# A ROBUST METHOD FOR DETECTION OF OCCLUSAL POSITION

*Masayoshi Kanoh, Kyoji Hashimura, Shohei Kato and Hidenori Itoh*

Nagoya Institute of Technology
Dept. of Intelligence and Computer Science
Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan
email: {kanoh, bridge, shohey, itoh}@ics.nitech.ac.jp

## ABSTRACT

We have developed a system that enables dental prostheses to be designed on a computer. The occlusal position for the upper and lower jaw objects should be found when designing dental prostheses on our system. However, there is often great difficulty in manually finding this position. To reduce this difficulty, we considered finding it and occluding jaw objects automatically. In positioning upper and lower jaw objects, some regions, such as teethridges and sides of teeth, do not completely close on one another, even if these objects are occluded. The evaluation value for these regions is an error, and consequently finding the occlusal position requires robust assessment. In this paper, we propose a method of evaluation using a robust estimator, the M-estimator. Applying multi-resolution representation, we also attempt to improve runtime.

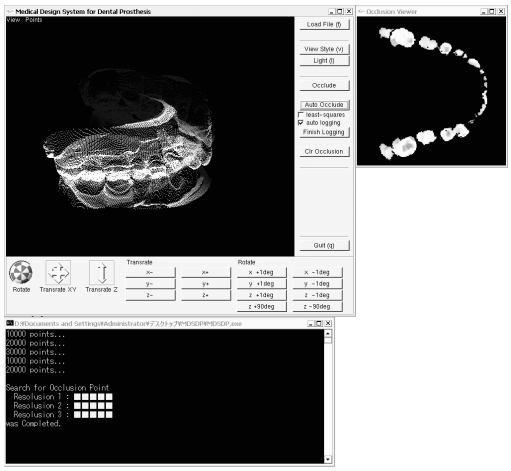


**Fig. 1**. System overview.

## 1. INTRODUCTION

In the process of designing a dental prosthesis, dental technicians first adjust the occlusion in the dental gypsum of a patient. They then make a plastic model of the tooth being treated from the adjusted gypsum. Finally, they cast the plastic model into a dental prosthesis. This process is usually done manually. This process therefore requires skillful technique of dental technicians. Here, we have considered a cost-saving technique that enables occlusions to be adjusted, and that can make the process more efficient. We have, thus, developed a system that enables dental prostheses to be designed on a computer (Fig. 1).

In designing dental prostheses on our system, dental technicians need to find the occlusal position for the upper and lower jaw objects, before they make the prosthesis. However, finding this position manually is extremely difficult. To reduce the work involved, we have considered a way of finding this position and occluding jaw objects automatically. In this paper, we describe a process for estimating the occlusal position, and propose a method of evaluating occlusion.

Efficient algorithms for detecting similar regions have been proposed by Kanoh et al. [1][2]. However, these can only be used to detecting the most similar pair from 2D contours. Jaw objects have a 3D shape, therefore these algorithms cannot be used to estimate the occlusal position.

Various approaches to detect similar regions in 3D have been proposed.

Hashimura et al. [3][4] extracted sub-surfaces called character surfaces, which they considered to be candidates for the occlusion, and then, detected the most similar pair of surfaces. It is very difficult to extract character surfaces skillfully estimating the occlusal position, because those of the upper and lower objects, which touch each other when occluded, have complex shapes. Therefore, these approaches cannot be applied to estimating the occlusal position.

Papaioannou et al. [5] detected the most similar pair from two objects in the following way. They first made a plane between the two objects. They then extracted surfaces which faced one another on the plane from the figures of the objects projected on the plane. Finally, they detected the most similar pair from these surfaces. In occlusion, these

