# Real-time Modeling of Face Deformation for 3D Head Pose Estimation

Kenji Oka and Yoichi Sato

Institute of Industrial Science, The University of Tokyo
4–6–1 Komaba, Meguro-ku, Tokyo, 153–8505, Japan
ysato@iis.u-tokyo.ac.jp

**Abstract.** We propose a new technique for simultaneously executing face deformation modeling and 3D head pose estimation. Previous methods for estimating 3D head pose require a preliminary training stage for the head model, and cannot start tracking the head pose until this stage is complete. In contrast, our proposed method can acquire and refine a user's deformable head model in parallel with tracking the user's head pose. This allows progressive improvement in the accuracy of the estimation of head pose and face deformation.
Our technique consists of three main steps. In the first step we estimate the 3D head pose using a head model that is obtained automatically. The second step finds true positions of feature points by using the resulting poses of the first step. Finally, the basis vectors of face deformation are calculated from the true positions of feature points to acquire a new deformable head model as a linear combination of the basis vectors.
The iteration of the three steps refines the deformable head model, thus improving the accuracy of head pose estimation progressively. The improvement has been successfully demonstrated via experiments.

## 1 Introduction

Tracking of 3D head pose is regarded as an important topic in computer vision. So far a number of researchers have developed methods for estimating 3D head pose. Many of those methods employ a rigid model that can only deal with 3D translation and rotation [1, 7, 14, 16, 20, 17]. Actually, the human face is often deformed significantly due to various factors, for example, change of facial expression, which causes deterioration of accuracy or failure of tracking.

This has motivated work that uses a model to represent deformation of a user's face, that is, a deformable head model. Black and Yacoob segment the human face into rigid parts and deformable parts, and estimate head pose and face expression with the segmented model [2]. Several researchers use 3D deformable surface models, for example mesh models, for estimating fine deformation of a user's face [10, 6, 3, 18, 5]. Recently, the Active Appearance Model (AAM) and similar methods have been studied by many researchers. Matthews and Baker presented good survey on AAM [15]. While those methods have the potential for good estimation, some sort of 3D geometrical model, for example the deformable

head model itself, must be prepared with manual feature extraction or 3D laser scanning. An approach for solving this problem is automatic tracking of several feature points, and then analyzing the coordinates of the tracked points for acquiring basis shape vectors of face deformation. In this approach, Gokturk et al. utilizes the Principal Component Analysis (PCA) [8], and Del Bue et al. makes use of the non-rigid factorization technique [4].

The previously proposed methods that use a deformable head model as described above have a common problem: they require a preliminary stage to acquire the head model. Those methods cannot start real-time tracking of 3D head pose before completing that stage, and they do not have a framework for refining the deformable head model using estimation results.

In this paper, we propose a new method for acquiring and refining a user's deformable head model in parallel with estimating the user's head pose in real time. This means that our method requires no cumbersome preparation for constructing a head model. The method for acquiring a deformable head model consists of three steps. In the first step we estimate the 3D head pose and the face deformation using a head model that is obtained automatically. Second, we find true positions of feature points by using the resulting poses of the first step. Finally, the basis vectors of face deformation are calculated from the true positions of the feature points to acquire a new deformable head model as a linear combination of the basis vectors. Since the newly acquired model is used for the next estimation of the 3D head pose and face deformation, our method can progressively improve the accuracy of estimating pose and deformation.

The main contributions of our study are summarized in the following three points: 1) real-time estimation of 3D head pose without a preliminary training stage, 2) real-time refinement of a deformable head model, and 3) progressive improvement of the accuracy of estimating head pose and face deformation. The improvement has been successfully demonstrated via experiments.

The reminder of this paper is organized as follows. In Section 2, we describe our method for estimating head pose and face deformation. We then propose a method for acquiring a deformable head model in Section 3. We show the experimental results of our method in Section 4. Finally, we conclude this paper in Section 5.

## 2 Real-time Estimation of 3D Head Pose with Deformable Head Model

In this section, we describe our method for estimating 3D head pose and deformation from image inputs from two calibrated cameras[1], the left camera and the right camera, that incorporate a deformable head model.

[1] Although we assume a two-camera configuration here, we can increase the number of cameras without altering the algorithm of our proposed method.

