UDC 004.93, DOI: 10.2298/CSIS1001231W

3D Point Pattern Matching Based on Spatial Geometric Flexibility

Xiaopeng Wei[1,2], Xiaoyong Fang[1,2], Qiang Zhang[2],
and Dongsheng Zhou[1,2]

[1] School of Mechanical and Engineering, Dalian University of Technology, Dalian, 116024 China
[2] Key Laboratory of Advanced Design and Intelligent Computing(Dalian University), Ministry of Education, Dalian, 116622, China
zhangq26@126.com

Abstract. We propose a new method for matching two 3D point sets of identical cardinality with global similarity but local non-rigid deformations and distribution errors. This problem arises from marker based optical motion capture (Mocap) systems for facial Mocap data. To establish one-to-one identifications, we introduce a forward 3D point pattern matching (PPM) method based on spatial geometric flexibility, which considers a non-rigid deformation between the two point-sets. First, a model normalization algorithm based on simple rules is presented to normalize the two point-sets into a fixed space. Second, a facial topological structure model is constructed, which is used to preserve spatial information for each FP. Finally, we introduce a Local Deformation Matrix (LDM) to rectify local searching vector to meet the local deformation. Experimental results confirm that this method is applicable for robust 3D point pattern matching of sparse point sets with underlying non-rigid deformation and similar distribution.

Keywords. Point pattern matching (PPM), Face model, Spatial geometric flexibility, Topological structure, Motion capture (Mocap), Non-rigid deformation.

1. Introduction

Marker based optical Mocap system is widely used in clinical gait analysis, sports studies, animation and computer games [1], [2], [3]. Passive reflective markers are captured as 3D coordinates by means of image processing. The captured 3D coordinates is unorganized, so there is no method to know which coordinates is correspondent to which marker [4], [5].

This problem could appear in the two steps of Mocap data process: first, model generation, then, noise reduction and missing marker recovery. The second issue of the process for each trail is beyond the scope of our method. Model generation is to build a prior model to track the non-missing data for each sequence. The model is always built base on the first frame data. The work could be divided into two steps: first, outerliers cleaning and missing marker recovery, then, identifications of each marker. Traditionally, this work is all manual[3], [5], [6]. We propose the work to identify the markers to the system model in an automatic process.

This study is focus on the model generation issue in marker-based optical Mocap system, we are aim to present an automatic way to identify the point-set of the first frame model for different subjects or sequences. We formulate this problem as 3D point pattern matching: assume we have a manually identified template point-set $\mathbb{Q}^S = \{\mathbf{Pt}_i^S \in \mathbb{R}^3 \,|\, 1 \le i \le N\}$, where the symbol S represents the system template model (S-Model), N is the number of the points. We are going to match the to-be-matched point-set to the S-model point-set (M-model) $\mathbb{Q}^M = \{\mathbf{Pt}_i^M \in \mathbb{R}^3 \,|\, 1 \le i \le N\}$. The to-be-matched model is the first frame model as a prior model for different sequences, the superscript S and M represent the corresponding model. Assume $\mathbb{Q}^S$ and $\mathbb{Q}^M$ have overall spatial distribution similarity and same cardinality. Typically, due to underlying local flexibility and deformation of expressional facial Mocap FPs, there exists no single global scale, nor an affine transformation for the model generation. Considerations should be given to locally non-rigid deformation and special nature of facial structure.

PPM is commonly encountered in computer vision, image analysis, computational geometry, and pattern recognition. While matching subjects existing underlying non-rigid and flexible deformation (like face), non-rigid mapping is called. Besl and McKay [7] present a heuristic *Iterative Closest Point (ICP)* algorithm, which use nearest-neighbor methods to assign correspondence. Rohr et al. [8] and Wahba [9] adopted thin-plate spline to parameterize the non-rigid mapping, in which the outliers do not disturb. The softassign and deterministic annealing [10], [11] formulate pattern matching problem as a heuristic fuzzy problem, which guarantee very trusty one-to-one correspondences. Chui and Rangarajan propose a thin-plate spline based algorithm (TPS-RPM), the TPS-RPM identifies the correspondences, rejects a fraction of the outlier points simultaneously [12], [13]. To fully use the local spatial information, Feng et al. [14] defined the Neighborhood Relative Angle-Context Distribution (NRACD) and used it to match models with underlying local deformation. Zheng and Doermann introduced a point matching algorithm for non-rigid shapes [15], [16]. They formulated local neighborhood structure as a graph, and then used relaxation labeling to refine matching results. In the methods mentioned above, we can see that, the previous work on non-rigid models mostly formulated as an global optimization way[7], [8], [9], [10], [11], [12], [13], while Ref. [14], [15], [16] start to consider the local structure for non-rigid subjects, however, none of them have take local deformation into account for non-rigid and flexible subjects. Considering local deformation, we found that it's a fast and robust way to identify correspondences between 3D point-sets with underlying locally non-rigid deformation and distribution error.

Based above, a spatial geometric flexibility based method is proposed, which considers local deformations between the S-model and M-model. Fig 1. is the flow chart of the algorithm, the LDM is Local Deformation Matrix, which is to correct the local match.