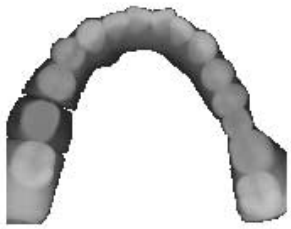

**Fig. 2**. Range image.

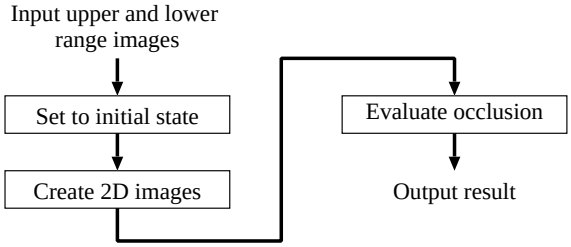


**Fig. 3**. Process overview.

surfaces include regions such as teethridges and sides of teeth. Their method established that the evaluation value for these regions was an error, because these regions do not close even if the objects are occluded. Therefore, as this value is unreliable, this method cannot correctly evaluate occlusion. In this paper, we propose a method of evaluation, that does robust assessments by saturating the evaluation error for these regions. We adopted the M-estimator to do this, and also applied multi-resolution representation to improve the runtime.

Collision detection is required when evaluating occlusion, and in the above approaches [3][4], mesh data was used. If mesh data is used to evaluate occlusion, this needs to be extremely precise. However, creating highly precise mesh data from jaw objects requires enormous meshes, which increases computational costs. Our method uses range images to reducing these costs.

## 2. DATA MEASUREMENT

Our system utilized range images measured with a range finder, MDX-15 produced by Roland DG Corporation. This finder enables a scanning pitch of 0.025 mm ($z$ axis) and 0.05 mm to 5 mm ($x/y$ axis). We used a value of 0.25 mm for the scanning pitch of $x/y$ axis. Fig. 2 is a range image derived from this finder, and it has 20,810 points.

## 3. PROCESS TO ESTIMATE OCCLUSAL POSITION

Estimating the occlusal position requires the detection of collision. However, range images have no surface information. As existing methods (e.g. [6][7][8]) only register range images, and cannot detect collision, we need a way of doing this.

Here, let us see the actual occlusion of jaws. Jaws occlude in the direction of motion. Additionally, the collision distribution part is nearly flat. Applying these characteristics, we avoided the collision problem by using 2D images to evaluate occlusion.

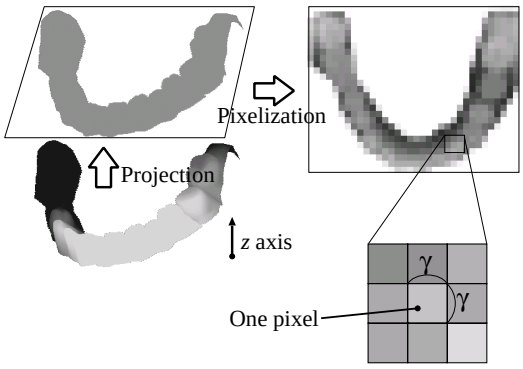


**Fig. 4**. Flow for creating 2D image (lower jaw).

### 3.1. Process Overview

Fig. 3 is an overview of the process for estimating the occlusal position. First, the range images of the upper and lower jaws are set to an initial state. Second, two 2D images are created from the upper and lower range images. Third, occlusion between the range images is evaluated by computing the similarity between the corresponding 2D images. Evaluation is done by moving the lower range image. Finally, a position is selected that has the most similarity.

### 3.2. Setting to Initial State

In this process, dental technicians manually set a pair of upper and lower range images to the initial state by moving the lower range image. The setting has to satisfy a condition where the minimal $z$ coordinate of the upper range image is larger than the maximal $z$ coordinate of the lower range image. Fig. 8 shows an example of the initial state.

### 3.3. Creating 2D Image to Evaluate Occlusion

Our method is used to evaluate occlusion through 2D images created by projecting range images on the $x$-$y$ plane. Fig. 4 has the flow for creating a 2D image.

First, let us define a rule for the pixelization scale to create the 2D image.

**Definition 3.1 pixelization scale**
Let us consider an image with pixels $\gamma \times \gamma$-wide on the $x$-$y$ plane. We call this an image at *pixelization scale* $\gamma$. □

Small images, i.e. low resolution, are created if the pixelization scale is large. High resolution images, on the other hand, are created if the pixelization scale is small. Images of any resolution can be created by controlling the pixelization scale.

