Marker-less Facial Motion Capture based on the Parts Recognition

Yasuhiro AKAGI
Kagoshima University
Korimoto, 1-21-24
Kagoshima
890-8580, JAPAN
akagi@ibe.kagosima-u.ac.jp

Ryo FURUKAWA
Hiroshima City University
3-4-1, OzukaHigashi
AsaMinami-Ku, Hiroshima
731-3194, JAPAN
ryo-f@hiroshima-cu.ac.jp

Ryusuke SAGAWA
AIST
1-1-1 Higashi
Tsukuba, Ibaraki
305-8561 JAPAN
ryusuke.sagawa@aist.go.jp

Koichi OGAWARA
Wakayama University
Sakaedani 930
Wakayama-city
640-8510, JAPAN
ogawara@sys.wakayama-u.ac.jp

Hiroshi KAWASAKI
Kagoshima University
Korimoto 1-21-24
Kagoshima
890-8580, JAPAN.
kawasaki@ibe.kagosima-u.ac.jp

ABSTRACT

A motion capture method is used to capture facial motion to create 3D animations and for recognizing facial expressions. Since the facial motion consists of non-rigid deformations of a skin, it is difficult to track a transition of a point on the face over time. Therefore, a number of methods based on markers have been proposed to solve this problem. However, since it is difficult to place the markers on a face or on an actual texture of the face, it is difficult to apply the marker-based capture methods. To overcome this problem, we propose a marker-less motion capture method for facial motions. Since the thickness of a skin varies in each facial part, the features of the motion of the each parts also vary. These features make the non-rigid tracking problem more difficult. To prevent the problem, we recognize five types of facial parts (nose, mouth, eye, cheek and obstacle) from 3D points of a face by using Random Forest algorithm. After the recognition of the facial parts, we track the motion of the each part by using a non-rigid registration algorithm based on the Gaussian Mixture Model. Since the motions of the each part are independently detected, we integrate the motions of the each part as 3D shape deformations for tracking the motions of the points on the whole face. We adopt a Free-Form Deformation technique which is based on the Radial Basis Function for the integration. This deformation method deforms 3D shapes seamlessly with pairs of key points: several numbers of points of a source face and the corresponding points of a target shape which are detected by the non-rigid registration algorithm. Finally, we represent the motion of the face as the deformation from the face of the initial frame to the others. In our results, we show that the proposed method enables us to detect the motion of the face more accurately.

Keywords

Facial animation, Motion tracking, Non-rigid registration, Deformation, Random Forest

1 INTRODUCTION

The facial animation and capturing motions are one of the important topics in the area of computer graphics and vision[1, 2, 3, 4]. The most important point for capturing motions is that the capturing method can correctly track the movement of all points (vertexes) from a frame to another frame of the motion. In case of a facial motion capture, since the movement of a skin on a face is mainly caused by a movement of muscles, it has more flexibility than that of a body movement caused by a movement of a bone and an axis. To detect the movement of a face between frames, a number of methods have been proposed. The method which tracks artificial markers on a face is the well-known approach to capture the motion of a face[3, 5]. By using the artificial markers, a capturing method can track the movement of the point robustly, even if a point on a face has fewer features for tracking. In other cases, when it is difficult to place the artificial markers on a face or when we want to capture the texture at the same time, marker-less capturing methods are used. Some of the marker-less methods detect corresponding points between the frames based on the features of the shape[6, 7]. These feature based approaches enable us to track a movement of a mouth, nose and eyes which have strong features of its shape. However, it is difficult to track a movement of a cheek and a forehead which have a few features by the feature based approaches. In this case, the non-rigid registration algorithms[8, 9] can be a solution to track the movement of a face. If we apply the non-rigid registration algorithm for tracking, the algorithm detects the transformation from a face in one frame to that of another frame smoothly. However, since the thickness of a skin varies in each part of a face, a movement of a face also varies in each part. This feature of the non-rigid registration algorithm erases the feature of the movement of a face.

To solve this problem, we propose a motion capture method that independently tracks movements of each part of a face. For this method, we define five parts of a face: nose, mouth, eye, cheek and obstacle. To recognize the facial parts from a face, we utilize a Random Forest[10]. After the recognition of the facial parts, we detect the movement of the each facial parts with the non-rigid registration algorithm[8, 9]. Finally, we integrate the movement of each part into a single one to represent a movement of the whole shape of a face.

