

Scattered data processing approach based on optical facial motion capture

Qiang Zhang*, Xiaoying Liang and Xiaopeng Wei

Key Laboratory of Advanced Design and Intelligent Computing (Dalian University), Ministry of Education, Dalian, China

Abstract. In recent years, animation reconstruction of facial expressions has become a popular research field in computer science and motion capture-based facial expression reconstruction is now emerging in this field. Based on the facial motion data obtained using a passive optical motion capture system, we propose a scattered data processing approach, which aims to solve the common problems of missing data and noise. To recover missing data, given the nonlinear relationships among neighbors with the current missing marker, we propose an improved version of a previous method, where we use the motion of three muscles rather than one to recover the missing data. To reduce the noise, we initially apply preprocessing to eliminate impulsive noise, before our proposed three-order quasi-uniform B-spline-based fitting method is used to reduce the remaining noise. Our experiments showed that the principles that underlie this method are simple and straightforward, and it delivered acceptable precision during reconstruction.

Keywords: Animation reconstruction, B-spline noise reduction, data recovery

1. Introduction

In recent years, because of the rapid development of the computer graphics industry, motion capture (Mo-Cap) has become one of the most popular techniques for acquiring natural human motion due to its high precision, real-time features, and convenience. Techniques have been introduced that allow the Mo-Cap device to be used to retrieve rigid body transformations and to process soft body motion, such as the facial movements of an actor. Animation reconstruction from a human face is one of the most popular fields in current computer science research and Mo-Cap-based facial expression reconstruction is an emerging issue in this field. Passive optical Mo-Cap is a technique that

observes moving markers from multiple angles, transforms the position of markers into three-dimensional (3D) coordinates by graphic processing, and obtains the movements of markers attached to the surface of a moving object in real time. However, 3D geometric models can not be driven directly by the source motion data captured by passive optical Mo-Cap systems. The major reasons for this are as follows: (1) there is unavoidable noise in the captured source data, which comes mainly from calibration errors, electrical noise, and jumping by sensor devices or markers; and (2) there are missing data. During the processing of captured motion data, multiple markers often appear to overlap or obscure each other, or they may be invisible compared with the situation when using two cameras, all of which makes the positions of these markers difficult to reconstruct and missing data are the outcomes. These two problems are referred to as the missing data and noise problems, which are the most

*Corresponding author: Qiang Zhang, Key Laboratory of Advanced Design and Intelligent Computing (Dalian University), Ministry of Education, Dalian 116622, China. E-mail: zhangq@dlu.edu.cn.

common problems encountered during the processing of captured motion data.

To obtain a better simulation, captured motion data are often used to reconstruct facial expressions in the field of facial Mo-Cap. Motion data processing is the foundation of facial motion reconstruction. In recent years, many effective methods have been proposed for missing data recovery. Lin and Ouhyoung [1] proposed the weighted summation of neighbor displacements to repair missing data, where the distance between neighboring markers and missing markers determined the weights of the movements of neighboring markers, so the weights became smaller as the distance increased. Guenter et al. [2] used the average displacements of valid neighbors to represent the movements of missing data, which repaired the missing data. To ensure that there were sufficient valid neighbors for the current missing points, they used a greedy algorithm to build the repaired sequence for missing markers. All of these methods exploit information related to neighboring markers to repair the missing data, thereby facilitating the recovery of missing data based on the local movement angle. However, the analysis of local movements using these methods is not sufficiently robust and data processing requires manual intervention. Edge et al. [3] used a Kalman filter to reduce noise and the discrete cosine transform (DCT) to reconstruct missing motion trajectory segments. However, this method was only applicable to time sequence information and it failed to exploit spatial information. Aristidou et al. [4] proposed a method based on a Kalman filter for predicting missing data, which ensured adequate precision by exploiting information related to neighboring markers to obtain a fair result.

After missing data recovery, however, data dithering still exists because of calibration errors, electrical noise, and jumping by sensor devices or markers, so the motion trajectory is not sufficiently smooth to reconstruct facial expression. Thus, if we use these data to reconstruct facial expressions, the animation produced will not be sufficiently natural and realistic. Given the non-rigid structure of the human face, noise reduction based on curve fitting is an effective trajectory smoothing method for facial motion data. Therefore, we consider the application of curve-based fitting to reduce noise and to smooth the motion trajectory. Many methods have been proposed to address the problems of noise reduction and trajectory smoothing. Liu et al. [5] proposed a curve fitting method based on the least squares principle, where they con-

sidered the problem of fitting polygons. However, a higher order polygon could not obtain a better fitting result and there was a limitation on the order of the polygon, because the computational complexity increased with the order. After considering the effects of low-order fitting, they proposed the addition of segments to ensure the smoothness of the fitting. Hsieh et al. [6] proposed an impulsive noise reduction method for rigid body motion data, which used B-spline wavelets and obtained good results. However, this method is aimed at human body to do experiments, although we might consider applying it to facial data processing. Horsch and Juttler [7] proposed the construction of a C1 consecutive spline curve. Using the dispersal of the spline curve to obtain the rotation and position, they drove a robot through a specific route. To some degree, this algorithm solved the problem. Each curve segment was grouped using a third-order Bezier curve and the control points of the curve were computed based on the position and velocity (tangent vector) of the robot during processing. Somasundaram and Parent [8] proposed a method for simple linear interpolation to smooth the trajectory and remove outliers. Fang [9] proposed an online curve modeling framework for constructing online curves to smooth the trajectories and obtained fair results. To obtain the final result, Bickel [10] presented a multiscale representation and acquisition method for animating high-resolution facial geometry and wrinkles. In a marker-less approach, Weise [11] introduced a face tracking algorithm that combined geometry and texture registration with pre-recorded animation priors during a single optimization. The 3D facial dynamics could be reconstructed in real-time without using face markers. Akhter et al. [12] presented bilinear spatiotemporal basis models and used the models for denoising and marker labeling in raw Mo-Cap data.