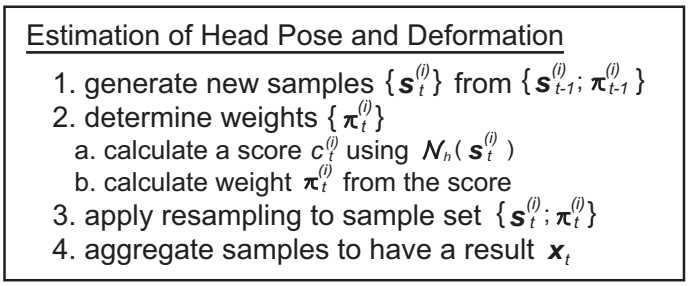


**Fig. 1.** Flow of estimating head pose and deformation

### 2.1 Deformable Head Model

In our method, the head model has $K$ feature points, and each feature point consists of two components: the 3D position in the model coordinate system fixed to a user's head at the frame $t$, and two small image templates. Let $\boldsymbol{M}_t$ be the $3K$-dimensional shape vector that consists of 3D coordinates of $K$ feature points in the model coordinate system. Also, $T_L$ and $T_R$ are defined as the image template sets for the left camera and the right camera respectively. Here, $K$ is set to 10 to represent these ten feature points: the inner and outer corners of both eyes, both corners of the mouth, both nostrils, and the inner corner of both brows.

The shape vector $\boldsymbol{M}_t$ of our deformable head model is formulated as:

$$\boldsymbol{M}_t = \bar{\boldsymbol{M}} + \boldsymbol{\mathcal{M}}\boldsymbol{a}_t \tag{1}$$

where $\bar{\boldsymbol{M}}$ is the mean shape vector, $\boldsymbol{\mathcal{M}}$ is the $3K \times B$ basis shape matrix, which consists of $B$ columns of the basis shape vectors, and $\boldsymbol{a}_t$ is a $B$-dimensional coefficient vector of $\boldsymbol{\mathcal{M}}$. Here, the shape $\boldsymbol{M}_t$ is represented as a linear combination of the constant basis shape vectors corresponding to the columns of $\boldsymbol{\mathcal{M}}$ in a similar way to other methods [8, 15, 5]. The limited size of $B$, $B = 5$ in this method enables us to represent the face deformation by a small number of parameters in $\boldsymbol{a}_t$, We will describe how the basis matrix $\boldsymbol{\mathcal{M}}$ and the mean vector $\bar{\boldsymbol{M}}$ are obtained later in Section 3.

### 2.2 Particle Filter for Estimating Head Pose and Face Deformation

During tracking we produce successive estimation of a $(6+B)$ dimensional state vector $\boldsymbol{x}_t = (\boldsymbol{p}_t^{\mathrm{T}}, \boldsymbol{a}_t^{\mathrm{T}})^{\mathrm{T}}$ for each image frame $t$. Here, $\boldsymbol{p}_t$ is the translation and the rotation from the world coordinate system to the model coordinate system. For pose estimation we make use of the deformable head model and the particle filtering technique.

A particle filter [9] represents the probability density function (PDF) of a state as a set of many discrete samples, each sample with a corresponding

weight. Hence, this sample set can approximate an arbitrary PDF including non-Gaussian ones. Our method uses the sample set $\{(\boldsymbol{s}_t^{(i)}; \pi_t^{(i)})\}(i = 1 \ldots N)$, which consists of $N$ discrete samples $\boldsymbol{s}_t^{(i)}$ in the $(6+B)$ dimensional state space and their corresponding weights $\pi_t^{(i)}$.

The main flow of our estimation method is shown in Fig.1. We first generate $N$ new samples $\{\boldsymbol{s}_t^{(i)}\}$ based on the sample set $\{(\boldsymbol{s}_{t-1}^{(i)}; \pi_{t-1}^{(i)})\}$ and the following motion model on the assumption of a uniform straight motion of a user's head between each pair of successive image frames:

$$\boldsymbol{s}_t^{(i)} = \boldsymbol{s}'_{t-1} + \tau \boldsymbol{v}_{t-1} + \boldsymbol{\omega} \tag{2}$$

where $\boldsymbol{s}'_{t-1}$ is a chosen sample from $\{(\boldsymbol{s}_{t-1}^{(i)}; \pi_{t-1}^{(i)})\}$, $\tau$ is the time interval between frames, $\boldsymbol{v}_{t-1}$ represents the velocity of the pose that is calculated at the end of the previous estimation step $t-1$, and $\boldsymbol{\omega}$ is system noise. In addition, $\boldsymbol{\omega}$ is a $(6+B)$ dimensional Gaussian noise vector with a zero mean, and the upper-left $6 \times 6$ elements of its covariance matrix corresponding to the pose parameters are adaptively controlled depending on the velocity of the head. We have found that such control of system noise improves the robustness against sudden abrupt motion while maintaining the high accuracy of estimating head pose at the same time [17]. On the other hand, the rest of the covariance matrix is a diagonal matrix whose diagonal elements are represented by a $B$-dimensional constant vector $\boldsymbol{\beta}$. Each element of $\boldsymbol{\beta}$ is proportional to the corresponding element of the standard deviation vector $\boldsymbol{\mu}$ of $\boldsymbol{a}_t$ which is calculated by PCA as explained in Section 3.3.

After we obtain new samples $\{\boldsymbol{s}_t^{(i)}\}$ we compute the weight $\pi_t^{(i)}$ by evaluating each sample $\boldsymbol{s}_t^{(i)}$ based on the set of current input images.