The calculation of the color value for the upper images at pixelization scale $\gamma$ is given as follows.

**Definition 3.2** Let $\mathcal{A}$ be a upper range image. Let $(x, y, z)^T \in \mathcal{A}$ be a point constructing $\mathcal{A}$. Further, let $\mathbf{A}^\gamma$ be an image at pixelization scale $\gamma$ created from $\mathcal{A}$. Then, the color value of a pixel $(i, j)$ in $\mathbf{A}^\gamma$ is defined as:

$$\mathrm{a}^\gamma_{i,j} = \min_{(x,y,z)^T \in \mathcal{A}} z, \quad (\gamma i \leq x < \gamma(i+1),\ \gamma j \leq y < \gamma(j+1)). \quad (1)$$

However, if there are no pixels assigned in the $xy$-area by this equation, the color value is defined as $\mathrm{a}^\gamma_{i,j} = \infty$. □

The calculation for the lower images differs from that for the upper images, and is given as follows.

**Definition 3.3** Let $\mathcal{B}^{\boldsymbol{R}}$ be a lower range image rotated by rotational matrix $\boldsymbol{R}$. Let $(x, y, z)^T \in \mathcal{B}^{\boldsymbol{R}}$ be a point constructing $\mathcal{B}^{\boldsymbol{R}}$. Further, let $\mathbf{B}^{\boldsymbol{R},\gamma}$ be an image at pixelization scale $\gamma$ created from $\mathcal{B}$. Then, the color value of a pixel $(i, j)$ in $\mathbf{B}^{\boldsymbol{R},\gamma}$ is defined as follows:

$$\mathrm{b}^{\boldsymbol{R},\gamma}_{i,j} = \max_{(x,y,z)^T \in \mathcal{B}^{\boldsymbol{R}}} z, \quad (\gamma i \leq x < \gamma(i+1),\ \gamma j \leq y < \gamma(j+1)). \quad (2)$$

However, if there are no pixels assigned in the $xy$-area by this equation, the color value is defined as $\mathrm{a}^\gamma_{i,j} = -\infty$. □

Using the color value, we can detect collision in the direction of the $z$ axis between the upper and lower jaws. The color value enables us to evaluate occlusion and detect collision simultaneously.

### 3.4. Method to Evaluate Occlusion with M-estimator

Our method use the all points of range images to estimate the occlusal position, because we do not extract surfaces to evaluate occlusion. The 2D images created in the manner we prescribed have color values for regions such as teethridges and sides of teeth that greatly differ from the other areas. Therefore, if the least squares method is used to estimate the occlusal position, the evaluation value in those regions is detected as large error. To saturate the error evaluated in those regions, we adopt a robust estimator, the M-estimator, to assess occlusion in this paper.

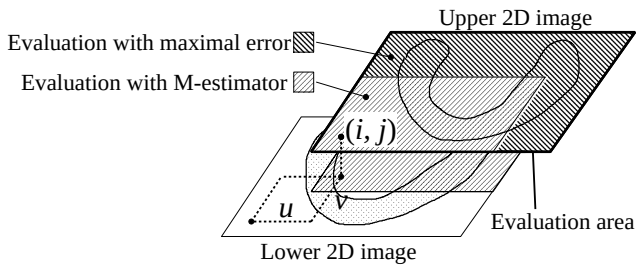


**Fig. 5**. Evaluation of occlusion with 2D image.

We define the M-estimators to evaluate occlusion as follows.

