our method runs in quadratic time but in the number of data feature points which is much smaller than data points. Besides, the quadratic time complexity is produced by constructing payoff matrix which can be easily accelerated in GPU. Since Super4PCS and the game-theoretic methods are implemented on different platforms, time comparison is not carried out in this paper. We may accelerate our method and carry on time comparison in the future research.

### 5.3. Visual comparison

In this subsection, we used range image as data surface and low resolution reconstruction as model surface. The registration results are shown in Fig. 11. We can see that GTM is easy to fail while Super4PCS and our method produce visually correct results. GTM



**Fig. 10.** Comparison of the RMS error (row 1), number of matches (row 2) and elapsed time (row 3) of GTM and our method under different noise (column 1) and subsampling rates (column 2).



**Fig. 11.** Registration results of surfaces with different resolutions using GTM (column 1), Super4PCS (column 2) and our method (column 3).

fails because in such situations adequate correct correspondences could not be established point-to-point. Super4PCS has better performance because it is quite easy to ensure 4 correct correspondences in the condition of different resolutions. However, visual

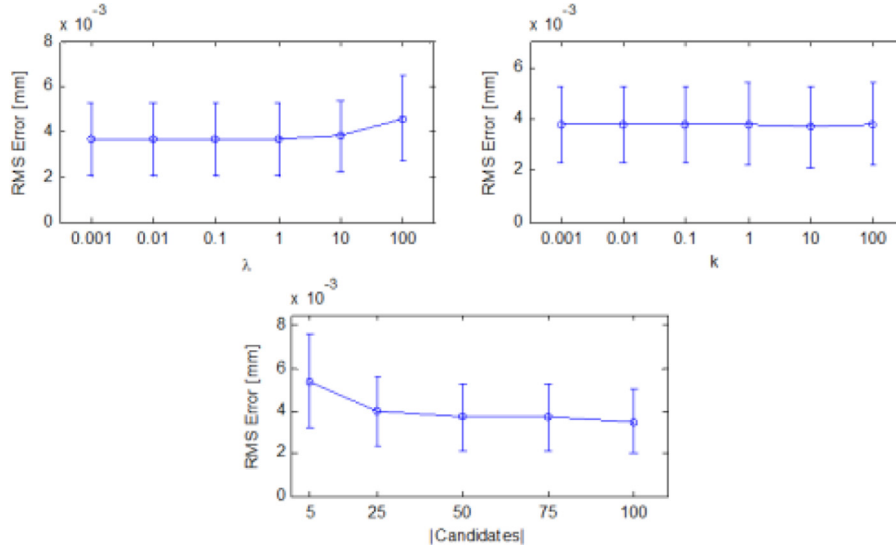
error (for example the ears of the bunny) can be observed because small numbers of correspondences have limited ability to find accurate transformation. Our method performs well because sub-vertex can provide more reliable corresponding relationship when model surface is in significantly low resolution.

#### 5.4. Sensitivity analysis

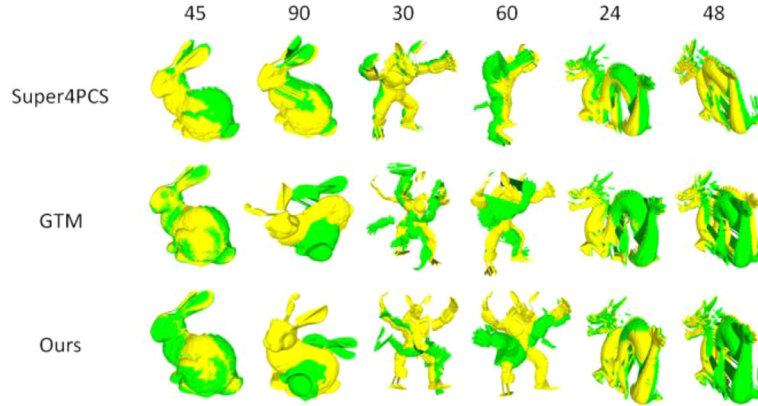
Our method basically contains 3 parameters:  $\lambda$  in payoff function,  $k$  in similarity function and the number of model candidates to establish sub-vertex to a given data point. The results of RMS error with the variation of each parameter are shown in Fig. 12. The results show that our method does not heavily rely on its parameters. Therefore, our method is robust and easy to apply.

#### 5.5. Limitations

Both GTM and our method are not applicable to featureless surfaces since they rely on local descriptors when establishing candidates. However, Super4PCS is not limited by this condition. We then discuss the influence of overlap on these methods through partial matching. The registration results with angle difference between the two scans are shown in Fig. 13. GTM and our method can align surfaces which are not 100% overlap. However, GTM is easy to fail with small overlap even if we discard matching pair when the distance between data feature point and its most similar model feature point is larger than a threshold. And what is more, reference point match in our method cannot be established correctly using GTM which leads to the failure of our method. The reason is that under the small overlap, candidates established by local descriptors consist of a lot of false matching pairs which are difficult to refine by GTM



**Fig. 12.** Sensitivity analysis of our method with respect to some basic parameters. The test has been performed using all the 3 picked models in the conditions of 70% noise and 50% subsampling rate.