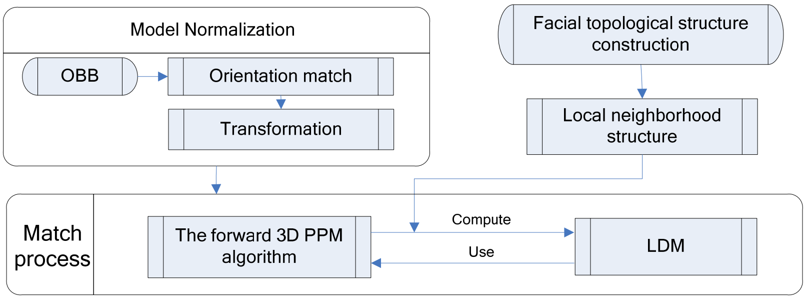
Model Normalization
OBB
Orientation match
Transformation
Facial topological structure construction
Local neighborhood structure
Match process
The forward 3D PPM algorithm
Compute
Use
LDM

Fig. 1. The flow chart of the algorithm.

2. Model Normalization

Human face is a non-rigid and flexible organic structure, which is comprised of complicated facial muscles and skin organism. To consider local non-rigid deformation, there should be a spatial platform, as the orientation and size are hard to be identical between different FP-sets. So, we proposed normalization method to put the two models into a fixed space.

In traditional way, to match a point with the known correspondence of its neighbor, searching vector is adopted to find the matching correspondence. As the position, size and orientation of the two models are not identical, the searching vectors in matching model ($\mathbf{V}_{search}^{M}$) have to transform to register $\mathbf{V}_{search}^{S}$, and the registration would repeat for one FP. However, our model normalization method puts the two models into one fixed space, the position, size and orientation of the models are normalized. It simplifies the registrations of searching vectors as all searching vectors are registered in one time, and makes the matching process more clear with low computational cost.

As mentioned above, the normalization could be divided into size and orientation normalization, accordingly, scale and rotation computation. To point-sets with known correspondence, the rotation computation could be formulated as well-known absolute orientation problem [17], [18]. However, in our application, the correspondences of the FP-sets are unknown. So the fundamental issue of the normalization is orientation normalization for the unorganized FP-sets.

2.1. Orientation matching problem

To determine a fixed space for FP-sets, we adopt bounding box technology. The bounding box technology is to find a fittest space for unorganized points.

2.1.1 Bounding box algorithm

The most commonly used bounding box algorithms are AABB (axis-aligned bounding box) [19], [20], Sphere [21], OBB (oriented bounding box) [22], [23] and FDH (fixed directions hulls) [24]. Both AABB and Sphere have not convincible spatial compactness. FDH is derived from AABB which could achieve convincible spatial compactness only by the premise that excessive fixed directions are selected. OBB use an optimal cuboid to bound 3D model, and could posse fine spatial compactness with only 3 orthogonal directions. As a most important fact, the facial topological structure

We choose OBB to construct bounding box for facial FP-sets for the following reasons.

-Fixed space for unorganized points.

-The directions is only 3.

-Geometric vivacity for facial structure: typically, the area of facial front-back side is largest, and the top-down side is smallest (as shown in Fig 2.).

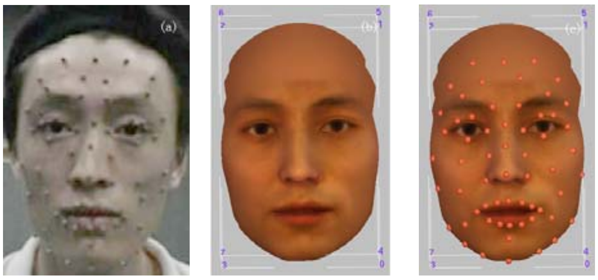

Fig. 2. The real and virtual effect of facial markers' distribution definition.(a) distribution of facial markers; (b)OBB on virtual face model without markers; (c) OBB on virtual face model with markers.

2.1.2 Orientation matching algorithm based on OBB

According to eigenvalue of the covariance matrix constructed by OBB [22], order the corresponding three eigenvectors as $\mathbf{m}_x$, $\mathbf{m}_y$, $\mathbf{m}_z$. $\mathbf{m}_x$ corresponds to the largest eigenvalue. Indicates the eight corner points of the

OBB box as $\mathbf{P}_0,\mathbf{P}_1,\mathbf{P}_2,\mathbf{P}_3\ \mathbf{P}_4,\mathbf{P}_5,\mathbf{P}_6,\mathbf{P}_7$. $\mathbf{P}_O$ is the center point of the box, $\mathbf{P}_O = 0.125\times\sum_{i=1}^{8}\mathbf{P}_i$.

We use the OBB technology [22] to construct a bounding box for S-Model and M-model. Then it's easy to scale the two boxes into a fixed space. To obtain scale matrix Mat_{Scale} , first, get projection distances d_{projx} , d_{projy} , d_{projz} , where $d_{projx} = (\mathbf{P}_6 - \mathbf{P}_0)\odot\mathbf{m}_x$, $d_{projy} = (\mathbf{P}_6 - \mathbf{P}_0)\odot\mathbf{m}_y$, $d_{projz} = (\mathbf{P}_6 - \mathbf{P}_0)\odot\mathbf{m}_z$, the $\odot$ is a vector dot product operator. Initialize Mat_{Scale} to a three order zero matrix, $\mathbf{Mat}_S$ (1,1)= d^S_{projx}/d^M_{projx} , $\mathbf{Mat}_S(2,2)$= d^S_{projy}/d^M_{projy} , $\mathbf{Mat}_S$ (3,3)= d^S_{projz}/d^M_{projz} . The scale matrix is stored for final normalization.