**Definition 3.4** Let us consider a 2D image pair $(\mathbf{A}^\gamma, \mathbf{B}^{\boldsymbol{R},\gamma})$ with pixelization scale $\gamma$. When all pixels in $\mathbf{B}^{\boldsymbol{R},\gamma}$ are moved by $\boldsymbol{t} = (u, v)$ (see Fig. 5), the color difference at a point $(i, j)$ is calculated in the following manner:

$$\varepsilon^{\boldsymbol{R},\gamma,\boldsymbol{t}}_{i,j} = \mathrm{a}^\gamma_{i,j} - \mathrm{b}^{\boldsymbol{R},\gamma}_{i+u,j+v}. \quad (3)$$

Then, the M-estimators between $\mathbf{A}^\gamma$ and $\mathbf{B}^{\boldsymbol{R},\gamma}$ moved by $\boldsymbol{t}$ are defined as follows:

$$\begin{aligned} \mathcal{M}(\boldsymbol{R}, \gamma, \boldsymbol{t}) &= \sum_i \sum_j \rho(\varepsilon^{\boldsymbol{R},\gamma,\boldsymbol{t}}_{i,j} - \varepsilon^{\boldsymbol{R},\gamma,\boldsymbol{t}}_{\min}, \sigma), \quad (4) \\ \varepsilon^{\boldsymbol{R},\gamma,\boldsymbol{t}}_{\min} &= \min_{k,l} \varepsilon^{\boldsymbol{R},\gamma,\boldsymbol{t}}_{k,l}, \end{aligned}$$

where $\sigma$ is a control parameter, and $\rho$ is a robust loss function. □

When the value of the estimators decreases, the occlusal position is evaluated as a better position.

To saturate error, we applied the Geman-McClure function [9][10] to the robust loss function:

$$\rho(x, \sigma) = \frac{x^2}{\sigma + x^2}. \quad (5)$$

The function of $\sigma = 1.0$ is plotted in Fig. 6. Using this function, we can assign a value of $1.0$, which is the saturated value of this function, to the evaluation value of pixels whose color values are infinity.

Here, we will describe the area we used to evaluate occlusion, which was the entire upper part of 2D image (Fig. 5). Where the upper and lower 2D images overlap each other, we used the M-estimators for the evaluation value. Where the upper and lower 2D images do not overlap, we used maximal error for the evaluation value. We can assign a value of $1.0$ to the maximal error, because we use the Geman-McClure function.

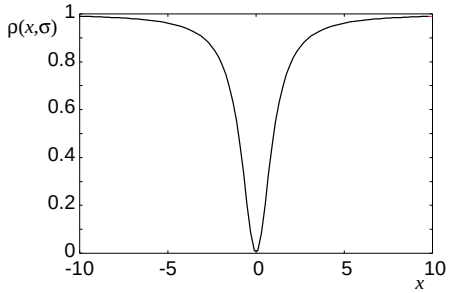


**Fig. 6**. $\rho$ function of Geman and McClure ($\sigma = 1.0$).

## 4. FASTER EVALUATION USING MULTI-RESOLUTION REPRESENTATION

The cost of estimating the optimal occlusal position increases when the pixelization scale for 2D images decrease. In this paper, we propose an algorithm for evaluating occlusion with multi-resolution representation, which can efficiently detects a quasi-optimal occlusal position.

Decreasing pixelization scale, this algorithm advances the search for evaluating occlusion.

Fig. 7 details our proposed algorithm.

Here, correct() rectifies the amount of translation according to the change of the pixelization scale. This function is given as:

$$\text{correct}(\gamma, \gamma', \boldsymbol{t}) = \frac{\gamma}{\gamma'}\boldsymbol{t}. \tag{6}$$

$S^{\boldsymbol{R},\gamma}$ is the search space for rotation, and is defined as follows.

**Definition 4.1** Let $\boldsymbol{X}_\theta$, $\boldsymbol{Y}_\theta$ and $\boldsymbol{Z}_\theta$ be the rotational matrices rotating with $\theta$ around the $x, y$ and $z$ axis, respectively. Further, let $\boldsymbol{R}$ be a given rotational matrix that is the basis for creating search space. Then, search space $S^{\boldsymbol{R},\gamma}$ for rotation at pixelization scale $\gamma$ is defined as:

$$\begin{aligned} S^{\boldsymbol{R},\gamma} \quad = \quad & \{\boldsymbol{R}\boldsymbol{X}_{\delta_x\gamma\theta_x}\boldsymbol{Y}_{\delta_y\gamma\theta_y}\boldsymbol{Z}_{\delta_z\gamma\theta_z} \mid \\ & -n \le \delta_x \le n, -n \le \delta_y \le n, \\ & -n \le \delta_z \le n\}, \end{aligned} \tag{7}$$

where $n$ is the parameter to determine the search area, and $\theta_x$, $\theta_y$ and $\theta_z$ are unit angles around the $x$,$y$ and $z$ axis respectively. □

$S^{\boldsymbol{t}}$ is the search space for translation, and is defined as follows.