Most of these previous data recovery methods exploited neighbors to repair the missing data and recovered the missing data based on the local movement angle. We also propose data recovery based on the effects of neighbors. Our method is also guided by a muscle motion mechanism proposed by Waters [13], which we have improved.

From the perspective of recovering missing data, the principle of the muscle motion mechanism proposed by Waters [13] only considers the influence of one neighboring point on the current missing marker, whereas we use multiple points. However, given the inevitable missing data problem, there might not be sufficient

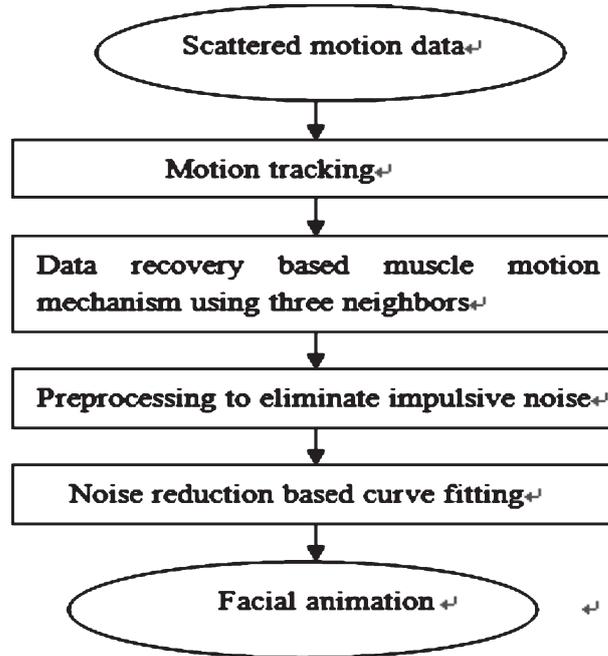


Fig. 1. Data processing flowchart.

neighbors available for recovery. For each marker, its most distant neighboring marker has the lowest effect on its movement, whereas the points derived from the three nearest markers have most effect on the current marker's movement. Thus, we propose a muscle motion mechanism based on three neighbors for data recovery.

After data recovery, given the effect of impulsive noise on the motion trajectory and the nonrigid structure of the human face, we initially apply preprocessing to eliminate impulsive noise, before using our proposed curve fitting and noise reduction methods, where we apply a third-order B-spline to fitting to obtain a smoother motion trajectory.

The remainder of this paper is organized as follows. An overview of the data processing flow is provided in Section 2. Section 3 describes the data recovery algorithm based on a muscle motion mechanism using three neighbors. Section 4 explains the noise reduction-based curve fitting algorithm, which is used to smooth the motion trajectory. The experimental results are reported in Section 5. Some illustrative examples are provided to demonstrate the effectiveness of the proposed algorithm. Finally, we give our concluding remarks and suggest future research directions in Section 6.

2. Algorithm overview

The inherent difficulty of working with facial Mo-Cap is related to the problem of how to use the captured scattered motion data to reconstruct facial motion expressions. Thus, for missing data recovery and noise reduction, we designed a data processing method, the essence of which is as follows. We apply an algorithm to the scattered motion data sequence to recognize their topology, before recovering missing data and eliminating noise.

In this section, we describe the flow of the motion data processing method, which is shown in Fig. 1. A facial motion capture sequence is the input of the processing task. First, using the scattered motion data captured by a passive optical Mo-Cap system and to perform an accurate calibration, we propose a motion tracking method to track the trajectories of the points in real time and we store all of the captured points based on their time sequence. Second, to obtain the complete motion data, we recover the missing data using a muscle motion mechanism based on three neighbors from the tracked motion data. Third, our proposed preprocessing method is used to reduce the impulsive noise based on a mid-value filter. Fourth, a third-order quasi-uniform B-spline-based curve fitting method is

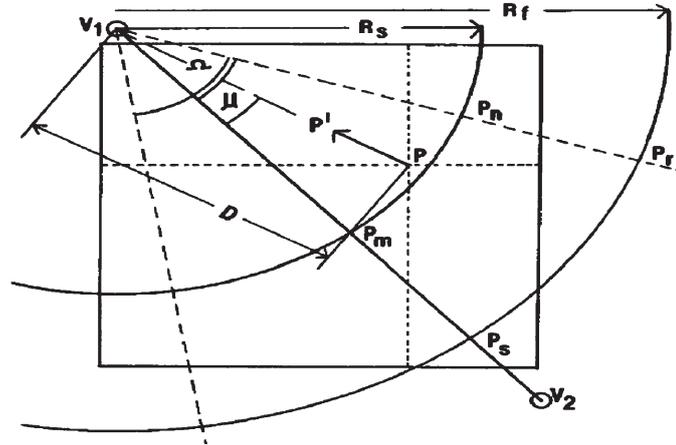


Fig. 2. Muscle vector representation [13], © 1987 ACM, Inc. Reprinted by permission, <http://doi.acm.org/10.1145/37401.37405>.

proposed to reduce the remaining noise and to obtain smooth motion trajectories. Finally, we reconstruct the facial expression motions using the smoothed data. An animation graph is the final output of motion data processing. Each step of this data processing method is necessary to solve the common problems. We validated the efficiency of our methods.

3. Data recovery using a muscle motion mechanism based on three neighbors

Motion tracking is used to follow the trajectories of markers in real time according to their time sequence, which is also the foundation of our motion data processing method. After explaining motion tracking, we describe how to apply the muscle motion mechanism to recover missing data from the closest three neighbors. Given the nonlinear relationships among the neighbors of the current missing marker, our method is guided by the muscle motion mechanism proposed by Waters [13], which uses one muscle, whereas we use three to recover a missing point.