Given a sample $\boldsymbol{s}_t^{(i)}$, we apply the normalized correlation-based function $\mathcal{N}_h(\boldsymbol{s}_t^{(i)})$ with the following processes. In this function, the shape of the head model is first deformed by the deformation elements $\boldsymbol{a}_t^{(i)}$ of $\boldsymbol{s}_t^{(i)}$ using Eq.(1). The deformed shape is then translated and rotated depending on the pose elements $\boldsymbol{p}_t^{(i)}$ of $\boldsymbol{s}_t^{(i)}$. After the 3D feature points of the transformed shape are projected onto the image plane $h$, the sum of matching scores is calculated between the neighboring region of each projected 2D point and the corresponding template included in the template set $T_h$ by normalized correlation. The sum is given as the output of $\mathcal{N}_h(\boldsymbol{s}_t^{(i)})$.

We apply $\mathcal{N}_h(\boldsymbol{s}_t^{(i)})$ to all image planes $h$ to produce a total score $c_t^{(i)}$ (Eq.(3)). We then calculate the weight $\pi_t^{(i)}$ from the total score $c_t^{(i)}$ using a Gaussian function as in Eq.(4). Finally, each weight $\pi_t^{(i)}$ is normalized so that the sum of the $\pi_t^{(i)}$ is equal to 1.

$$c_t^{(i)} = \sum_{h \in \{L,R\}} \mathcal{N}_h(\boldsymbol{s}_t^{(i)}) \tag{3}$$

$$\pi_t^{(i)} \propto \exp\left(-\frac{\left(2K - c_t^{(i)}\right)^2}{2\sigma^2} - \frac{1}{2}\sum_{b=1}^{B}\left(\frac{a_{t,b}^{(i)}}{\mu_b}\right)^2\right) \tag{4}$$

Here, $\sigma$ is the standard deviation of the Gaussian function and is empirically set to 3.0, $a_{t,b}^{(i)}$ is the $b$-th element of $\boldsymbol{a}_t^{(i)}$, and $\mu_b$ is the $b$-th element of $\boldsymbol{\mu}$. Note that in Eq.(4) we multiply the function with regard to $\boldsymbol{a}_t^{(i)}$ by using standard deviation vector $\boldsymbol{\mu}$ in order to prevent excessive face deformation.

We finally calculate the state vector $\boldsymbol{x}_t$ representing the current pose $\boldsymbol{p}_t$ and deformation $\boldsymbol{a}_t$ by using the sample set $\{(\boldsymbol{s}_t^{(i)}; \pi_t^{(i)})\}$. In this calculation, we aggregate only the neighborhood of the sample with the maximum weight using the following equation:

$$w_t^{(i)} = \begin{cases} 1 \text{ if } \|\boldsymbol{s}_t^{(i)} - \boldsymbol{s}_t^{(M)}\| < d \\ 0 \text{ else} \end{cases} \tag{5}$$

$$\boldsymbol{x}_t = \frac{\sum_{i=1}^{N} \boldsymbol{s}_t^{(i)} \pi_t^{(i)} w_t^{(i)}}{\sum_{i=1}^{N} \pi_t^{(i)} w_t^{(i)}} \tag{6}$$

where $\pi_t^{(M)}$ is the maximum of $\{\pi_t^{(i)}\}$, and $\boldsymbol{s}_t^{(M)}$ is the sample corresponding to $\pi_t^{(M)}$. In the current implementation, the value of $d$ is empirically determined.

We also calculate the velocity $\boldsymbol{v}_t$ of $\boldsymbol{x}_t$ for the estimation of the next frame:

$$\boldsymbol{v}_t = \frac{\boldsymbol{x}_t - \boldsymbol{x}_{t-1}}{\tau} \tag{7}$$

where the last $B$ elements of $\boldsymbol{v}_t$ corresponding to face deformation are set to 0, because the variation of the face deformation parameters does not match well with the assumption of uniform straight motion.

### 2.3 Halfway Partitioned Sampling

We could obtain the new sample set $\{(\boldsymbol{s}_t^{(i)}; \pi_t^{(i)})\}$ using the procedure described above. Actually, instead of the procedure described above, we apply the following sampling and weighting method which is similar to the partitioned sampling technique [13] in principle. We call this sampling technique *the halfway partitioned sampling*.

According to our observations, the motion of the human head and face can be categorized into two typical situations: rigid transformation of head pose with little face deformation, and face deformation with little transformation of head pose. For efficiently handling such situations, we first apply the drift of the pose elements from Eq.(2) to just half of the total samples; to the other half, we apply only the deformation elements' operation from Eq.(2). Then we determine the weights of those samples by Eq.(3) and (4).

After that, we apply a standard resampling technique, that is total resampling in the all dimensions, to the sample set $\{(\boldsymbol{s}_t^{(i)}; \pi_t^{(i)})\}$ to improve the accuracy of the PDF. Even if face deformation and rigid transformation occur simultaneously, our method can handle such cases owing to this resampling process.