Fig 3. shows the optimal cuboid and the three directions colored by red ($\mathbf{m}_x$), green ($\mathbf{m}_y$) and blue ($\mathbf{m}_z$) from OBB technology. As the OBB is a statistical methods derived from PCA, it's a black box method. Hence for point-sets under same distribution, each of the three directions may be reversed as long as there are small geometric changes. As shown in Fig 3a. and Fig 3b., the M_x directions are opposite. We call this phenomenon as **orientation inconstancy** (OI).

Based on OBB technology, we could normalize the orientation of the point-sets by **simple rules**. Firstly, we can first easily determine the nasal tip FP, so we can find a direction that human face towards to; then, to find the second direction, we can find the two points at the lower jaw and near the ear bottom, which is different geometrically from the points on the forehead. We formulate the two rules into algorithm (discuss later), and then the orientation could be normalized.

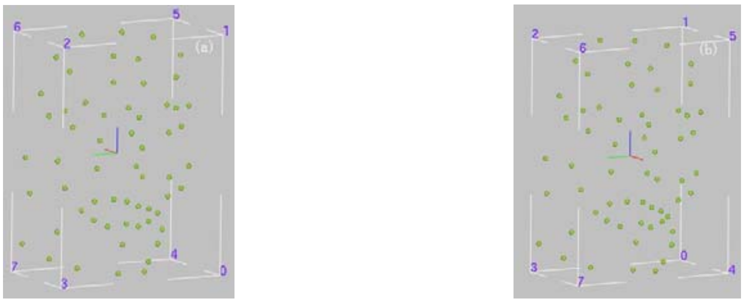

Fig. 3. Orientation inconstancy of First frame data templates. (a) the face facing against the direction $\mathbf{m}_x$; (a) the face facing along the direction $\mathbf{m}_x$.

After the OI had been resolved, the orientation normalization could be formulated as a common rotation computation. It's a common sense in computer graphics, provided that the 3D model's orientation has been given; it does easily obtain a rotation matrix to match orientation of one model to another. Therefore, the orientation matching algorithm is to mainly overcome the OI.

Matching $\mathbf{m}_x$ direction (rule 1)

Step 1: Find the farthest FP along $-M_x$ and M_x in the OBB box of S-model and M-model: Pt^S_{xi} and $Pt^M_{xi}\,(i=0,1)$.

$$\mathbf{Pt}_{x0} = \mathbf{Pt}_k \Big|_{(1\le k\le N)} (\mathbf{Pt}_k - \mathbf{P}_O)\odot(-\mathbf{m}_x) = \max_{0\le j\le N}(\mathbf{Pt}_j - \mathbf{P}_O)\odot(-\mathbf{m}_x) \qquad (1)$$

$$\mathbf{Pt}_{x1} = \mathbf{Pt}_k \Big|_{(1\le k\le N)} (\mathbf{Pt}_k - \mathbf{P}_O)\odot\mathbf{m}_x = \max_{0\le j\le N}(\mathbf{Pt}_j - \mathbf{P}_O)\odot\mathbf{m}_x \qquad (2)$$

Step 2: Compute the variance of the distances from $\mathbf{Pt}_{xi}(i=0,1)$ to from $\mathbf{Pt}_{xi}$ to four OBB vertices $P_{(i*4+j)}$ $(0\le j<4)$: $var_{xi}(i=0,1)$, obey:

$$\overline{d}_{xi} = \frac{1}{4}\sum_{j=0}^{3} \|\,\mathbf{Pt}_{xi} - \mathbf{P}_{4\times i+j}\,\| \qquad (3)$$

$$var_{xi} = \frac{1}{3}\sum_{j=0}^{3} (\|\,\mathbf{Pt}_{xi} - \mathbf{P}_{4\times i+j}\,\| - \overline{d}_{xi})^2 \qquad (4)$$

Where, $\overline{d}_{xi}$ is the mean value of distances from Pt_{xi} to the four OBB vertices located in corresponding side of OBB box.

Step 3: Match $\mathbf{m}_x$ direction by comparing var_{x0} to var_{x1}.

Compare var_{x0} to var_{x1}, if the later is smaller, means the FP $\mathbf{Pt}_{x1}$ is more likely nasal tip point, otherwise, $\mathbf{m}_x = -\mathbf{m}_x$, and: $\mathbf{P}_{temp} = \mathbf{P}_j$, $\mathbf{P}_j = \mathbf{P}_{4+j}$, $\mathbf{P}_{4+j} = \mathbf{P}_{temp}$ $(0\le j<4)$, where $\mathbf{P}_{temp}$ is a temporary point.