**Definition 4.2** Let $\boldsymbol{t}$ be a given two dimensional vector that is the basis for creating search space. Then, search space $S^{\boldsymbol{t}}$

**Detection of Occlusal Position**

Input :upper and lower range images, $\mathcal{A}$ and $\mathcal{B}$
Output :transformation of lower jaw $(\boldsymbol{R}, \boldsymbol{t})$

1 **begin**
2 $k := N$; % $k$ is parameter for pixelization scale
% $N$ is number of resolutions
3 $\boldsymbol{R} := \boldsymbol{I}$; % $\boldsymbol{I}$ is unit matrix
4 decide on $\boldsymbol{t}$ from $\gamma_k$ and positions of $\mathcal{A}$ and $\mathcal{B}$;
5 **repeat**
6 $error := \infty$;
7 $\boldsymbol{R}'' := \boldsymbol{R}$;
8 $\boldsymbol{t}'' := \boldsymbol{t}$;
9 create $\mathbf{A}^{\gamma_k}$;
10 **for each** $\boldsymbol{R}' \in S^{\boldsymbol{R},\gamma_k}$
11 **begin**
12 create $\mathbf{B}^{\boldsymbol{R}',\gamma_k}$;
13 **for each** $\boldsymbol{t}' \in S^{\boldsymbol{t}}$;
14 **begin**
15 $e := \mathcal{M}(\boldsymbol{R}', \gamma_k, \boldsymbol{t}')$;
16 **if** ($e < error$) **then**
17 **begin**
18 $error := e$;
19 $(\boldsymbol{R}'', \boldsymbol{t}'') := (\boldsymbol{R}', \boldsymbol{t}')$;
20 **end**
21 **end**
22 **end**
23 $(\boldsymbol{R}, \boldsymbol{t}) := (\boldsymbol{R}'', \boldsymbol{t}'')$;
24 **if** ($\gamma_k = \gamma_1$) % $\gamma_1$ is minimal pixelization scale
25 **then** $escape := true$;
26 **else begin**
27 $\boldsymbol{t} := \text{correct}(\gamma_k, \gamma_{k-1}, \boldsymbol{t})$;
28 $k := k - 1$;
29 **end**
30 **until** ($escape$)
31 **end**.

**Fig. 7**. Evaluation algorithm for estimating occlusal position with multi-resolution representation.

for translation at pixelization scale $\gamma$ is defined as:

$$\begin{aligned} S^{\boldsymbol{t}} \quad = \quad & \{\boldsymbol{t} + (\delta_i, \delta_j) \mid \\ & -m \le \delta_i \le m, -m \le \delta_j \le m\}, \end{aligned} \tag{8}$$

where $m$ is the parameter to determine the search area. □

Our algorithm can be used evaluate occlusion with changing pixelization scale from maximum $\gamma_N$ to minimum $\gamma_1$. First, the upper 2D image is created and the search area for rotation is determined (see *ll.* 9 and 10 in Fig. 7). Second, the lower 2D image is created and the search area for translation is determined (see *ll.* 12 and 13 in Fig. 7). Then, to searching for a pair $(\boldsymbol{R}'', \boldsymbol{t}'')$ that has the smallest error in the given search area, the evaluation process is iterated (see *ll.* 10 – 22 in Fig. 7). At minimal pixelization scale $\gamma_1$,


**Fig. 8**. Initial state.

**Fig. 9**. Estimated state.

(a) 0-1.0 mm (b) 0-0.3 mm (c) 0-0.6 mm (d) 0-0.9 mm

**Fig. 10**. Occlusal state.

iteration is finished, and $(\boldsymbol{R}, \boldsymbol{t})$ is output, and if this is not $\gamma_1$, $\boldsymbol{t}$ is corrected, $k$ is decreased, and the search continues (see *ll*. 24 – 29 in Fig. 7)

When 2D images are only moved in translation in our algorithm, we do not need to re-create them, and occlusion can be evaluated simply by moving them. Therefore, our algorithm can reduce the matrix calculation needed to evaluate occlusion.