3.1. Muscle motion mechanism

Waters' muscle model [13] is a widely used animation model. Muscle $V1V2$ is decided by two points: the point of bony attachment and the point of skin attachment. Figure 2 illustrates how the adjacent tissue, such as the node P , is affected by this muscle vector contraction. We can assume maximum displacement at the point of attachment to the skin and zero displacement

at the point of bony attachment. A decline in the displacement is dissipated through the adjoining tissue across the sectors $PmPn$ and $VIPs$. Using a nonlinear interpolant, it is possible to represent the simple action of a muscle, such as that shown in Fig. 2. P is moving toward P' and the muscle vector is described in two dimensions. Point $P(x,y)$ is displaced by $P'(x',y')$.

$V1$ and $V2$ are used to construct a linear muscle. Any point P located in the mesh grouped by the field $VIPrPs$ will be displaced by P' in the direction of the vector PVI .

The point P exists in the field of $VIPnPm$, so the displacing radius factor R is defined as:

$$R = \cos \left(\left(1 - \frac{D}{R_s} \right) \times \frac{\pi}{2} \right) \quad (1)$$

The point P exists in the field of $PnPrPsPm$, so the displacing radius factor R is defined as:

$$R = \cos \left(\frac{D - R_s}{R_f - R_s} \times \frac{\pi}{2} \right) \quad (2)$$

where R determines the position of P' . Waters used a muscle motion mechanism to animate facial animation. Thus, we used this mechanism to repair missing data and we set a new weight formula for factor R , while also taking the cosine function as the design function for weight R .

3.2. Missing data recovery

The marker position can be estimated using the muscle motion mechanism. In this study, constraints on the

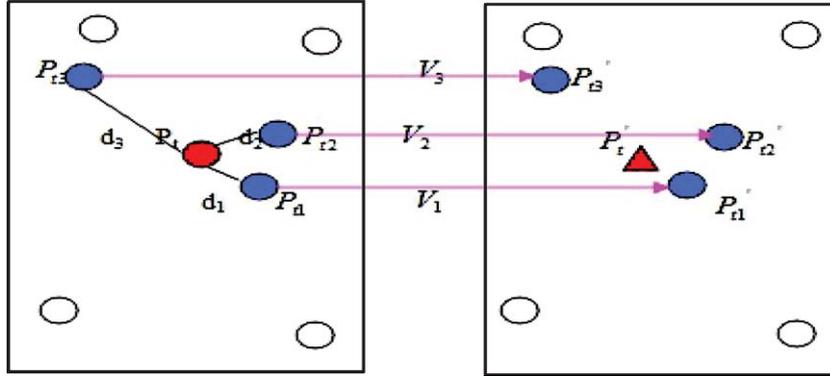


Fig. 3. Missing data recovery method.

three neighbors are used to produce a more reliable and convincing algorithm.

Why do we only use three neighbors to recover the motion data? First, given the inevitable missing data problem, there might not be sufficient neighbors available for recovery. Second, for each marker, its most distant marker has the lowest influence on its movement. Thus, the points derived from the three nearest markers have most influence on the current marker's movement. This is why our proposed muscle motion mechanism is based on using three neighbors for data recovery.

The details of the algorithm are as follows. To recover the position of a specific missing marker, it is necessary to find the three nearest non-missing markers in the previous and the current frames. According to the facial muscle motion mechanism, we use the movements of the three points to predict the missing markers' position. The overall method is illustrated in Fig. 3.

In Fig. 3, the left section shows the marker positions in the previous frame while the right section shows the current frame. \circ represents non-missing points and \blacktriangle represents the missing point P_t' . P_{t1} , P_{t2} , and P_{t3} are the three nearest points to point P_t in the previous frame, and P_{t1}' , P_{t2}' , and P_{t3}' represent the corresponding points in the current frame for points P_{t1} , P_{t2} , and P_{t3} in the previous frame, respectively. V_1 , V_2 , and V_3 denote the vectors formed between the corresponding points in the previous and current frames.

3.2.1. Data recovery based on three neighbors

The position of a missing point in the current frame is predicted based on the motion data in the previous

frame. We briefly summarize the basic steps of motion data recovery algorithm to clarify the details of this approach, as follows.

- (1) Find point P_t in the previous frame, which corresponds to a specific missing point.
- (2) Determine the three nearest points to P_t , i.e., P_{t1} , P_{t2} , and P_{t3} , as well as the distance set d_1 , d_2 , and d_3 , and the vector set V_1 , V_2 , and V_3 for the nearest points in the previous and current frames.
- (3) To normalize each distance, it is necessary to determine the sum d_{all} of the distance set.

$$d_{all} = d_1 + d_2 + d_3 \quad (3)$$

- (4) The improved formula for the facial muscle mechanism is based on using the movements of three neighbors to determine the movement of missing one. As the weight increases, the distance becomes nearer to the missing point. The sum of the weights for the three neighbors is close to 1, which indicates that the three neighbors represent all of the effects of the missing point. The weight w_i of the vector set is determined using the improved formula (4), which is designed for three neighbors.

$$\omega_i = \left[\frac{1 + \cos\left(\frac{d_i}{d_{all}} \times \pi\right)}{2} \right]^4 \quad (4)$$

- (5) To determine the displaced point P_t' based on the predicted current missing point, we need to add the point P_t to the calculated effect of the motion vector. For only three neighbors, we use formula 5.

$$P'_t = P_t + \sum_{i=1}^3 \omega_i V_i \quad (5)$$

3.2.2. Data recovery based on more than three neighbors

Our method employs a muscle motion mechanism to use the neighborhood structure to determine the position of missing or occluded markers during capture. We only employ three neighbors to recover the missing point, which provides a fair result. However, most previous studies used more than three points, e.g., one-ring neighbors, to recover the missing point. To compare our method with previous approaches, we designed another formula for determining the weights, which was adapted to more than three neighbors.