2 RELATED WORK

In this section, we describe the related works about the facial motion capture and the registration of 3D objects.

2.1 Facial Motion Capture

Since it is difficult to create the facial animations manually, a number of methods are proposed to capture the facial motion automatically. There are two types of approaches to capture the facial motion: a marker-based approach and a marker-less approach.

The marker-based approach has the advantage that it can more robustly detect the movement of the markers than the marker-less approaches. Huang *et al.* proposed the method to capture a high-fidelity facial movement by using one hundred markers[3]. This method can capture realistic and dynamic wrinkles and fine-scale facial details. However, the marker-based approach requires time to set up the marker and it is difficult to use to capture the scene, since the object has to be in contact with the face. Bickel *et al.* proposed the method that uses painting markers to capture the detailed deformation of the face such as the wrinkles[11][5]. These marker-based approaches have a common problem that it is laborious to put the markers and it is difficult to capture the natural texture at the same time with the motion.

On the other hand, the marker-less approaches are proposed to solve the problems of markers. Sibbing *et al.* uses the feature tracker like the KLT tracker[12] to detect the movement of a face[4]. They proposed the Surfel Fitting method to fit the deformed face to the scanned 3D points of the face. Weise *et al.* constructs the facial performance database of a person to detect the motion of the face from 2D image and 3D point set[13]. This method finds the similar poses from the database and reconstructs the facial expression by combining the similar poses. This kind of method uses 2D texture information. Therefore, the motion capture method based only on the 3D point is required and the following approaches are conducted.

2.2 Registration of 3D objects

The registration algorithm is commonly used to integrate two point sets that are scanned from different positions and angles. It is also useful to detect the movement of an object. Therefore, there are a number of approaches to capture the motion of the object by using the registration algorithms.

2.2.1 Rigid Registration

The rigid registration algorithms detect the translation and rotation of an object from two point sets. The ICP algorithm[14] is one of the well-known approaches. In this paper, we try to detect the non-rigid movement of a face. Although, the scanned data of a face is almost non-rigid, it still contains a rigid transformation. Therefore, we use the Normal Distributions Transform (NDT) algorithm[2] to estimate the rigid transformation of a face. The NDT is robustly estimating the transformation by using a mixture of a Gaussian distribution to represent the distribution of the points[15][16].

2.2.2 Non-Rigid Registration

The non-rigid registration algorithms become one of the most important topics in computer vision and graphic fields and it is useful to detect the movement of the point sets[7, 8, 6, 17]. Myronenko et al. proposed the Coherent Point Drift(CPD) algorithm[9] that regards the distribution of the point set as Gaussian Mixture Models and minimizes the distance of the Gaussian centroids. The method proposed Chui *et al.* is a similar approach that uses Expectation-Maximization (EM) algorithm with the basis function as the thin-plate spline (EM+TPS)[18]. Jian *et al.* advanced this kind of approach by using the L2 distance between Gaussian mixtures representing two point sets[8]. They compare the performance of four types of approaches: L2 distance and TPS based approach (L2+TPS), L2 distance and the Gaussian based RBF (GRBF) based approach (L2+GBRF), TPS and CPD. Since these algorithms have the acceptable robustness and accuracy, we use the L2+TPS algorithm to detect the movement of facial parts.

Tevs *et al.* proposed a novel method which is called an Animation Cartography[7]. This method detects the correspondence of all points from a frame to the next frame by using the graph matching algorithm. This method detects the holes to interpolate them by considering the global correspondences. Li *et al.* proposed a robust reconstruction method for a moving object[6]. This method uses the template model which is made from the shape of the first frame and key points of the template. To detect the movement, they deform the template model to fit the shape of the next frame[19].

2.3 Recognition of a human body

In our approach, we recognize the facial parts from an input point data. In terms of studies on the recognition of a human body, various methods have been proposed. Shotton et al. proposed an efficient method to quickly and accurately predict 3D positions of body joints from a depth image[20]. They use the Random Forest[10] to recognize the body parts. Dantone also uses the Random Forest based approach to recognize the facial parts in real-time[21]. One of the advantages of the Random Forest is that it quickly decides the class of the input, even when the input data is huge and the feature vector is composed of large dimensions.