Matching $\mathbf{m}_z$ direction (rule 2)

Step 1: Obtain the nearest FP $\mathbf{P}_{zj}$ $(0\le j<4)$ of $\mathbf{P}_j$ $(0\le j<4)$:

$$\mathbf{Pt}_{zj} = \mathbf{Pt}_k \Big|_{(1\le k\le N)} (\mathbf{Pt}_k - \mathbf{P}_j) = \min_{0\le l\le N}\left\{(\mathbf{Pt}_l - \mathbf{P}_j)\right\} \qquad (5)$$

and the $\{\mathbf{P}_j(0\le j<4)\}$ are at the farthest plane along the $-\mathbf{m}_x$ of the OBB box, and the $\mathbf{m}_x$ directions have been normalize. Indicate $d^S_j = \|\,\mathbf{P}^S_{zj}\mathbf{P}^S_O\,\|$ $(0\le j<4)$ is the distances from $\mathbf{P}_{zj}$ to the center point ($\mathbf{P}_O$). $\mathbf{V}\in R^{3\times1}$, $\mathbf{V}^M_j = \mathbf{P}^M_{zj} - \mathbf{P}^M_O$, $\mathbf{Mat}^M_{T0} = (\mathbf{V}^M_0, \mathbf{V}^M_1, \mathbf{V}^M_2, \mathbf{V}^M_3)$, $\mathbf{Mat}^M_{T1}$

$= (\mathbf{V}_1^M, \mathbf{V}_0^M, \mathbf{V}_3^M, \mathbf{V}_2^M)$, $\mathbf{Mat}_{Vi}^M = \mathbf{Mat}_{Ti}^M \times Mat_s (i = 0,1)$. Store the lengths: $d_{ij}^M = (\sum_{k=1}^{3} (\mathbf{Mat}_{Vi}^M (k, j))^2)^{\frac{1}{2}}$.

$$\overline{d}_{zi} = \frac{1}{4} \sum_{j=0}^{3} d_j^S - d_{ij}^M \qquad (6)$$

$$var_{zi} = \frac{1}{3} \sum_{j=0}^{3} (d_j^S - d_{ij}^M - \overline{d}_{zi})^2 \qquad (7)$$

Step 2: Match M_z direction by comparing the two variances.

Compute var_{zi} with formula 3 and 4. Compare var_{z0} and var_{z1} , if the former is smaller, means the states that whether $\mathbf{m}_z$ directs at lower jaw in S-model and M-model are same, otherwise, we force them to be same: $\mathbf{m}_z^M = -\mathbf{m}_z^M$.

After matched $\mathbf{m}_x$ and $\mathbf{m}_z$, $\mathbf{m}_y$ is easy to match with orthogonality of the three directions: $\mathbf{m}_y = \mathbf{m}_x \otimes \mathbf{m}_z$, where $\otimes$ is a cross-product operator. This orientation matching algorithm is a forward calculation method, with extremely low computational cost, and highly geometric vivacity.

2.2. Transformation

As scale and orientation are given, it's convenient to transform M-model to the S-model space (as shown in Fig.4).

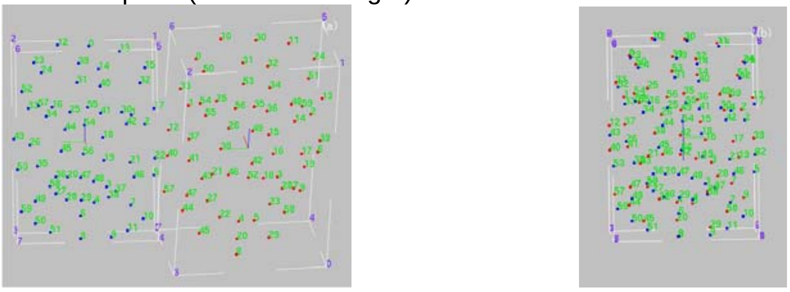

Fig. 4. Demonstration before and after model normalization. (a) before normalization; (b) after normalization

Step 1: Define a matrix $\mathbf{Mat}_{\mathbf{Pt}}^M$ for M-model to preserve the FPs' coordinates, wherein, $\mathbf{Mat}_{\mathbf{Pt}}^M (i,1) = \mathbf{Pt}_i^M .x$, $\mathbf{Mat}_{\mathbf{Pt}}^M (i,2) = \mathbf{Pt}_i^M .y$, $\mathbf{Mat}_{\mathbf{Pt}}^M (i,3) = \mathbf{Pt}_i^M .z \ (1 \le i \le N)$.

Step 2: Translate M-model to the global coordinate system. $\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,1)=\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,1)-\mathbf{P}_{O}^{M}.x$, $\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,2)=\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,2)-\mathbf{P}_{O}^{M}.y$, $\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,3)=\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,3)-\mathbf{P}_{O}^{M}.z$

Step 3: Rotate M-model to global coordinate system, then scale to uniform the sizes of M-model and S-model. Followed that, rotate M-model to keep its orientation identical to S-model. $\mathbf{Mat}_{RotO}^{M}=(\mathbf{m}_{x}^{M},\mathbf{m}_{y}^{M},\mathbf{m}_{z}^{M})^{-1}$, $\mathbf{Mat}_{RotS}^{M}=(\mathbf{m}_{x}^{S},\mathbf{m}_{y}^{S},\mathbf{m}_{z}^{S})$. $\mathbf{Mat}_{\mathbf{Pt}}^{M}=\mathbf{Mat}_{RotS}^{M}\times\mathbf{Mat}_{S}^{M}\times\mathbf{Mat}_{RotO}^{M}\times\mathbf{Mat}_{\mathbf{Pt}}^{M}$.

Step 4: Translate M-model to the space of S-model. $\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,1)=\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,1)+\mathbf{P}_{O}^{S}.x$, $\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,2)=\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,2)+\mathbf{P}_{O}^{S}.y$, $\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,3)=\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,3)+\mathbf{P}_{O}^{S}.z$. Back-feed the coordinate matrix to the FPs' coordinates, $\mathbf{Pt}_{i}^{M}.x=\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,1)$, $\mathbf{Pt}_{i}^{M}.y=\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,2)$, $\mathbf{Pt}_{i}^{M}.z=\mathbf{Mat}_{\mathbf{Pt}}^{M}(i,3)$ $(1\le i\le N)$, as shown in Fig. 4b.