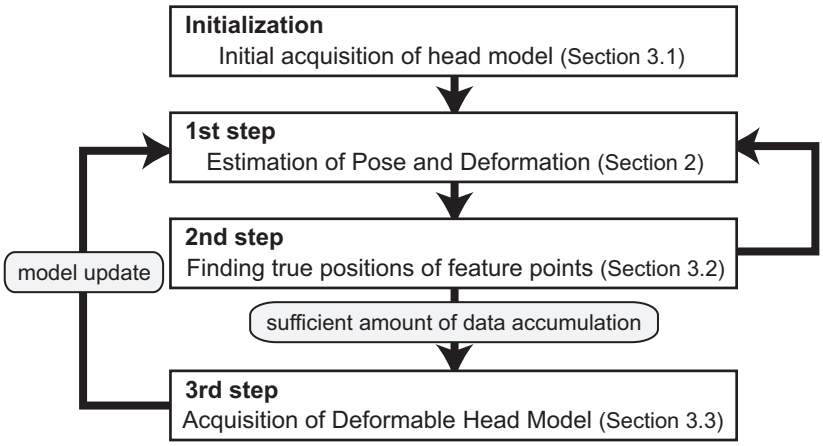


**Fig. 2.** Flow of acquiring deformable head model

## 3 Method for Acquiring Deformable Head Model

In this section, we explain the method for acquiring the deformable head model of a user's head. This method consists of an automatic initialization step and three model acquisition steps as shown in Fig.2.

In the automatic initialization step, we construct the rigid head model as the initial head model. This initialization step is described in Section 3.1. After the initialization, we execute the three steps for acquiring a deformable head model. At the first step, we estimate the 3D head pose $\boldsymbol{p}_t$ and face deformation $\boldsymbol{a}_t$ from input images in real time, as described in Section 2. In the second step, we find the true positions of feature points in each input frame by using $\boldsymbol{p}_t$ and $\boldsymbol{a}_t$, as described in Section 3.2. Finally, in the third step, we calculate the mean shape vector $\bar{\boldsymbol{M}}$ and the basis shape matrix $\boldsymbol{\mathcal{M}}$ in Eq.(1) by using the PCA as described in Section 3.3.

The new deformable head model is then used for the next estimation of 3D head pose and face deformation in the first step. This framework allows progressive improvement of the accuracy for estimating head pose and face deformation in parallel with refining a user's deformable head model.

### 3.1 Initial Acquisition of Head Model

The initialization step automatically constructs a 3D rigid model of a user's head. In this step, we utilize the OKAO vision library developed by OMRON Corporation [12]. This library is used for detecting a face and 6 facial feature points, that is, the inner and outer corners of both eyes, both corners of the mouth, from input images. The other feature points are detected as the distinct features [19] satisfying certain geometrical relations given *a priori*.

We first try to detect those feature points from the left image, and then search for the corresponding points based on epipolar constraints from the right image. After that, the 3D shape $\boldsymbol{M}$ is calculated based on triangulation, and the 3D shape and image template set $T_L$, $T_R$ are registered together.

Note that we cannot estimate the deformation vector $\boldsymbol{a}_t$ when we have only the rigid model just after this initialization step. In such situation, $\boldsymbol{a}_t$ is set to zero vector.

### 3.2 Finding True 3D Positions of Feature Points

The purpose of this step, the second step of acquiring a deformable head model, is to find the true 3D positions of feature points from each input image frame. For constructing an accurate deformable head model, we have to collect the exact positions of each feature point. However, these positions do not necessarily coincide with the positions given in $\boldsymbol{M}_t$ which are calculated from Eq.(1) and the estimated $\boldsymbol{a}_t$.

For this purpose, we make use of the feature tracking technique [11, 19]. Let $\boldsymbol{M}'_t$ be the $3K$-dimensional vector that represents the true 3D coordinates of $K$ feature points in the model coordinate system. For reliably finding $\boldsymbol{M}'_t$, we refer to $\boldsymbol{p}_t$ and $\boldsymbol{a}_t$ which are estimated in the first step (Section 2).

At first, we define a function $\mathcal{P}_h$ that first transforms $\boldsymbol{M}'_t$ by the head pose $\boldsymbol{p}_t$ and then projects the transformed points onto the image plane $h$:

$$\boldsymbol{m}_{h,t} = \mathcal{P}_h(\boldsymbol{p}_t, \boldsymbol{M}'_t) \tag{8}$$

where $\boldsymbol{m}_{h,t}$ is a $2K$-dimensional vector that consists of the 2D coordinates of $K$ projected points. We also define a $K$-dimensional intensity vector $\boldsymbol{I}_t^h(\boldsymbol{m}_{h,t})$ whose $k$-th element is the intensity of the $k$-th 2D position represented by $\boldsymbol{m}_{h,t}$ in the input image frame $t$ from the camera $h$.