3 OVERVIEW

Our method is composed of four steps as shown in the Figure 1. In step (a), we create surfaces of a face from a set of point cloud generated by the 3D scanning method[22]. We use a 3D shape feature which is called First Point Feature Histogram (FPFH) [23] as a feature vector for a Random Forest to recognize the facial parts. At this time, we also create a height map (the height is equal to the depth from the camera) which is used for the step (d).

In step (b), we train the Random Forest to recognize the facial parts. Then, we classify the facial parts into five types: nose, mouth, eye, cheek and obstacle. The feature vector of each point of the face is defined by using the FPFH and a normalized position. To calculate the normalized position, we define the origin point at the tip of the nose and normalize positions of all points into -1.0 to 1.0. We create the training set by selecting the each part manually. We will describe the details of this step in Sec. 4.

In step (c), by using the results from the step (b), we detect the movement of the each facial part independently. We calculate the transformation of the each point from the point set belongs to first frame to it belongs to other frames by using the L2-TPS algorithm.

In step (d), we integrate the transformation of the each point that is estimated in the step (c) into the small number of the key points to represent the deformation of the face. We use a deformation method which is based on the Radial Basis Function (RBF)[24, 1]. Finally, the motions of the face are represented by the deformations that deform the face shape from the first frame to other frames. We will describe the details both the step (c) and (d) in Sec. 5.

4 FACIAL PARTS RECOGNITION

In this section, we describe how to recognize the facial parts from 3D points of a face (the step (b) in Figure 1). Since the number of 3D points included in the motion of a face is usually huge, it is required that the recognition algorithm has an acceptable performance to process over $100,000$ points. The Random Forest algorithm (RF) has the advantage in speed to process the huge data. On the other hand, RF has a problem called "over fitting". This is when RF has a tendency to recognize an unknown object as one category (it is difficult to reject the data) in case of a generic object recognition. In case of a facial parts recognition for motion tracking, there are a few numbers of unknown objects like outliers and obstacles. To overcome the problem, we add the obstacles into one class of the facial parts. Finally, we define the five classifications of the facial parts: "nose", "mouth", "eye", "cheek" and "obstacle". The reason why we don't differ the left and right parts is that if the number of the parts is large, each possibility value of the parts of a point is more widely spread. Then, the points which are not determined any facial parts are increased. To prevent this, we select the five classifications of the facial parts. In the following sections, we will discuss about a feature vector for recognizing the facial parts and training process of the RF.

4.1 Feature Vector for Recognizing the Facial Parts

It is important to define the feature vector for the Random Forest to recognize the facial parts exactly. A feature vector which has enough information about the 3D shape of a face is suitable for recognizing the face. Therefore, we utilize the FPFH method[23] to compose the feature vector because the FPFH efficiently represents the feature of the local shape of an object. FPFH is a feature descriptor for point clouds that consists of a histogram of the mean curvature between a point and neighboring points. Because each facial part has its own features in the curvature, we select FPFH as the feature vector. In our preliminary experiment, there are some errors when we recognize the facial parts form a face with only FPFH as a feature vector. Then, we add a normalized position of a point to the feature vector. To

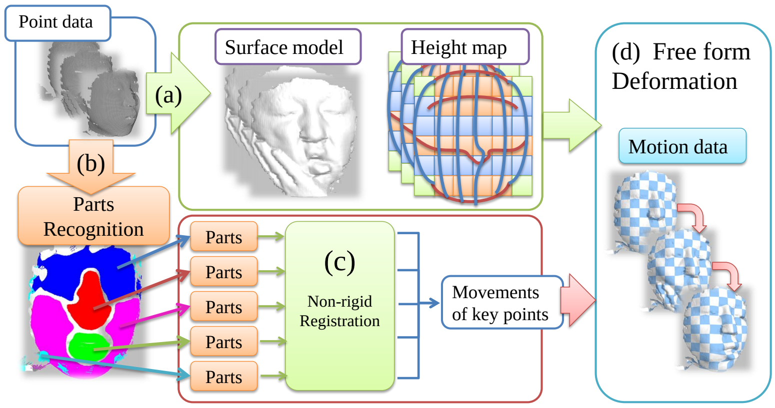

Figure 1: Overview of the marker-less facial motion capture. (a)Initial registration and triangulation form the scanned 3D points. (b)Facial parts recognition using the random forest. (c)Non-rigid registration for tracking the movement of each part. (d)Face deformations based on the movement of the parts.