## 5. EXPERIMENTAL RESULTS

We did experiments on estimating the occlusal position to verify effectiveness of our algorithm. In these experiments, in order for the evaluation error to be 0.9 when the distance between the upper and lower jaws was 1.0 mm, we used a value of $\sigma = 1.0$. We also used values of $N = 3$, $\{\gamma_k\}_{k=1,\cdots,N} = \{0.3, 0.6, 0.9\}$, $n = 2$, $m = 2$ and $\theta_x = \theta_y = \theta_z = \pi/30$ for the parameters in our algorithm.

When the initial state is that depicted in Fig. 8, the estimated occlusal position with our algorithm is that in Fig. 9. Our algorithm took about 30 seconds for the estimation (Pentium III / 500 MHz). Fig. 10 shows the occlusal state of Fig. 9 which is denoted by gray-scale images. In the figure, (a) denotes the distance between the upper and lower jaws in a range from 0 to 1.0 mm. When the gray-scale approaches black, the distance decreases. Additionally, (b), (c) and (d) denote ranges from 0 to 0.3 mm, 0 to 0.6 mm and 0 to 0.9 mm indicated by black color, respectively.

In the next experiment, we compared our algorithm with the least squares method, in order to assess the robustness of our algorithm. The same as the least squares method, we changed the $\rho$ function of the M-estimator of our algorithm into $\rho(x) = x^2$. In the evaluation area with maximal error, we used a value of 2,500 for maximal error. Moreover, in the evaluation area with the least squares method, when the color values of 2D image pixels were infinity, we also used a value of 2,500 for the evaluation values of these pixels, because the least squares method cannot be used to evaluate pixels whose color values are infinity. We determined a value of 2,500 from the fact that the greatest distance between the upper and lower jaws was shorter than 50 mm when the jaws were occluded.

Fig. 11 and 12 compare our algorithm with the least squares method. In these figures, (a) is the initial state, (b)(c) and (d) are the estimated states with our algorithm, and (e)(f) and (g) are the estimated states with the least squares method. These results indicate that our algorithm can provide a better estimate of the occlusal position than the least squares method. It greatly inhibits evaluation errors, and provides robust evaluations of occlusion.

## 6. CONCLUSION

We proposed an evaluation method that enabled the occlusal position to be estimated with the M-Estimator in this paper. It provided robust assessment by saturating evaluation error. We also proposed an algorithm with multi-resolution representation to improve the estimation runtime. In feature work, we intend to propose a method of designing dental prostheses.

## 7. REFERENCES


[1] Masayoshi Kanoh, Shohei Kato, and Hidenori Itoh, "Efficient joint detection considering complexity of contours," *Lecture Notes in Artificial Intelligence*, vol. 1886, pp. 588–598, 2000.

[2] Masayoshi Kanoh, Shogo Yasuhara, Shohei Kato, and Hidenori Itoh, "Similarity-based approach to earthenware reconstruction," *11th International Conference on Image Analysis and Processing(ICIAP 2001)*, pp. 478–483, 2001.

[3] Kyoji Hashimura, Shohei Kato, and Hidenori Itoh, "Finding a quasi-optimal joining position of potsherds in 3d multi-resolution mesh representation," *2nd International Conference on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing*, pp. 677–684, 2001.