**Fig. 13.** Comparison results for partial matching of Super4PCS (row 1), GTM (row 2) and our method (row 3). The digits on the top indicate the angles between the data and the model range scans. The larger angle means that there is small overlap between the two scans.

or our method. The Armadillo model is even harder for correct registration because its nearly symmetrical structure further hinders the effectiveness of the local descriptors. Therefore, featureless surface and small overlap are the main limitations of our method.

## 6. Conclusions

In this article, we propose a modified method for rigid surface registration in challenging surface quality conditions. Different from traditional registration methods, the proposed method aims at establishing sub-vertex correspondence in the framework of evolutionary game.

Our method has been tested on multiple public datasets, and compared with GTM and state-of-the-art Super4PCS. The quantitative assessment demonstrates improvements of our method in accuracy, efficiency and robustness. Especially, our method is able to obtain good registration quality even in some extreme situations of high noise and low resolutions. We also performed sensitivity analysis and found our method is insensitive to parameter disturbance. Finally, we discussed the limitations of our method.

## Acknowledgments

This work is supported by the National High-tech R&D Program of Institute of Automation, Chinese Academy of Sciences, grant 2012AA011903 and grant 2015AA015905.

## References

- [1] J. Geng, Structured-light 3D surface imaging: a tutorial, *Adv. Opt. Photon.* 3 (2011) 128–160.
- [2] K.S. Arun, et al., Least-squares fitting of two 3-d point sets, *IEEE Trans. Pattern Anal. Mach. Intell.* 9 (1987) 698–700.
- [3] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (1992) 239–256.
- [4] A. Albarelli, et al., A game-theoretic approach to fine surface registration without initial motion estimation, in: *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2010, pp. 430–437.
- [5] Y. Diez, et al., A qualitative review on 3d coarse registration methods, *ACM Comput. Surv. (CSUR)* 47 (2015) 1–36.
- [6] A.S. Mian, et al., A novel representation and feature matching algorithm for automatic pairwise registration of range images, *Int. J. Comput. Vis.* 66 (2006) 19–40.
- [7] J. Novatnack, K. Nishino, Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images, in: *Proceedings of the European Conference on Computer Vision*, 2008, pp. 440–453.
- [8] R.B. Rusu, et al., Fast point feature histogram (fpfh) for 3d registration, in: *Proceedings of IEEE Conference on Robotics and Automation*, 2009, pp. 3212–3217.
- [9] A. Albarelli, et al., Loosely distinctive features for robust surface alignment, in: *ECCV 2010*, 2010, pp. 519–532.



- [10] Y. Guo, et al., 3D object recognition in cluttered scenes with local surface features: a survey, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (2014) 2270–2287.
- [11] B. Bellekens, et al., A benchmark survey of rigid 3d point cloud registration algorithm, *Int. J. Adv. Intell. Syst.* 8 (2015) 118–127.
- [12] S. Rusinkiewicz, M. Levoy, Efficient variants of the icp algorithm, in: *Proceedings of the Conference on 3D Digital Imaging and Modeling*, 2001, pp. 145–152.
- [13] U. Castellani, A. Bartoli, 3D shape registration, in: N. Pears, Y.H. Liu, P. Bunting (Eds.), *3D Imaging, Analysis and Applications*, Springer, 2012, pp. 221–264.
- [14] S. Bouaziz, et al., Sparse iterative closest point, *Comput. Graph. Forum* 32 (2013) 113–123.
- [15] N. Mellado, et al., Super 4pcs fast global point cloud registration via smart indexing, *Comput. Graph. Forum* 33 (2014) 205–215.
- [16] A. Albarelli, et al., Fast and accurate surface alignment through an isometry-enforcing game, *Pattern Recognit.* 48 (2015) 2209–2226.
- [17] S.R. Bulò, I.M. Bomze, Infection and immunization: a new class of evolutionary game dynamics, *Games Econ. Behav.* 71 (2011) 193–211.
- [18] J.W. Weibull, *Evolutionary Game Theory*, MIT Press, 1995.
- [19] E. Rodolà, et al., A scale independent selection process for 3d object recognition in cluttered scenes, *Int. J. Comput. Vis.* 102 (2013) 129–145.
- [20] N. Mellado, et al., Relative scale estimation and 3d registration of multi-model geometry using growing least squares, *IEEE Trans. Vis. Comput. Graph* (2016), doi:10.1109/TVCG.2015.2505287.
- [21] L.M. Su, et al., Augmented reality during robot-assisted laparoscopic partial nephrectomy: toward real-time 3d-ct to stereoscopic video registration, *Urology* 73 (2009) 896–900.
- [22] Y.D. Li, P.H. Gu, Free-form surface inspection techniques state of the art review, *Comput. Aided Des.* 36 (2004) 1395–1417.
- [23] Z. Xie, et al., A high-accuracy method for fine registration of overlapping point clouds, *Image Vis. Comput.* 28 (2010) 563–570.
- [24] A. Zaharescu, et al., Surface feature detection and description with applications to mesh matching, in: *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2009, pp. 373–380.
- [25] A.E. Johnson, *Spin-images: a representation for 3-d surface matching*, Carnegie Mellon University, America, 1997 Ph.D. dissertation.
- [26] G. Turk, M. Levoy, Zippered polygon meshes from range images, in: *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, 1994, pp. 311–318.
- [27] Q.H. Du, Metrics for 3d rotations: comparison and analysis, *J. Math Imaging Vis.* 35 (2009) 155–164.