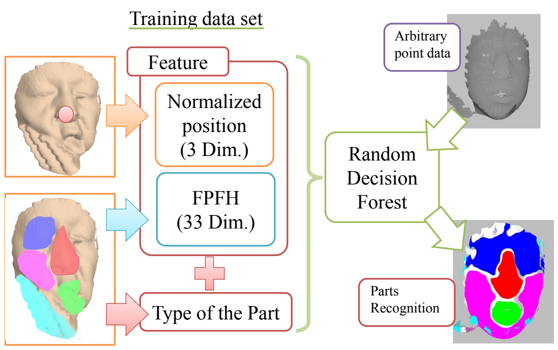

Figure 2: Flow diagram of the recognition of the facial parts.

denote the normalized position, we define the center of the face as the tip of the nose and normalize the size of the face into the range of -1.0 to 1.0. This additional feature makes each decision tree of the Random Forest simpler. The feature values of the FPFH are usually similar between left eye and right eye. However those of the normalized position are different. This strong feature is useful to make the decision process of facial parts simpler. Hence, the feature vector has the 33 dimensions information from the FPFH and three dimensions from the normalized position for representing the feature of a point.

4.2 Training Process

We train the RF by using the classification of the facial parts and the feature vector. To make training sets, we choose 6 frames from the input motion and specify the area of the facial parts manually. In this approach, users have to make the training sets for every input data set. This is the limitation of our method, however this approach has the benefits that it enables us to reject scanning noises and an obstacle. Then, we also select the tip

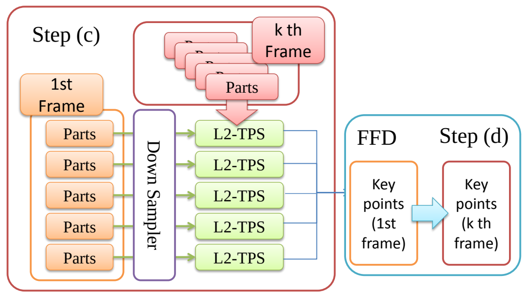

Figure 3: Flow diagram of the motion tracking and deformation of a face.

of the nose for creating the feature vector. The trained RF is used in the motion tracking process.

5 MOTION TRACKING FOR THE FACIAL PARTS

In this section, we explain about the motion tracing process of a face (step (c) in Figure 1). Figure 3 is the detailed flow diagram of the step (c) to (d) in Figure 1. We describe a set of points representing a face in t th frame as $F^t = (v_0^t, \cdots, v_i^t, \cdots, v_N^t)^{\mathrm{T}}$. And a facial part c of the face is described as $P_c^t = (v_x^t, v_y^t, \cdots)^{\mathrm{T}}$.

Before the tracking process, we recognize the facial parts P_c^t by using the Random Forest $RF(x)$. The $RF(x)$ gives the value of the possibility of each facial part for each point. We accept the point which has the possibility of over 90% as the part P_c^t(Eq. 1).

$$RD(F^t)(the\, possibility\, of\, c\, is\, over\, 90\%) \rightarrow P_c^t \quad (1)$$

In the following sections, we describe the details of the motion tracking process.

5.1 Sampling of tracking points

Our method tracks movements of each part P_c^t on the Non-rigid Registration method named "L2+TPS" (described in section 2.2.2). Since the L2+TPS is designed to deal with less than a thousand points, it is difficult to apply our face model (it contains hundreds of thousands of points.) However, since L2+TPS is useful way to track a deformation of a non-rigid object, we reduce the number of the points for tracking (Figure 3 (a)). We apply the down sampling technique which is called approximate voxel grid filtering [25] to reduce the number of the points within 800 to 1500. We describe the reduced point set of P_c^t as $\hat{P}_c^t$. Then, since most of points of P_c^t are not tracked by L2+TPS by the reduction of points, we will calculate movements of the reduced points by interpolating the movement of $\hat{P}_c^t$ in section 6.