After normalization, the influence of orientation and size difference between S-model and M-model is removed, so it is independent of orientation and size.

3. Construction of human facial topological structure

The facial organs such as mouth, eyes are unclosed, the movements of the upper and lower mouth lips are always different largely (same as the eye lips). Therefore, it's unreasonable to consider human face as a whole. Based on this consideration, firstly, we use Delaunay triangulation technology to construct a facial surface, and then neighborhood structures for FPs are obtained, as shown in Fig 5a. Secondly, by human computer interaction technology, we remove improper neighborhood relation, as shown in Fig. 5b. and 5c. Fig. 5a. shows some improper neighborhood relations between upper lower eyelids, upper lower lips, nasal tip and upper lip. Only with the structure considered the openness of eyes and mouth, the local deformation could be computed in nature.

3.1. 3D Triangulation and improper relationship removal

We use a simple 3D triangulation to construct the surface. First, we project all the FPs to the front plane of the OBB box, whose normal is $\mathbf{m}_{x}$ .Then the Delaunay triangulation is processed in 2D. To remove improper relationship, the human machine technology is used. We use an OOP (object oriented programming) in VC++, all the FP and relation lines are a pickable object. We developed a tool to remove or add relation lines for the facial surface. The particulars are beyond the scope of the paper.

3.2. Obtain local neighborhood relations

With the constructed facial topological structure, the neighbors of each FP are known. A relation line is defined to connect two adjacent points in the topological structure. Traverse the relation lines to obtain neighboring FPs' indices for each facial FP. The local neighborhood structure includes spatial relation such as angles, distances, orientations and so on.

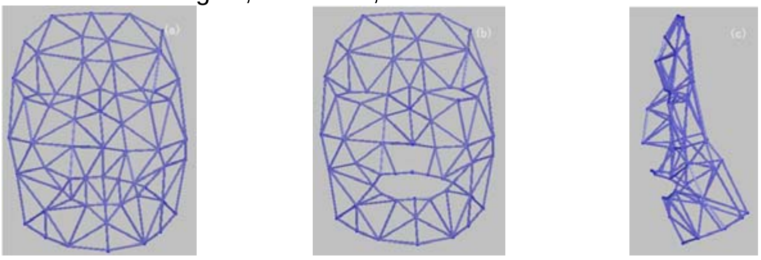

Fig. 5. Facial topological structure definition. (a) before removing improper relations after triangulation ;(b) facial topological structure, front view; (c) facial topological structure, side view.

4. Robust 3D point pattern matching based on spatial geometric flexibility

Def 1: let current base point index is $i_{curbase}$, current matching point index be $i_{curmatch}$, the searching vector in S-model be $\mathbf{V}^S_{Search} = \mathbf{Pt}^S_{i^S_{curmatch}} - \mathbf{Pt}^S_{i^S_{curbase}}$, LDM be $\mathbf{Mat}_{LDM}$, and searching radius be $r_{i^S_{curmatch}}$, if there exists a point whose index is $i^M_{curmatch}$ and it is in the sphere with center $\mathbf{Pt}^M_{i^M_{curmatch}}$ ($\mathbf{Pt}^M_{i^M_{curmatch}} = \mathbf{Pt}^M_{i^M_{curbase}} + \mathbf{Mat}_{LDM} \times \mathbf{V}^S_{Search}$) and radius $r_{i^S_{curmatch}}$, then $i^S_{curbase}$ and $i^M_{curbase}$ constitute a point match $\begin{pmatrix} i^S_{curmatch} \\ i^M_{curmatch} \end{pmatrix}$, we call this temporary success match (TSM) of $\mathbf{Pt}^S_{i^S_{curmatch}}$.

Def 2: let the size of the indices vector $vec^S_{i^S_{curbase}}$ (preserve the local neighborhood points' indices of $i^S_{curbase}$) be $N^S_{veccurbase}$, and the TSM number of points correspond to $vec^S_{i^S_{curbase}}$ be N^S_{TSM} , if $N^S_{TSM} / N^S_{veccurbase} > \varepsilon_{submatch}$, we call the point match $\begin{pmatrix} i^S_{curbase} \\ i^M_{curbase} \end{pmatrix}$ final success match (FSM) of $\mathbf{Pt}^S_{i^S_{curbase}}$. We call the way determining FSM by the relative amount of TSMs Temporary Feedback Method (**TFM**).