- (1) To use more than three neighbors to predict the missing point, we designed another formula for w_i , which is shown in (6):

$$\omega_i = \cos \left(\frac{d_i - R_s}{R_f - R_s} \times \frac{\pi}{2.5} \right) \quad (6)$$

where R_s is the nearest neighbor to the missing point and R_f is the most distant neighbor. This formula is adapted for the muscle motion mechanism.

- (2) To determine the displaced point P'_t based on the prediction of the current missing point, we use the following formula (7).

$$P'_t = P_t + \frac{\sum_{i=1}^n (\omega_i V_i)}{\sum_{i=1}^n \omega_i} \quad (7)$$

4. Noise reduction based on curve fitting

Our data recovery algorithm makes a contribution to the automatic maintenance of the naturalness of facial animation. In some cases, however, the outcomes of data recovery cannot satisfy the needs of users. Thus, we propose a technique that allows a more flexible motion trajectory, where curve fitting is applied to reduce the noise and to yield a smooth motion trajectory.

During the processing of captured facial motion data, few methods employ B-spline curve fitting to reduce the noise. Thus, given the continuity of the curve and the simplicity of this computation, we developed a noise reduction method based on quasi-uniform

B-spline fitting to smooth motion data. However, a low-order B-spline could not obtain a good result because of its simplicity and it failed to smooth the motion trajectories. By contrast, a high-order B-spline produced a smooth result but it lost some details of the motion data, while it also required complex computations. Therefore, we perform preprocessing to eliminate impulsive noise and we then apply the proposed third-order quasi-uniform B-spline-based fitting method to reduce noise.

Smooth motion trajectories are obtained using these two processes for noise reduction. The first is preprocessing based on a mid-value filter to reduce impulsive noise. The second is noise reduction based on B-spline curve fitting to obtain a smoother motion trajectory.

4.1. Impulsive noise removal

4.1.1. Impulsive noise model

The statistical analysis of impulsive noise uses the noise produced by spatial feature markers in a time-series. We build a model of impulsive noise and describe the temporal distribution of the noise data. The amplitude of the impulsive noise components should be large for impulsive noise-embedded motion data. The noise components are identified as coefficients with high magnitudes. The reduction of impulsive noise follows a two-step procedure: the identification of noise components; and the replacement of noise components with interpolated components, using their neighboring coefficients.

4.1.2. Impulsive noise processing

Impulsive noise preprocessing is conducted as follows. If the given geometric appearance of most of the control points is smooth and only a few control points are not smooth (referred to as a “bad point” or impulsive noise), we use a method that modifies the bad point to eliminate impulsive noise. The impulsive noise reduction procedure follows a two-step procedure: (1) identification of bad points; and (2) modification of bad points.

- (1) Identification of bad points

Given control points $P_j (j=0,1,2, \dots, n)$, the threshold k for judging bad points is given by (8).

$$k = \frac{\sum_{i=1}^n P_{i-1} P_i}{n} \quad (8)$$

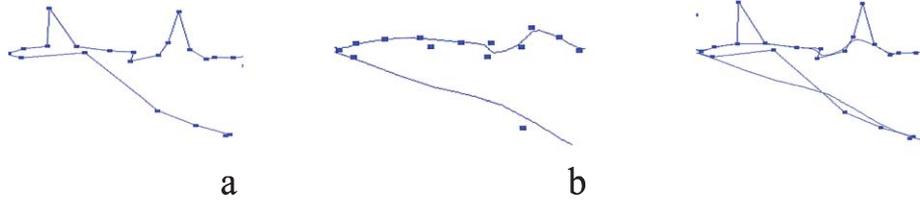


Fig. 4. Motion tracking chart with impulsive noise and curve fitting with preprocessing. (a) Motion tracking chart with impulsive noise. (b) Trajectory chart after curve fitting based on preprocessing. (c) Motion tracking chart with impulsive noise and the curve fitting chart obtained by preprocessing.

Bad points are identified using formula (9):

$$\begin{cases} P_{i-1}P_i \geq 1.5k \\ P_iP_{i+1} \geq 1.5k \end{cases} \quad (9)$$

where P_i denotes a bad point. The value of i ranges between 1 and $n-1$. k represents the threshold used to judge bad points, the value of which is dynamic and it changes with the trajectories of different markers.

2) Modification of bad points

We use a mid-value filter to replace the bad point. Given that P_i is a bad point, the formula used to modify it is as follows.

$$P_i = \frac{P_{i-1} + P_{i+1}}{2} \quad (10)$$

Thus, P_i is modified to replace the bad point.

For clarity, we briefly describe the preprocessing of impulsive noise in tracked motion data, where the results are shown in Fig. 4.

As shown in Fig. 4, the motion trajectory of a point is not sufficiently smooth because of the effect of impulsive noise. However, it becomes smoother after preprocessing, mainly because the effect of impulsive noise is eliminated.

4.2. B-spline curve fitting

4.2.1. B-spline curve

The starting points of the B-spline defined by Clark [14, 15] are as follows:

- 1) This is a local approach, rather than interpolation for the given points;
- 2) There are continuity requirements between neighbors that approach curve segments.

The most commonly used mathematical representation of an n -order B-spline curve is given by (11):

$$P_{i,n}(t) = \sum_{k=0}^n P_{i+k} F_{k,n}(t) \quad (11)$$

$$F_{k,n}(t) = 1/n! \sum_{j=0}^{n-k} (-1)^j C_{n+1}^j (t+n-k-j)^n \quad (12)$$

where P_{i+k} is a control point and $F_{k,n}(t)$ represents the corresponding base function to P_{i+k} . $t \in [0, 1]$. In the formula (10), $i = 0, 1, 2, \dots, m-2, m-1, m$. Thus, based on the formula given above, a B-spline curve is defined by segments. Given $m+n+1$ points P_i ($i = 0, 1, 2, 3, \dots, m+n$), we can define $m+1$ segments of an n -order parametric curve.