By using those definitions, we produce the energy function $E_t^I$ to minimize as follows:

$$E_t^I = \sum_{\substack{\text{ROI} \\ h \in \{L,\, R\}}} \left\{ \begin{array}{c} \rho \|\boldsymbol{I}_t^h(\boldsymbol{m}_{h,t}) - \boldsymbol{I}_{t-1}^h(\boldsymbol{m}_{h,t-1})\|^2 \\ + \|\boldsymbol{I}_t^h(\boldsymbol{m}_{h,t}) - \boldsymbol{I}_1^h(\boldsymbol{m}_{h,1})\|^2 \end{array} \right\} \tag{9}$$

Here, the first term in Eq.(9) is the standard energy function representing the difference between the $K$ Regions Of Interest (ROIs) in the current image $\boldsymbol{I}_t^h$ and their corresponding ROIs in the previous image $\boldsymbol{I}_{t-1}^h$. In contrast, the second term works for the minimization of the difference between the current image $\boldsymbol{I}_t^h$ and the first image $\boldsymbol{I}_1^h$. This term is useful for avoiding the drift of feature points as used also in [8]. In addition, $\rho$ is a constant for determining the ratio between the effect of the first term and that of the second term. In the current implementation, $\rho$ is empirically set to 4, and the size of ROI is $16 \times 16$.

We also introduce the additional term $E_t^M$ based on the estimated shape $\boldsymbol{M}_t$. This term plays a very important role for preventing failure of tracking the

feature points especially when a user's head pose changes significantly.

$$E_t^M = \|\boldsymbol{M}'_t - \boldsymbol{M}_t\|^2 \tag{10}$$

This function means that we find each point of $\boldsymbol{M}'_t$ in the neighboring region of each point of $\boldsymbol{M}_t$. Such method for finding the point reduces significantly the probability of losing tracking of feature points. Furthermore, as the deformable head model is refined more accurately, the minimization of $E_t^M$ becomes more effective for finding the correct 3D coordinates of feature points.

Hence, we minimize the following energy function for the purpose of finding $\boldsymbol{M}'_t$:

$$E_t = E_t^I + \epsilon E_t^M \tag{11}$$

where $\epsilon$ is a constant, and it is empirically set to 2000.

$\boldsymbol{M}'_t$ is then found by minimizing $E_t$ in a similar way to [8]. That is, we calculate the difference $d\boldsymbol{M}'_t = \boldsymbol{M}'_t - \boldsymbol{M}'_{t-1}$ successively in each input frame. This is achieved by setting the derivative of $E_t$ with respect to $d\boldsymbol{M}'_t$ to 0.

While the technique described above yields a good tracking result $\boldsymbol{M}'_t$, the components caused by rigid transformation are occasionally involved in $\boldsymbol{M}'_t$. This might lead to incorrect deformable head models that cannot appropriately distinguish face deformation from rigid transformation.

For this reason, we need to eliminate the components of transformation involved in $\boldsymbol{M}'_t$ in a similar way to the method used in [15]. We first calculate the mean shape $\bar{\boldsymbol{M}}'$ of the series from $\boldsymbol{M}'_1$ to $\boldsymbol{M}'_{t-1}$. Then, we apply 3D translation and rotation to $\boldsymbol{M}'_t$ so that the sum of the square distance between the corresponding points of $\boldsymbol{M}'_t$ and $\bar{\boldsymbol{M}}'$ is minimized. While this operation can eliminate the unwanted components due to rigid transformation, it might have an adverse affect on the correctly calculated $\boldsymbol{M}'_t$. Therefore, we apply this operation only if necessary: when the distance between $\boldsymbol{M}'_t$ and $\boldsymbol{M}_t$ exceeds the constant threshold.

### 3.3 Acquisition of Deformable Head Model by PCA

In the third step of acquiring a deformable head model, we calculate the mean shape vector $\bar{\boldsymbol{M}}$ and the basis shape matrix $\boldsymbol{\mathcal{M}}$ in Eq.(1). Our method applies the PCA to the accumulated correct shape set $\{\boldsymbol{M}'_t\}$; then uses the first $B$ basis vectors to form $\boldsymbol{\mathcal{M}}$ for representing face deformation in a similar way to the method by Gokturk et al. [8] This contributes to preventing unfeasible deformation of the human face as well as reducing the number of dimensions of the state vector $\boldsymbol{x}_t$.