5.2 Discussion about the target frame of the tracking

When we track the movement of a face from a multi-frame data, there are two types of approaches for tracking the movement. One is the approach which tracks the movement between the neighboring frames (from P_c^t to $P_c^{(t+1)}$). The other is that it tracks the movement from the initial frame to other ones (from P_c^0 to P_c^t). The former approach has an advantage that since the difference between P_c^t and $P_c^{(t+1)}$ is usually small, it is easy to fit P_c^t to $P_c^{(t+1)}$ by using non-rigid registration algorithms. However, if the number of frames is large, errors that are caused by each tracking step are accumulated in large amounts. The latter approach has an advantage that even if a tracking result from P_c^0 to P_c^t contains error, the result does not affect the other tracking results. And if the shapes of a face around the end of the motion are similar to the shape of the initial frame, the tracking results will be better than the former approach. In case of facial motions, since differences from the initial shape to the other ones are smaller than those of body motions, L2+TPS can directly track the movement form P_c^0 to P_c^t. Therefor we adopt the latter approach for tracking.

5.3 Motion Tracking based on the Parts Recognition

Equation 2 represents the tracking procedural from $\hat{P}_c^0$ to $\hat{P}_c^t$ by using L2+TPS.

$$L2+TPS_{\hat{P}_c^0 \rightarrow \hat{P}_c^t}(\hat{P}_c^0) = Q_c^t \tag{2}$$

Where Q_c^t represents registered positions from $\hat{P}_c^0$ to fit to $\hat{P}_c^t$. Q_c^t and $\hat{P}_c^1$ has same number of points and the each point of Q_c^t is related to the each point of $\hat{P}_c^t$. In the following chapter, we will describe an integration process of tracking results Q_c^t to detect the movements of all points of a face F^0.

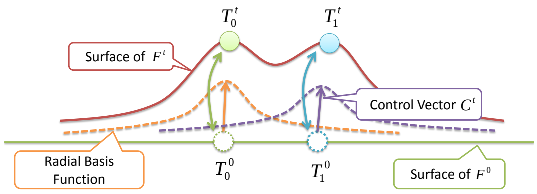

Figure 4: RBF based deformation.

6 INTEGRATION OF TRACKING RESULTS OF THE FACIAL PARTS

The tracking results Q_c^t are the sparse information about movements of each facial part c. We calculate movements from F^0 to other frames by applying a shape deformation method. In this process, since we don't consider the difference of the facial parts, we describe a tracking result of the entire face as $Q_c^t = \cup_c Q_c^t$. Now, L2+TPS registration method gives the positions of all points Q^t on the face F^t. When a shape deformation method deforms F^0 to fit to F^t, Q^0 should be able to deform to Q^t by the method. Therefore, we utilize the shape deformation method described in the following section.

6.1 Shape deformation method based on Radial Basis Function

We introduce the shape deformation method based on RBF which deforms an object by interpolating moving points[1]. The basic idea of the shape deformation is as follows (Figure 4). Now, we put a point T_0^0 on a surface $F^0 = (v_0^0, \cdots, v_i^0, \cdots, v_N^0)^{\mathrm{T}}$ and give a deformation target point T_0^t. When only the point T_0^0 moves to T_0^t, a deformed surface $\hat{F}^t$ is easy to determine by the following equation.

$$v_i^t = v_i^0 + RBF(||T_0^0 - v_i^0||)(T_0^t - T_0^0) \tag{3}$$

By using the equation 3, T_0^0 is correctly moved to T_0^t. However, if we add another pair of point T_1^0 and T_1^t, the deformed surface doesn't pass the T_0^t and T_1^t because of the movement vectors $(T_0^0 - v_i^0)$ and $(T_0^0 - v_i^0)$ affect each other by RBF. To solve this problem, the shape deformation method detects vectors which are called the "control vector" C^t. By considering the effect of moving points, the C^t is detected to make $\hat{F}^t$ passing T_0^t and T_0^t. In this study, we have the pairs of moving points Q^0 and Q^t. Then, we describe the movement of the points as $M^t = Q^t - Q^0$ and the distance between the points of Q^0 is given by the following matrix D.

$$D(Q^1, Q^1) = \begin{bmatrix} ||Q_0^1 - Q_0^1|| & \cdots & ||Q_0^1 - Q_m^1|| \\ \vdots & \ddots & \vdots \\ ||Q_m^1 - Q_0^1|| & \cdots & ||Q_m^1 - Q_m^1|| \end{bmatrix}. \tag{4}$$

Table 1: Size of frames and points of input data.