4.2.2. Curve-fitting algorithm

In this section, we describe the performance of sequence-level facial motion data processing after data recovery and preprocessing. In addition to eliminating the impulsive noise using a mid-value filter, we can also edit a sequence of frames by manipulating the corresponding B-spline curve-based fit directly to reduce the remaining noise. We selected a B-spline curve to reduce noise mainly because of its smooth interpolation and local control properties.

The order of a B-spline is correlated to its computational complexity, so it is necessary to consider this problem for the B-spline order we aim to use. The use of a high-order curve to reduce noise can produce a smooth motion trajectory but it may cause the loss of some detailed information, and it also has high computational complexity. By contrast, the use of a low-order curve cannot produce a sufficiently smooth motion trajectory but it retains detailed motion information and has good real-time performance. Given the preprocessing method used for impulsive noise described above, we can simply offset the shortcomings of curve fitting

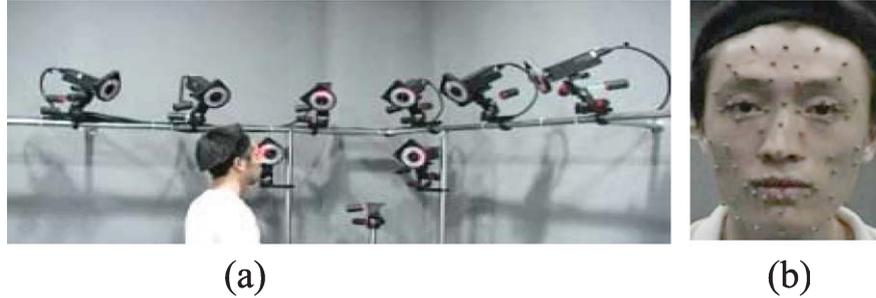


Fig. 5. Facial Mo-Cap environment (a) and facial marker setup (b).

by using a low-order curve. Thus, we use third-order quasi-unique B-spline-based fitting to reduce the noise to obtain adequate motion trajectory results.

The ends of the third-order B-spline curve are not located at the margins of the polygon, or at the start and end points. Thus, we need to design the curve to start or end at a given point, so we use the quasi-uniform B-spline curve, triple vertex method [16] to reduce noise. The curve used in this method passes through the start and end points of the motion trajectory.

In our experimental study, an actress with 60 markers on her face was asked to perform facial expressions. We focused on fitting the trajectories of these 60 markers by using a third-order B-spline curve to reduce the noise, which was conducted as follows. First, the valid traced points and the control points of each trajectory were obtained. Second, the 3D coordinates of the traced points were calculated based on the control points. Finally, the calculated traced points were used to process the traced motion data to obtain a smooth motion trajectory.

5. Results

To test the feasibility of our proposed methods, we developed a data processing system for passive optical Mo-Cap.

To test the robustness and efficiency of the system, we performed several experiments related to motion tracking, data recovery, impulsive noise preprocessing, and reduction of the remaining noise based on curve fitting. The experimental details are provided in this section. The test platform was a compatible PC with a Pentium(R) Dual-Core CPU E5400 processor, 2 GB of memory, and the operating system was Windows 7. The methods used were based on the nonlinear

relationships among neighbors with the muscle motion mechanism, which had a high level of computational complexity.

5.1. Experimental setting

We processed motion data captured by a passive optical Mo-Cap system, DVMC-8820 [17], which comprised eight infra-red (IR) cameras with four million pixels and a capture rate of 60 Hz. In the experiment, 60 IR sensor markers were placed on the faces of performers. During performances, head movements were restricted within a small range, i.e., the rotation angle was $<5^\circ$ and global shifts were less than 1/20th of the head length, as shown in Fig. 5.

5.2. Experimental results and analysis

To validate the feasibility of the data recovery and noise reduction methods proposed in Sections 3 and 4, we performed experiments based on the VC++ platform and using an OpenGL graphic library, which addressed the common problems of missing data and noise. The results were as follows.

(1) Validation of the motion data recovery method

To confirm the validity of our method, we implemented Guenter's [2] algorithm, which uses the average displacements of valid neighbors to recover missing points, whereas our method uses three neighbors. We also used the algorithm proposed by Lin and Ouhyoung [1], which employs the weighted sum of the displacements of more than three neighbors, and we designed a new formula to generate the weights. Our method uses the muscle motion mechanism for three neighbors to recover missing data. Finally, we reconstructed the facial expressions from the recovered

Method ↵	Frame 200 ↵	Frame 210 ↵	Frame 220 ↵	Frame 230 ↵	Frame 240 ↵	Frame 250 ↵
Average recovery method ↵ (Guenter) ↵						
Lots of neighbors recovery method ↵ (Lin. Ouyang) ↵						
Three neighbors recovery method ↵ (ours) ↵						

Fig. 6. Frames showing facial motion reconstruction with different recovery methods.

motion data using all three recovery methods and the results are shown in Fig. 6.

In Fig. 6, the blue points are the captured motion data and the red points are the recovered motion data. As shown in Fig. 6, when the mouth moved frequently with large variations, missing data and noise were often present in the captured data, so the animation produced using these data to reconstruct facial expressions was discontinuous and unnatural. To address this problem, we applied three separate recovery methods to our captured motion data. To validate our method's validity, we reconstructed facial motion expressions using Guenter's [2] algorithm with three neighbors, and we also conducted facial reconstruction with Lin and Ouyang's [1] method based on the weighted summation of the displacements neighbors to recover missing data. Finally, we complete the facial reconstruction process using our method based on the muscle motion mechanism with three neighbors. Figure 6 shows that the animation produced using our method was more natural and realistic than those obtained with the algorithms proposed by Guenter and Lin and Ouyang. In our method, we only use three neighbors to obtain fair results, which is a simple approach.