Here, we briefly describe how to acquire $\bar{\boldsymbol{M}}$ and $\boldsymbol{\mathcal{M}}$. To be precise, $\{\boldsymbol{M}'_t\}$ consists of only the shape $\boldsymbol{M}'_t$ when a user is facing toward the cameras judging from the estimated pose $\boldsymbol{p}_t$; this is because we desire to use as reliable data as possible for acquiring the deformable head model. We first calculate the mean shape vector $\bar{\boldsymbol{M}}$ from $\{\boldsymbol{M}'_t\}$. Then, in $\{\boldsymbol{M}'_t\}$, we count the $\boldsymbol{M}'_t$ satisfying the

condition where the distance between $\boldsymbol{M}'_t$ and $\bar{\boldsymbol{M}}$ exceeds the predetermined threshold. If this number exceeds the predetermined number $L$ ($L = 600$ in the current implementation), we apply the PCA to $\{\boldsymbol{M}'_t\}$. This condition is necessary for judging whether $\{\boldsymbol{M}'_t\}$ includes sufficient amount of shape deformation. By the PCA-based operation, we obtain the basis shape matrix $\boldsymbol{\mathcal{M}}$ and the $B$-dimensional standard deviation vector $\boldsymbol{\mu}$, each of whose elements represents the standard deviation of its corresponding column of $\boldsymbol{\mathcal{M}}$. $\boldsymbol{\mu}$ is equivalent to the standard deviation of distribution of $\boldsymbol{a}_t$ in Eq.(1). Thus, $\boldsymbol{\mu}$ is used for determining the variance of random noise in Eq.(2) and the weight of each sample in Eq.(4).

## 4 Experimental Evaluation

We have conducted experiments to evaluate the performance of our proposed method. Our system consists of a Windows-based PC with Intel Pentium4 3.0-GHz and two CCD black-and-white digital video cameras connected via IEEE-1394. Each image frame was captured at the resolution of $640 \times 480$. The size of image templates for normalized correlation was set to $16 \times 16$, and a set of 1000 samples was used for particle filtering. Our method runs at 30 frames per second with this configuration, including the 1st step and the 2nd step of Fig.2. In addition, the 3rd step of Fig.2, that is, the PCA-based calculation of the shape vectors, can also execute at very short execution time without spoiling real-time performance (30fps) of the proposed system.

We prepared an image sequences of a user moving his head pose with occasional face deformation. This image sequence was 60 seconds long and therefore contained 1800 frames. By using the first 1200 frames, the user's deformable head model was acquired with our proposed method. Then, we estimated the 3D head pose and face deformation from the last 600 image frames using the acquired deformable head model. For the first approximately 150 out of 600 image frames, the user's head moved by rigid transformation accompanied by little face deformation. After that, the user moved his head accompanied by face deformation, for example, opening and closing his mouth.

For comparison, we also conducted head pose estimation from the same 600 image frames using the rigid head model. This rigid head model was acquired from the initialization step in Section 3.1. We compared those two estimation results.

Fig.3 shows the estimation results using the rigid head model and the deformable head model. In this figure, the thin lines show the results with the rigid head model, and the thick lines represent the ones using the deformable head model. For the first 150 frames, both estimation results are similar each other. This means the deformable head model can estimate rigid transformation without generating unwanted face deformation. On the other hand, we can see the clear difference between both results for the remaining 450 frames. As shown in this figure, the results using the deformable model are far more stable than the ones using the rigid model. Hence, the deformable head model constructed

by our proposed method has the capability to handle the face deformation in contrast to the rigid head model.

Fig.4 shows the resulting images. In this figure, we have drawn the model coordinate axes corresponding to the estimated 3D head pose, and the 2D points onto which the estimated shape $\boldsymbol{M}_t$ is projected. The left column of the figure is the results using the rigid head model, and the right column shows the results using the deformable head model. Also from those results, we can confirm that our deformable head model handles face deformation successfully.

We can see the video of this experiment on the Web. [2] This video demonstrates the stability of pose estimation with our deformable head model.

## 5 Conclusions

In this paper, we proposed a new method for acquiring and refining a user's deformable head model in parallel with estimating the user's 3D head pose in real time. The main contributions of our study are summarized in the following three points: 1) real-time estimation of 3D head pose without a preliminary training stage, 2) real-time refinement of a deformable head model, and 3) progressive improvement of the accuracy of estimating head pose and face deformation. The improvement has been successfully demonstrated via experiments. We believe that this work is the first example to achieve simultaneous execution of face deformation modeling and 3D head pose estimation in real-time.

For further study, we are planning to use the Candid Covariance-free Incremental PCA (CCIPCA) [21] that allows basis vectors to be updated at each input image frame.

## Acknowledgment

A part of this work was supported by Grants-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology in Japan (No. 13224051). We also thank Omron Corporation for providing the OKAO vision library used in our method.