Data set	Slap	Smile	Stretch	Grip
No. of frames	46	55	48	48
No. of points	115169	190599	321142	161517

Then, the relationship of the control vector C^t and the movement of the points M^t is as following equation.

$$T^t = RBF(D(Q^1, Q^1))C^t \quad (5)$$

Since M^t and Q^1 are the known values, we can detect C^t by using the invert matrix of D (equation 6).

$$RBF(D(Q^1, Q^1))^{-1} T^t = C^t \quad (6)$$

Finally, by using the control vector C^t, an arbitrary point v_i^0 of the face F^0 is deformed to the $\hat{F}^t$ by the following equation.

$$v_i^t = RBF(D(v_i^0, Q^1))C^t \quad (7)$$

Through this process, the movement from F^0 to $\hat{F}^t$ is detected.

6.2 kernel function

We discuss a kernel function of RBF which defines the feature of the interpolation between points. Since we track the movement of the facial parts independently, it has a feature that there are a few points of Q^t between the boundaries of the parts. If we use a Gaussian like function as a kernel function of RBF, the movement of the area where the density of the points is sparse is smaller than the other area. Thus making an unnatural movement of the face. Therefore, we select a kernel function as Euclid distance $RBF(x) = ||x||$ which affects the movement to the wider area. This kernel function is also used in the research of facial deformation by Noh *et al.* [24].

7 RESULTS AND DISCUSSIONS

In this chapter, we show tracking results of four types of motions and discuss the accuracy of the results. The four types of motions are as follows: slapping a face (Slap), make a smile (Smile), vertically shrunk face turn to expand (Stretch) and it is an example of general object that a hand grips two rubber balls (figure 5). Table 1 shows numbers of points consists of the initial frame and frames of each motion.

7.1 Accuracy of the captured motion

For evaluating the accuracy of the proposed method and previous Non-rigid registration methods (L2+TPS[8], L2+GRBF[8], EM+TPS[18], CPD[9]), we calculate the

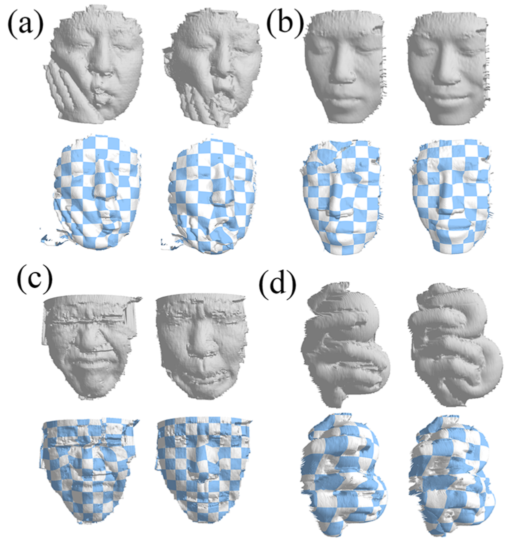

Figure 5: Examples of the input motions and tracked ones (upper row:input, bottom row:tracked). (a)Slap. (b)Smile (c)Stretch (d)Grip

Table 2: Conversion of the mean distances between the input point clouds and the captured faces of the each algorithm.

	Mean distances (mm)				
Data set	Our	L2+TPS	L2+GBRF	EM+TPS	CPD
Slap	1.70	2.23	2.37	2.38	2.35
Smile	0.56	0.78	0.54	0.55	0.79
Stretch	1.27	1.55	1.46	1.55	1.63
Grip	0.84	1.55	1.52	1.85	1.55

mean distance between the tracked face $\hat{F}^t$ and the input faces F^t as errors of tracking. Table 2 shows the conversion of the mean distances of each method.

This result shows that the proposed method method succeeds to reduce the errors of the motion "Slap", "Stretch" and "Grip". Especially in the motion "Slap", even though this motion includes a hand contacted to the face, we can reduce the errors. Figure 6 visualizes the errors of some frames of the motion. In this result, the errors on the nose and the cheek are reduced. This result shows that by dividing the facial parts for tracing, we can reduce making wrong correspondences of the tracking.