[4] Kyoji Hashimura, Shohei Kato, and Hidenori Itoh, "A parametric search strategy for a quasi-optimal joining

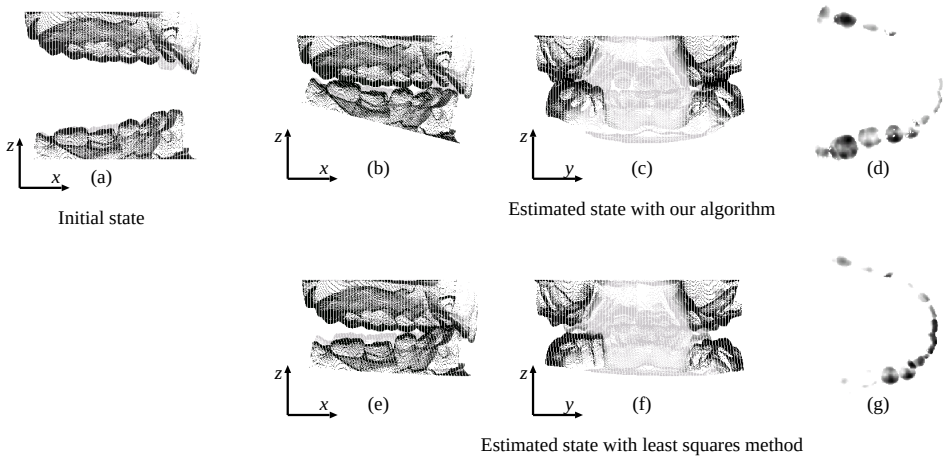


**Fig. 11**. Comparison of our algorithm and least squares method (1): (a) is initial state; (b), (c) and (d) are estimated states with our algorithm; (e), (f) and (g) are estimated states with least squares method.

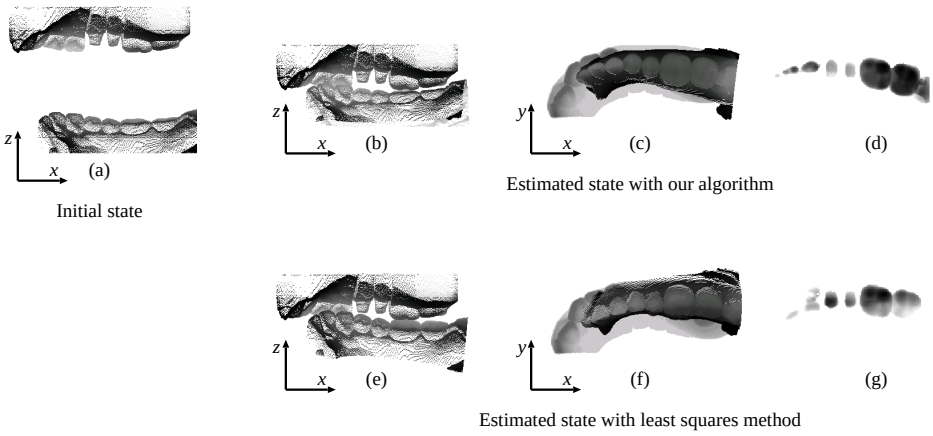


**Fig. 12**. Comparison of our algorithm and least squares method (2): (a) is initial state; (b), (c) and (d) are estimated states with our algorithm; (e), (f) and (g) are estimated states with least squares method.

position of 3d potsherds," *International Journal of Computer & Information Science*, vol. 3, no. 2, pp. 94–104, 2002.

[5] Georigos Papaioannou, Evaggelia-Aggeliki Karabassi, and Theoharis Theoharis, "Reconstruction of three-dimensional objects through matching of their parts," *IEEE Trans. PAMI*, vol. 24, no. 1, pp. 114–124, 2002.

[6] Ko Nishino and Katsushi Ikeuchi, "Robust simultaneous registration of multiple range images," *Asian Conference of Computer Vision*, pp. 454–461, 2002.

[7] P.J. Besl and N.D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. PAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[8] Zhengyou Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.

[9] S. Geman and D. E. McClure, "Statistical methods for tomographic image reconstruction," *Bulletin of the International Statistical Institute*, pp. 5–21, 1987.

[10] M. J. Black and P. Anandan, "The robust estimation of multiple motions parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.