(2) Validation of the effect of preprocessing

To validate the effects of preprocessing, we performed additional experiments using the same conditions and the results are shown in Fig. 7.

In Fig. 7, the blue points are the captured points and the yellow points are the recognized impulsive noise. As shown in Fig. 7, impulsive noise was embedded in the captured motion data. The motion trajectory was not sufficiently smooth after curve fitting without preprocessing, which was mainly attributable to impulsive noise. However, it was smoother after using preprocessing to eliminate the impulsive noise.

(3) Validation of the effects of using high- or low-order B-spline curves for curve fitting

Curve fitting was completed by modifying the order of curve in the same conditions as the other experiments, and the results are shown in Fig. 8.

In Fig. 8, the motion trajectories of two points are assumed on and curve fitting was performed based on B-spline curves with different orders. As shown in Fig. 8, the curve-fitting results became smoother as the order of the curve increased, but the complexity

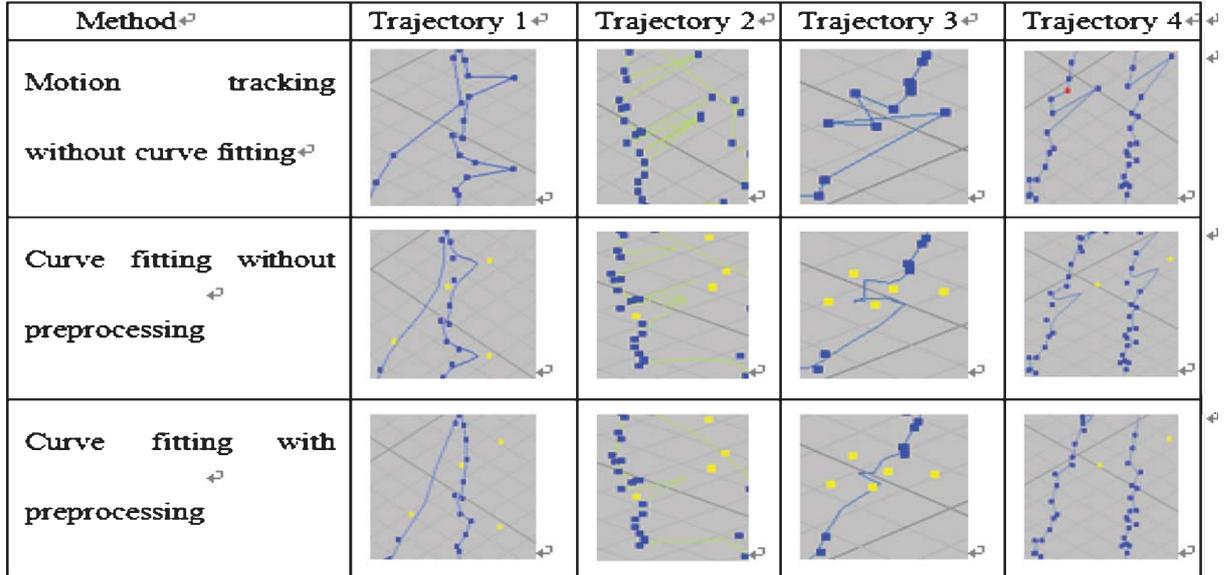


Fig. 7. Trajectories produced for four points using different methods.

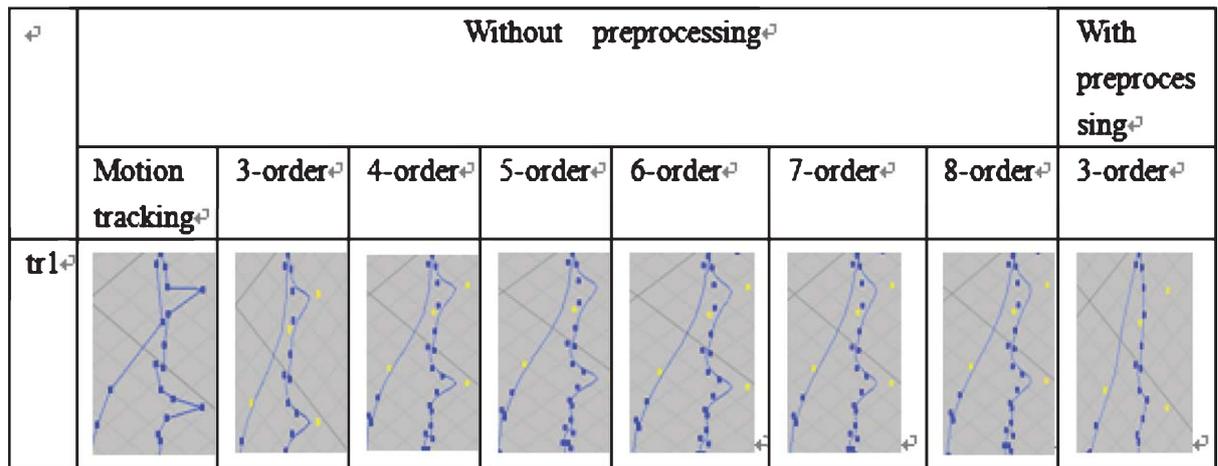


Fig. 8. Trajectories produced using two points with different methods.

also increased. After comparing the effects of low-order curve fitting with preprocessing, we found that the results obtained with low-order curve fitting and preprocessing were better than those with high-order curve fitting without preprocessing. Based on the computational complexity and the curve-fitting results, we used a third-order quasi-uniform B-spline to complete curve fitting with preprocessing. To support this conclusion, we calculated the variances of the vectors

grouped using two neighbors in the motion trajectory with and without preprocessing. The results are shown in Table 1.