7.2 Evaluation of identity between the initial face and the tracked face

In the section 7.1, we evaluated the distances between the initial face and the tracked one. Although it has a meaning to evaluate the fitness of both the faces, it is not clear that a point of the initial face correctly moves to the identical point of other ones. Then, to evaluate the identity between the initial face and other

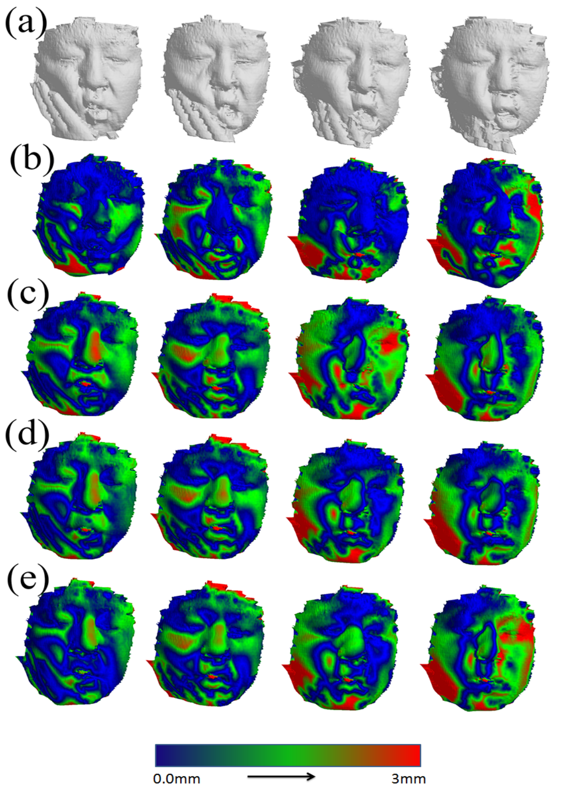

Figure 6: Visualization of errors on the face. (a)Input motion. (b)Results tracked by proposed method. (c)L2+TPS. (d)L2+GBRF. (e)EM+TPS.

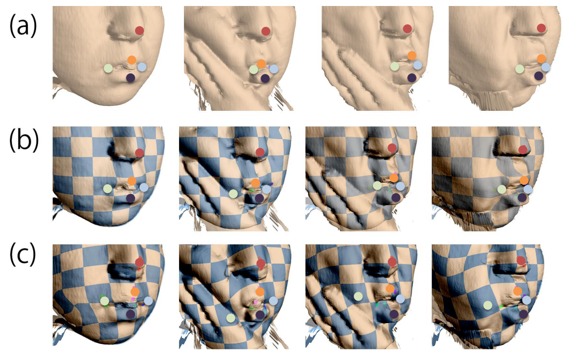

Figure 7: Accuracy of the motion. (a)Grand-truth (b)Proposed method (c)L2-TPS

ones, we manually made the ground truth of the movement around the mouth by giving same positions of five key points in each frame (the top row of the figure 7). Figure 8 shows the mean distance between the tracked points and the ground truth data. These results show that the proposed method can detect the movement of the face more accurately than other methods. However, our method has several errors in the positions of the several key points. This will be our next challenging task to improve in our method.

8 CONCLUSION

The conclusion of this paper is as follows:

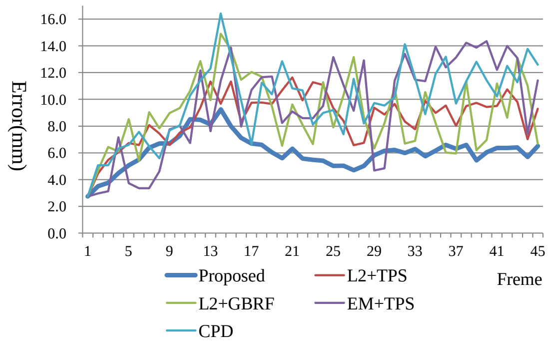

Figure 8: Conversions of the tracking error of the key points around the mouth.

- The proposed method succeeded to detect the movement of a face as the deformation from the initial shape to the other shapes.
- The recognition of the facial parts provides the efficient constraint to detect the correct transformation of the key points.
- We show that the proposed method enables us to detect the motion of the face more accurately than the other non-rigid registration algorithms.

The proposed method still has several errors to detect the shape of the facial parts. In the future, we will utilize the texture information of a face to capture more accurate motions.