After normalization, the flow of robust 3D point pattern match algorithm based on spatial geometric flexibility is programmed in following pseudocode:

```
Push nasal tip FP's index into StackTempBase ($i^S_{nose}$ to
TempBaseStack$^S$ and $i^M_{nose}$ to TempBaseStack$^M$), Loop Time =0;
While TempBaseStack $\neq$Null
  For i = 1 to size of StackTempBase
    i_curbase = TempBaseStack[i];
    Compute and preserve TSMs for current base point
Pt$^S_{i\_curbase}$;
    Load information of local topological structure
    If Loop Time $\neq$ 0
          If $N^S_{TSM}/N^S_{veccurbase} > \xi_{submatch}$
           $\begin{pmatrix} i^S_{curmatch} \\ i^M_{curmatch} \end{pmatrix}$ is confirmed as a FSM.
          Endif
    Endif
     Update LDM for base point, and distribute the LDM;
          Push $i^S_{curbase}$into TempStack;
  EndFor
  Displace TempBaseStack by TempStack, empty TempStack,
plus 1 to Loop Time.
EndWhile
Empty TempBaseStack.
```

To insure the robustness of each point correspondence, we use a double insurance method. Firstly, we use known correspondence to search neighboring FP $\mathbf{Pt}^M_j$ corresponding to $\mathbf{Pt}^M_i$ (*i,j*<*N*). Secondly, we use Temporary Feedback Method to confirm a final match (discuss later). Above all, we assign an identity matrix to every LDM, which actually can not represent local deformation. The first FSM $\begin{pmatrix} i^S_{nose} \\ i^M_{nose} \end{pmatrix}$, is to be used as reference point correspondence. As the FSM is increasing, the LDM is updated to meet the local deformation. If the FSMs of the current base point is no less than two, we can compute the LDM, then the local deformation could be used for matching.

4.1. Algorithm of LDM computation

After normalization, the size and orientation between S-model and M-model are identical. As the FSM increasing in matching process, known correspondences are increasing. With these known correspondences, we formulate local rotation as a least square optimization:

$$\underset{\mathbf{R}_i}{\operatorname{argmin}} \sum_{k \in Y_i} \| \mathbf{R}_i \cdot (\mathbf{Pt}_{i,k}^M - \mathbf{Pt}_i^M) - (\mathbf{Pt}_{i,k}^S - \mathbf{Pt}_i^S) \|^2 \tag{8}$$

where, we denote the indices of one-ring neighbor for marker *i* as Y_i , $\mathbf{Pt}_{i,k}$ is the *k*th (in Y_i) neighbor of the *i*th marker. $\mathbf{R}_i^t \in \Re^{3\times3}$ indicates the optimum rotation for marker *i* respectively. We adopt the method of Horn [25] to find the optimum rotation matrix $\mathbf{Mat}_{LDM}$ for *i*th marker: $\mathbf{R}_i$. Besides the rotation, the space for tensile deformation is committed to the dynamic searching radius. However, Horn's method could not solve the situation the neighborhood points are less than three [18].The minimum number of neighborhood points for every FP is two. So, when neighborhood points for current base point is two, just set $\varepsilon_{submatch}$ bigger than 0.5 , we can insure the current base point is FSM(because it requires more than 0.5 × 2 neighborhood points are TSM. In our experiments, we set it to be 0.67). Based above, a simple approach is proposed to compute LDM with the two neighboring FPs.

Step 1: Construct a local plane with current base point and the two neighboring FPs. Obtain normal $\mathbf{V}_{norm}$ of the plane. Let the closest point of $\mathbf{Pt}_{i_{curbase}}$'s index be i_{near} , and the other's index be i_{far} , $\mathbf{V}_{near}$ = $\mathbf{Pt}_{i_{near}}$ - $\mathbf{Pt}_{i_{curbase}}$, $\mathbf{V}_{far}$ = $\mathbf{Pt}_{i_{far}}$ - $\mathbf{Pt}_{i_{curbase}}$, so:

$$\mathbf{V}_{norm} = \mathbf{V}_{near} \times \mathbf{V}_{far} \tag{9}$$

Step 2: Compute the orthogonal vector of $\mathbf{V}_{near}$ and $\mathbf{V}_{norm}$: $\mathbf{V}_{cons}$, then compute the LDM:

$$\mathbf{V}_{cons} = \mathbf{V}_{near} \times \mathbf{V}_{norm} \tag{10}$$

$$\mathbf{Mat}_{LDM} = (\mathbf{V}_{cons}^M, \mathbf{V}_{near}^M, \mathbf{V}_{norm}^M) \times (\mathbf{V}_{cons}^S, \mathbf{V}_{near}^S, \mathbf{V}_{norm}^S)^{-1} \tag{11}$$

4.2. Local searching with dynamic searching radius

Use the local topological structure we preserved to get the closet point $\mathbf{Pt}_j^S$ of $\mathbf{Pt}_i^S$ ($i \neq j$), $D_i = \left\| \mathbf{Pt}_j^S - \mathbf{Pt}_i^S \right\|$. We set a searching threshold value ε_r

($0 < \varepsilon_r < 1$) to control searching range for each FP: $r_i \ = \varepsilon_r . D_i$. Basically, the bigger ε_r is, the better tolerance to data distribution error our method has. According to experience from our experiments, ε_r is recommended to be close to 0.5.

5. Experiments and results

In our experiments, we use 60 markers for facial marker setup (as shown in Fig 2.). To evaluate the performance of our algorithm, we select five M-models which are with expressions: narrowing one eye, surprise, eyebrow up, laughing exaggeratedly, mouth and eyes widely open respectively(as shown in Fig 6b. to Fig 6f.), and we select a model with neutral expression(as shown in Fig 6a.) as our S-model. The algorithm described in the preceding sections has been implemented in a Visual C++ program. Our experiments were run on a PC with two 1.86GHz Pentium IV processors and 2GB memory.