As shown in Table 1, the variances of the motion trajectory without preprocessing became smaller as the order of the curve increased and the motion trajectory became smoother. However, the variance obtained using third-order curve fitting with preprocessing was smaller than that with high-order curve fitting without

Table 1
Comparison of the variances based on curve fitting with and without reprocessing

variance	Without preprocessing					With preprocessing
	3-order	4-order	5-order	6-order	7-order	3-order
Trajectory 1	0.0525081	0.0397194	0.0318661	0.0265171	0.0228267	0.0327899
Trajectory 2	0.0767762	0.0577493	0.0465663	0.0391085	0.0338817	0.0393396
Trajectory 3	0.0199221	0.0157452	0.0131446	0.0114090	0.0102553	0.0105272
Trajectory 4	0.0594217	0.465779	0.0394040	0.0348019	0.0316751	0.0387226
Trajectory 5	0.0874339	0.0771959	0.0709256	0.0666266	0.0635580	0.0658072
Trajectory 6	0.0552419	0.0494668	0.0460674	0.0437914	0.0422197	0.0431175

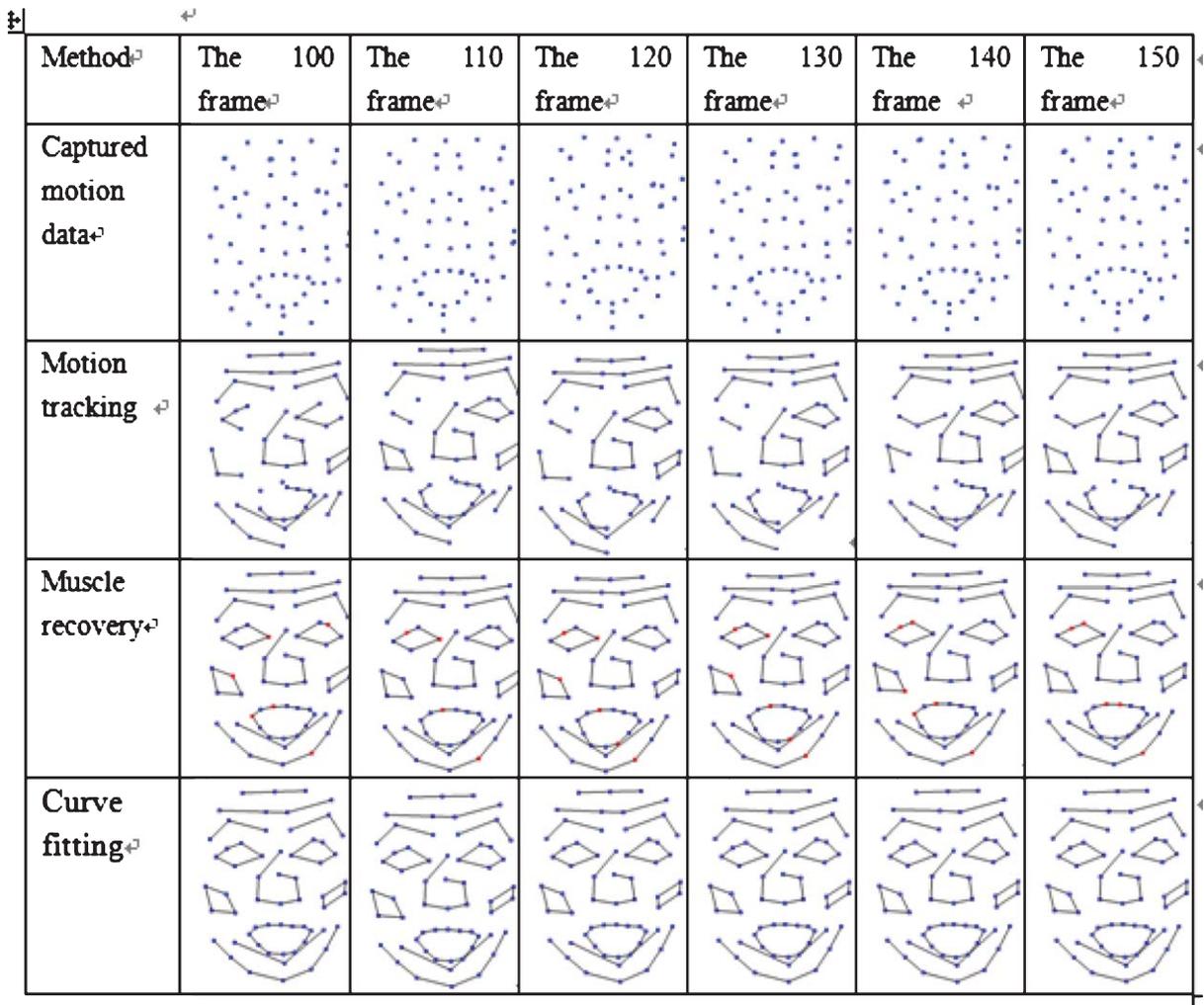


Fig. 9. Frames showing reconstruction using the overall motion data processing method.

preprocessing. Thus, given the simplicity of this approach and the fitting results, we reduced the remaining noise using a third-order B-spline curve with preprocessing.

(4) Validation of the reconstruction results obtained using the overall motion data processing method

In Fig. 9, the green points are the captured motion data and the red points are the recovered data. If a

point was missing, the lines between the point and its neighbors were not drawn. In Fig. 9, the captured motion data are scattered and there no relationships among them. Thus, we conducted motion tracking processing to obtain each point's trajectory and there were missing points in some frames. Using the tracked motion data, we recovered the missing points with our method. To generate a smooth facial motion trajectory, we reduced the noise using B-spline curve-based fitting and obtained fair results.

We conducted numerous facial motion data processing experiments using our system, including demonstrating the efficiency of preprocessing, the fit quality of B-spline curves with different order, the effects of motion data recovery using different methods, and assessments of the naturalness of the reconstructed animations produced by processing facial motion sequences.