9 ACKNOWLEDGMENT

This work was supported in part by the Funding Program for Next Generation World-Leading Researchers(NEXT Program) No.LR030 and Grant-in-Aid for Young Scientists(B) No.25870570 in Japan.

10 REFERENCES

[1] Botsch, M., Kobbelt, L.: Real-time shape editing using radial basis functions. Comput. Graph. Forum **24** (2005) 611–621

[2] Magnusson, M.: The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection. PhD thesis, Orebro University (2009) Orebro Studies in Technology 36.

[3] Huang, H., Chai, J., Tong, X., Wu, H.T.: Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. ACM Trans. Graph. **30** (2011) 74:1–74:10

[4] Sibbing, D., Habbecke, M., Kobbelt, L.: Markerless reconstruction and synthesis of dynamic facial expressions. Comput. Vis. Image Underst. **115** (2011) 668–680

[5] Bickel, B., Lang, M., Botsch, M., Otaduy, M.A., Gross, M.: Pose-space animation and transfer of facial details. In: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA ’08, Eurographics Association (2008) 57–66

[6] Li, H., Adams, B., Guibas, L.J., Pauly, M.: Robust single-view geometry and motion reconstruction. ACM Trans. Graph. **28** (2009) 175:1–175:10

[7] Tevs, A., Berner, A., Wand, M., Ihrke, I., Bokeloh, M., Kerber, J., Seidel, H.P.: Animation cartography –intrinsic reconstruction of shape and motion. ACM Trans. Graph. **31** (2012) 12:1–12:15

[8] Jian, B., Vemuri, B.C.: Robust point set registration using gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **33** (2011) 1633–1645

[9] Myronenko, A., Song, X.: Point set registration: Coherent point drift. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 2262–2275

[10] Breiman, L.: Random forests. Mach. Learn. **45** (2001) 5–32

[11] Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M., Pfister, H., Gross, M.: Multi-scale capture of facial geometry and motion. ACM Trans. Graph. **26** (2007)

[12] Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR’94). (1994) 593 – 600

[13] Weise, T., Bouaziz, S., Li, H., Pauly, M.: Realtime performance-based facial animation. In: ACM SIGGRAPH 2011 papers. SIGGRAPH ’11, New York, NY, USA, ACM (2011) 77:1–77:10

[14] Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14** (1992) 239–256

[15] Pathak, K., Birk, A., Vaškevičius, N., Poppinga, J.: Fast registration based on noisy planes with unknown correspondences for 3-d mapping. Trans. Rob. **26** (2010) 424–441

[16] Stoyanov, T., Magnusson, M., Almqvist, H., Lilienthal, A.J.: On the Accuracy of the 3D Normal Distributions Transform as a Tool for Spatial Representation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). (2011)

[17] Zhang, W., Wang, Q., Tang, X.: Real time feature based 3-d deformable face tracking. In: Proceedings of the 10th European Conference on Computer Vision: Part II. ECCV ’08, Berlin, Heidelberg, Springer-Verlag (2008) 720–732

[18] Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. Comput. Vis. Image Underst. **89** (2003) 114–141

[19] Sumner, R.W., Schmid, J., Pauly, M.: Embedded deformation for shape manipulation. In: ACM SIGGRAPH 2007 papers. SIGGRAPH ’07, ACM (2007)

[20] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. CVPR ’11, Washington, DC, USA, IEEE Computer Society (2011) 1297–1304

[21] Dantone, M., Gall, J., Fanelli, G., Gool, L.V.: Real-time facial feature detection using conditional regression forests. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition. (2012)

[22] Sagawa, R., Kawasaki, H., Furukawa, R., Kiyota, S.: Dense one-shot 3d reconstruction by detecting continuous regions with parallel line projection. In: Proc. 13th IEEE International Conference on Conputer Vison(ICCV 2011). (2011) 1911–1918

[23] Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: Proceedings of the 2009 IEEE international conference on Robotics and Automation. ICRA’09, Piscataway, NJ, USA, IEEE Press (2009) 1848–1853

[24] Noh, J.y., Fidaleo, D., Neumann, U.: Animated deformations with radial basis functions. In: Proceedings of the ACM symposium on Virtual reality software and technology. VRST ’00, New York, NY, USA, ACM (2000) 166–174

[25] Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China (2011)