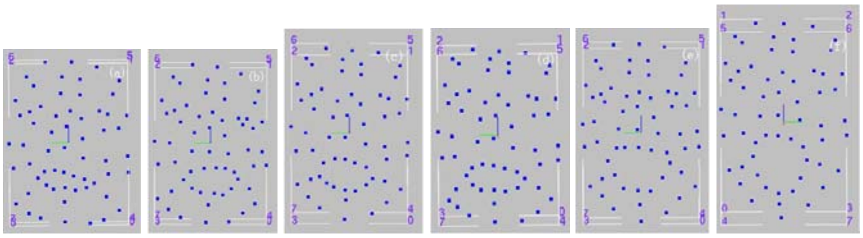

Fig. 6. First frame data templates with different expressions. (a) neutral;(b)narrowing one eye;(c)surprise;(d)eyebrow up;(e)laughing exaggeratedly;(f)mouth and eyes widely open.

We first show the effectiveness of our normalization method. Fig 7. shows the visual effects of experiment results, where red colored model is represents the normalized M-model. As we can see from this Figure, no matter how big the expressional difference between S-model and M-model is, the performance of our normalization algorithm is acceptable.

Next, we compare our FP matching algorithm with two different ways: using LDM (Fig 7c., Fig 7e. and Fig 7g.) and without using LDM (Fig 7d., Fig 7f. and Fig 7h.).Table 1. shows the result or comparison, wherein **FSM** column shows the number of final successful matched point, **Time** column shows machine running time. In addition, ε_r column represents the value which starts from zero and satisfies that 100% FPs are FSM. From this table, to models with expression which has slight local motions, such as the model with expression narrowing one eye, we use our algorithm without using LDM. Because local neighborhood structure was considered, it still achieved good results even with less computational cost. Along with increasing of the local motions' extent, it's easy to break in cycle process (as shown in Fig 7d., Fig 7f. and Fig 7h.). The reason is our method without using LDM didn't consider the effect of local motions and deformations. Due to local motions and deformations are considered, our algorithm with using LDM could match FPs for facial Mocap frame data templates almost perfectly. Although our algorithm with using LDM pays more computational cost, it still solves non-rigid and flexible matching in a considerable time. Some models with special or nearly extreme local deformations such as the model with expression: mouth and eyes widely open (as shown in Fig 6f.), are perfectly matched with S-model in our algorithm with using LDM.

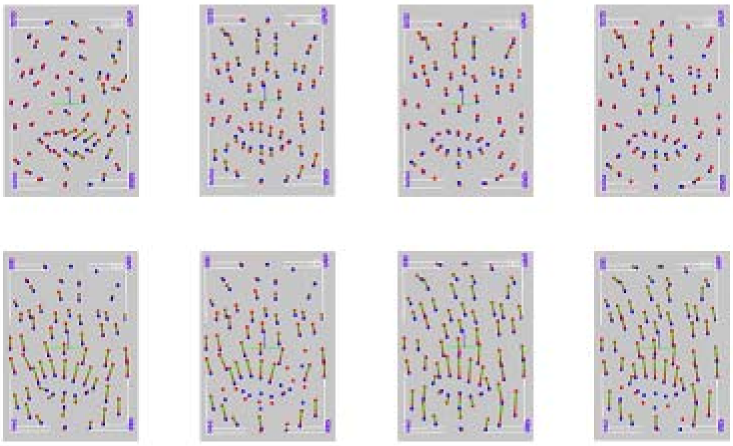

Fig. 7. The matching results of first frame data templates with different expressions.(a) the result of matching M1-model; (b) the result of matching M2-model; (c) the result of matching M3-model using LDM; (d) the result of matching M3-model without using LDM; (e) the result of matching M4-model using LDM; (f) the result of matching M4-model without using LDM; (g) the result of matching M5-model using LDM; (h) the result of matching M5-model without using LDM.

6. Conclusion

This paper proposed a robust PPM for matching unorganized point-sets with underlying non-rigid deformation. Our method is a forward calculation method with highly geometric vivacity. To consider local deformation, we normalize the point-sets into a fixed space, and then check the minor difference to derive local deformation. With the local deformation, the calculated LDM to rectify searching space, we achieved a more robust matching effort than considering local structure without deformation. This method considers the nature of the distribution of the FPs to find some rules to normalize the orientation. This model normalization is geometric intuitive, but can hardly be extended to all non-rigid point-sets, only if we can find some rules from the point distribution. The proposed method has been found effective for solving 3D facial point pattern matching problem for marker-based optical Mocap system. Experimental results demonstrate the robustness in different situation, with considerable low amount of computational cost.

Table 1. The experiment results of the Facial point matching algorithm

Expressions in Matching Models (M-model)	Using LDM			Without using LDM			Searching threshold value(ε_r)
	FSM	Loop Time	Time (ms)	FSM	Loop Time	Time (ms)	
Narrowing one eye (Fig 76)	60	9	47	60	7	32	0.303
Surprise (Fig 6c)	60	7	47	60	9	32	0.340
Eyebrow up (Fig 6d)	60	9	47	55	7	32	0.192
Laughing exaggeratedly (Fig 6e)	60	9	31	47	8	31	0.312
Mouth and eyes widely open (Fig 6f)	60	8	31	51	8	31	0.348

7. Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No. 60875046), by Program for New Century Excellent Talents in University (Grant No. NCET-06-0298), by the Key Project of Chinese Ministry of Education.(No.209029), by China Postdoctoral Science Foundation ((Grant No. 2009045 1266), by the Program for Liaoning Innovative Research Team in University (Grant Nos.2008T004 and 2009T005), by the Program for Liaoning Science and Technology Research in University (Grant Nos.2009S008, 2009S009) and by the Program for Liaoning Excellent Talents in University (Grant No.2006R06).