In this study, we improved the muscle motion mechanism proposed by Waters [13] by using the movements of three neighbors to predict the movement of a missing point. Based on a mid-value filter, we conducted preprocessing to reduce impulsive noise. We also reduced the remaining noise using third-order quasi-uniform B-spline curve-based fitting, which obtained a smooth motion trajectory. Our overall method facilitated real-time animation and obtained fair facial motion reconstruction results. The demonstration video of the results can refer to Ref. [18].

6. Conclusions

In this study, we developed a 3D facial Mo-Cap system, where a data recovery method based on a muscle motion mechanism with three neighbors was used to recover missing data. We also developed a noise reduction method that used B-spline based curve fitting after preprocessing.

Based on tracked facial motion data, a muscle motion mechanism using three neighbors was applied to reconstruct the missing data. However, the data recovery approach proposed by Waters [13] used only one muscle to recover the movement of a missing point, which often led to problems with distortion because it ignored the correlations among neighbors. The success of our method is attributable to the use of the muscle motion mechanism and our exploitation of the rich correlations in captured facial motion data.

For impulsive noise preprocessing, a noise reduction method based on B-spline fitting was used to smooth

motion trajectory. This method automatically avoids, or at least minimizes, the effects of impulsive noise by preprocessing. Furthermore, the B-spline curve is a powerful and flexible fitting method, which produces a smooth motion trajectory. Preprocessing impulsive noise based on a mid-value filter also compensated for the weaknesses of the fitting results obtained using a low-order B-spline curve, thereby yielding a smooth motion trajectory.

Using these methods, we implemented the overall data processing procedure, which was highly efficient and it met the requirements by producing natural facial expressions. Fair results were obtained using our method but there is still a problem because our method requires manual intervention. The key focus of our future research is to design a robust method that can be used for marker-less facial deformation-based tracking.

Acknowledgements

This work is supported by the Program for Changjiang Scholars and Innovative Research Team in University (No. IRT1109), the National Natural Science Foundation of China (Nos. 61370141, 61300015), the Program for Liaoning Science and Technology Research in University (No. LS2010008), the Program for Liaoning Innovative Research Team in University (No. LT2011018), Natural Science Foundation of Liaoning Province (No. 2013020007), the Program for Liaoning Excellent Talents in University (No. LJQ2013132), the Program for Liaoning Key Lab of Intelligent Information Processing and Network Technology in University and by "Liaoning BaiQianWan Talents Program (No. 2011921009)".

References

- [1] I.C. Lin, M. Ouhyoung, Mirror Mo-Cap, Automatic and efficient capture of dense 3D facial motion parameters from video, *The Visual Computer* **21**(6) (2005), 355–372.
- [2] B. Guenter, C. Grimm and D. Wood, et al., Making faces, *International Conference on Computer Graphics and Interactive Techniques*, ACM SIGGRAPH 2005 Courses, Los Angeles, California, 2005, 31.
- [3] J.D. Edge, M.A. Sánchez, S. Maddock, Animating speech from motion fragments. *Technical Report CS-04-02*, Department of Computer Science, University of Sheffield.
- [4] A. Aristidou, J. Cameron, J. Lasenby, Predicting missing markers to drive real-time centre of rotation estimation. Department of Engineering, University of Cambridge, 2008.

- [5] L. Xiaoli, C. Chunmei, Segment curve fitting method based on least square principle. guang dong: Foshan technich scientific technology college, 2004.
- [6] C.-C. Hsieh, P.L. Kuo, *An impulsive noise reduction agent for rigid body motion data using B-spline wavelets*. Department of Industrial and Information Management, Nataional Cheng Kung University, I, Ta-Sheuh Road, Tainan 70101, Taiwan, ROC 2007.
- [7] T. Horsch and B. Juttler, Cartesian spline interpolation for industrial robots, *Computer Aided Design* **30**(3) (1998), 217–224.
- [8] A. Somasundaram, R. Parent, A facial animation system for expressive audio-visual speech. OSU-CISRC-4/06-TR46, Department of Computer Science and Engineering, The Ohio State University: Columbus, OH, 2006.
- [9] X.Y. Fang, X.P. Wei, Q. Zhang and D.S. Zhou, *Forward non-rigid motion tracking for facial MoCap*, The Visual Computer, 2013. doi: 10.1007/s00371-013-0790-8
- [10] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, M. Gross, Multi-Scale Capture of Facial Geometry and Motion, *Proceedings of ACM SIGGRAPH (San Diego, USA, August 5-9, 2007)*, *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 33.1–33.10.
- [11] T. Weise, S. Bouaziz, H. Li, M. Pauly, Realtime performance-based facial animation. *ACM Transactions on Graphics*, SIGGRAPH, 2011.
- [12] I. Akhter, T. Simon, S. Kahn, I. Matthews, Y. Sheikh, Bilinear spatiotemporal basis models. *Proc. SIGGRAPH*, 2012.
- [13] K. Waters, A muscle model for animating three-dimensional facial expression, *SIGGRAPH '87 Proceedings of the 14th annual conference on Computer graphics and interactive techniques* **21** (1987), 17–24.
- [14] J.H. Clark, *3-D Design of Free-Form B-Spline Surfaces*, Ph. D Thesis, University of Utah, Salt Lake City, Utah, 1974.
- [15] J.H. Clark, Some properties of B-splines. *Proc.2nd USA-Japan Computer Conference, AFIPS*, Montvale, N.J., (1975), pp. 542–545.
- [16] Z. Xinxiong, *Freedom curve sculpt technique*. Beijing: Science Press, 2001.1.
- [17] Dalian Doreal Software Company. [2008-09-30]. <http://www.dorealsoft.com/>.
- [18] Key Laboratory of Advanced Design and Intelligent Computing (Dalian University), Ministry of Education [2013-12-08]. <http://202.199.159.247/download.asp>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