## References

1. Azarbayejani, A., Starner, T., Horowitz, B., Pentland, A.: Visually controlled graphics. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 15, No. 6. (1993) 602–605
2. Black, M.,Yacoob, Y.: Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. Proc. IEEE ICCV '95. (1995) 374–381
3. DeCarlo, D., Metaxas, D.: Optical flow constraints on deformable models with applications to face tracking. Int. J. Computer Vision, Vol. 38, No. 2. (2000) 99–127

[2] http://www.hci.iis.u-tokyo.ac.jp/~oka/AMFG2005.html

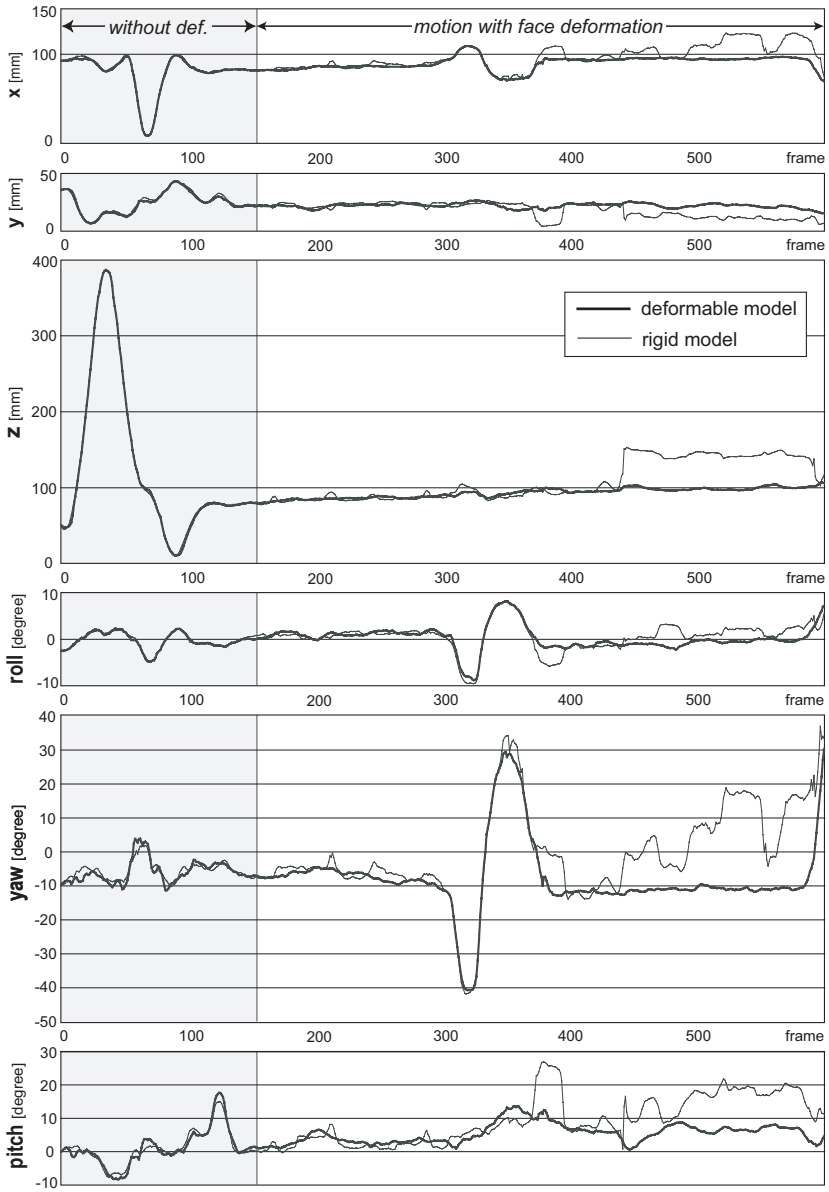


**Fig. 3.** Estimation results with rigid and deformable head model. Translation consists of $x$, $y$, and $z$: $x$ represents the horizontal motion, $y$ shows the vertical motion, $z$ corresponds to the depth-directional motion. Rotation consists of $roll$, $yaw$, and $pitch$: $roll$ is the rotation around the axis toward the front, $yaw$ corresponds to the pan-directional rotation, and $pitch$ represents the tilt-directional rotation.