8. References

1. Moeslund, T. B. and Granum, E.: A Survey of Computer Vision-Based Human Motion Capture. Computer Vision and Image Understanding. 81, 231–268. (2001)
2. Moeslund, T. B. and Hilton, A. and Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding. 104, 90–126. (2006)

3. Richards, J. G.: The measurement of human motion: A comparison of commercially available systems. Human Movement Science. 18, 589-602. (1999)
4. Gleicher, M.: Animation from Observation: Motion Capture and Motion Editing. Computer Graphics. 33, 4, 51-54. (1999)
5. Li, B. and Holstein, H.: Using k-d Trees for Robust 3D Point Pattern Matching. Proceedings of Fourth International Conference on 3D Digital Imaging and Modeling, 95–102. (2003)
6. Jobbagy, A., Furnee, E., Harcos, P. Tarczy, M. Early detection of Parkinson's disease through automatic movement evaluation. Engineering in Medicine and Biology Magazine. 1998, 17(2):81-88
7. Besl, P. J. and McKay, D.: A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 ,2, 239-256. (1992)
8. Rohr, K. and Stiehl, H.S. and Sprengel, R. and Beil, W. and Buzug, T.M. and Weese, J. and Kuhn, M.H.: Point-Based Elastic Registration of Medical Image Data Using Approximating Thin-Plate Splines. Visualization in Biomedical Computing. 1131, 297-306. (1996)
9. Wahba, G.: Spline models for observational data. SIAM, Philadelphia, PA. (1990)
10. Chui, H. and Rambo, J. and Duncan, J. and Schultz, R. and Rangarajan, A.: Registration of cortical anatomical structures via robust 3Dpoint matching. Information Processing in Medical Imaging (IPMI),Springer, Berlin, 168–181. (1999)
11. Gold, S. and Rangarajan, A. and Lu, C.P. and Pappu, S. and Mjolsness, E.: New algorithms for 2-D and 3-D point matching: pose estimation and correspondence. Pattern Recognition. 31, 8, 1019–1031. (1998)
12. Chui, H and Rangarajan, A.: A new algorithm for non-rigid point matching. Computer Vision and Pattern Recognition. 2, 44-51. (2000)
13. Chui, H and Rangarajan, A.: A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding.2003,89: 114–141
14. Feng, J. and Horace, H and Ip, S. and Lai, L.Y. and Linney, A.: Robust point correspondence matching and similarity measuring for 3D models by relative angle-context distributions. Image and Vision Computing. 26, 6, 761-775. (2008)
15. Zheng, Y. and Doermann, D.: Robust Point Matching for Non-Rigid Shapes: A Relaxation Labeling Based Approach. IEEE Trans. Pattern Anal. Mach. Intell. 28, 4, 643-649. (2004)
16. Zheng,Y. and Doermann, D.: Robust Point Matching for Nonrigid Shapes by Preserving Local Neighborhood Structures. IEEE Transactions on Pattern Analysis and Machine Intelligence. 28, 4, 643-649. (2006)
17. Kanatani. K.: Analysis of 3-D Rotation Fitting. IEEE Transactions on Pattern Analysis and Machine Intelligence. 16, 5, 543-549. (1994)
18. Park, S. and Hodgins, J. K.: Capturing and animating skin deformation in human motion, ACM Transactions on Graphics (TOG). July, 25, 3. (2006)
19. Ponamgi, M. and Manocha, D. and Lin, M.: Incremental Algorithms for Collision Detection Between Polygonal Models. IEEE Transactions on Visualization and Computer Graphics. 3, 1, 51-64. (1997)
20. van den Bergen, G.: Efficient collision detection of complex deformable models using AABB trees. Journal of Graphics Tools. 2, 4, 1-13. (1997)
21. Hubband, P. M.: Collision Detection for Interactive Graphics Applications. IEEE Transaction on Visualization and Computer Graphics. 1, 3, 218-230. (1995)
22. Gottschalk, S. and Lin, M. C. and Manocha, D.: OBB tree: A hierarchical structure for rapid interference detection. Proceedings of the ACM SIGGRAPH 1996. New Orleans, USA, 171-180. (1996)

23. Barequet, G. and Har-Peled, S.: Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. Journal of Algorithms. 38, 1, 91-109. (2001)
24. Kay, T. L. and Kajiya, J. T.: Ray Tracing Complex Scenes. Computer Graphics. 20, 4, 269-278. (1983)
25. Horn, B. K. P.: Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A: Optics, Image Science, and Vision. 5, 7, 1127-1135. (1988)

Xiaopeng Wei is a professor at Dalian University of Technology and Dalian University, Dalian, China. His research area include computer animation, intelligent CAD. So far, he has (co-) authored about 160 papers published.

Xiaoyong Fang and **Dongsheng Zhou** are Ph.D students at Dalian University of Technology. Their research direction is computer animation.

Qiang Zhang is a professor at Dalian University, Dalian, China. His research interests are computer animation, intelligent computing. Now he has served as editorial boards of seven international journals and has edited special issues in journals such as Neurocomputing and International Journal of Computer Applications in Technology.

Received: April 25, 2009; Accepted: October 26, 2009.