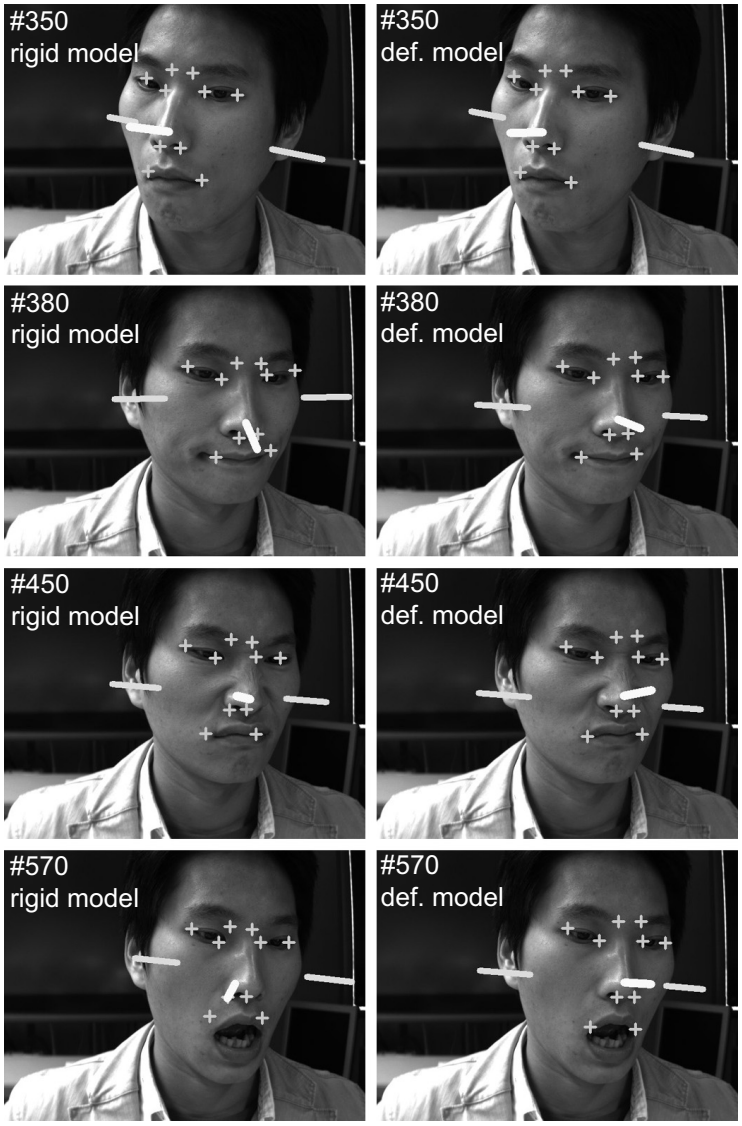


**Fig. 4.** Resulting images: the images of the left column are the estimation results using the rigid head model, and the images of the right column are the estimation results using the deformable head model

4. Del Bue, A., Smeraldi, F., Agapito, L.: Non-rigid structure from motion using non-parametric tracking and non-linear optimization. Proc. IEEE CVPRW 2004, Vol. 1: Articulated and Non-Rigid Motion. (2004)
5. Dornaika, F., Davoine, F.: Head and facial animation tracking using appearance-adaptive models and particle filters. Proc. IEEE CVPRW 2004, Vol. 10: Real-Time Vision for Human-Computer Interaction. (2004)
6. Fua, P.: Using model-driven bundle-adjustment to model heads from raw video sequences. Proc. IEEE ICCV '99, Vol. 1. (1999) 46–53
7. Gee, A., Cipolla, R.: Fast visual tracking by temporal consensus. Image and Vision Computing, Vol. 14. (1996) 105–114
8. Gokturk, S., Bouguet, J., Grzeszczuk, R.: A data-driven model for monocular face tracking. Proc. IEEE ICCV 2001, Vol. 2. (2001) 701–708
9. Isard, M., Blake, A.: Condensation– conditional density propagation for visual tracking. Int. J. Computer Vision, Vol. 29, No. 1. (1998) 5–28
10. Jebara, T., Pentland, A.: Parametrized structure from motion for 3D adaptive feedback tracking of faces. Proc. IEEE CVPR '97. (1997) 144–150
11. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. Proc. Int. Joint Conf. Artificial Intelligence. (1981) 674–679
12. Lao, S., Kozuru, T., Okamoto, T., Yamashita, T., Tabata, N., Kawade, M.: A fast 360-degree rotation invariant face detection system. Demo session of IEEE ICCV 2003. (2003)
13. MacCormick, J., Isard, M.: Partitioned sampling, articulated objects, and interface-quality hand tracking. Proc. ECCV 2000, Vol. 2. (2000) 3–19
14. Matsumoto, Y., Zelinsky, A.: An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. Proc. IEEE FG 2000. (2000) 499–504
15. Matthews, I., Baker, S.: Active appearance models revisited. Int. J. Computer Vision, Vol. 60, No. 2. (2004) 135–164
16. Morency, L., Rahimi, A., Darrell, T.: Adaptive view-based appearance models. Proc. IEEE CVPR 2003, Vol. 1. (2003) 803–810
17. Oka, K., Sato, Y., Nakanishi, Y., Koike, H.: Head pose estimation system based on particle filtering with adaptive diffusion control. Proc. IAPR MVA 2005. (2005) 586–589
18. Shan, Y., Liu, Z., Zhang, Z.: Model-based bundle adjustment with application to face modeling. Proc. IEEE ICCV 2001, Vol. 2. (2001) 644–651
19. Shi, J., Tomasi, C.: Good features to track. Proc. IEEE CVPR '94. (1994) 593–600
20. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3D tracking using online and offline information. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 26, No. 10. (2004) 1380–1384
21. Weng, J., Zhang, Y., Hwang, W.: Candid Covariance-Free Incremental Principal Component Analysis. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 25, No. 8. (2003) 1034–1040