MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan

14-5

Head Pose Estimation System Based on Particle Filtering with Adaptive Diffusion Control

Kenji Oka
The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, Japan
oka@iis.u-tokyo.ac.jp

Yoichi Sato
The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, Japan
ysato@iis.u-tokyo.ac.jp

Yasuto Nakanishi
Tokyo University of Agricalture and Technology
2-24-16 Naka-cho, Koganei-city, Tokyo, Japan
yasuto@cc.tuat.ac.jp

Hideki Koike
University of Electro-Communications
1-5-1 Chofugaoka, Chofu-city, Tokyo, Japan
koike@is.uec.ac.jp

Abstract

In this paper, we propose a new tracking system based on a stochastic filtering framework for reliably estimating the 3D pose of a user's head in real-time. Our system estimates the pose of a user's head in each image frame whose 3D model is automatically obtained at an initialization step. In particular, our estimation method is designed to control the diffusion factor of a motion model adaptively. This technique contributes significantly to improving the following performance simultaneously: the robust tracking against abrupt head motion and the accurate pose estimation when the user is staring at a point in a scene. The performance of our proposed method has been successfully demonstrated via experiments.

1 Introduction

Users' attention plays an important role in designing human-computer interfaces (HCIs) used effectively and intuitively in real environments. Since users' attentions correlate well with their gaze points, real-time sensing of gaze direction or face orientation is considered as one of the key components for HCIs. This motivated us to develop a new vision-based method for estimating the 3D pose, i.e., position and orientation, of a user's head in real-time.

A number of vision-based techniques for tracking objects have been proposed by other researchers in the past. Among such techniques, the ones based on particle filtering[3] can handle challenging situations that contain clutter, occlusion, and noise. This advantage is very important for HCI applications because those applications are often expected to be able to track objects in the various environments including such situations. For this reason, some of the previously proposed methods have utilized particle filtering for estimating a user's 3D head pose[7, 1, 2, 5].

In addition to robustness against clutter and occlusions, it is very important to realize the following two aspects simultaneously for applying vision-based tracking methods for HCI applications: dealing with abrupt fast motions of a user's head and estimating the 3D pose of a user's head accurately when the user is staring at a point in a scene. Furthermore, it is also required to run both initialization and tracking fully-automatically. Unfortunately, the previously proposed methods based on particle filtering fail to achieve these important elements simultaneously.

The aim of this work is to develop a new vision-based method which can estimate the 3D head pose with high accuracy in real-time and, at the same time, is robust against sudden abrupt motions of a user's head. The key component of our proposed method is adaptive control of diffusion factors in a motion model of a user's head used in particle filtering.

In addition, our system realizes automatic initialization for estimating head pose. This contributes a great deal to the development of the HCI application that an arbitrary user can utilize.

In this paper, we will describe the details of our proposed system and demonstrate its performance improvement by adaptive diffusion control via experiments using real images.

2 Head Pose Estimation System

In this section, we describe our proposed system for real-time 3D tracking of a user's head pose from image inputs from two cameras[1].

Our system mainly consists of two steps: the automatic initialization step for creating a 3D model of a user's head with multiple feature points, and the tracking step by using consecutive image frames and the created 3D model based on particle filtering[3]. The main flow of our system is shown in Figure 1.

2.1 Initialization step

Our proposed system uses the 3D model of a user's head with K facial feature points. Here, K is set to 10 in this work, i.e., inner and outer corners of both eyes, both corners of the mouth, both nostrils, and the inner corners of both brows. Each feature point has a 3D position in the model coordinate system fixed to a user's head and two corresponding image templates for the left and right cameras.

In the initialization step, we utilize the OKAO vision

[1] Although we describe the system with two cameras in this paper, we can increase or decrease the number of cameras given the similar framework.

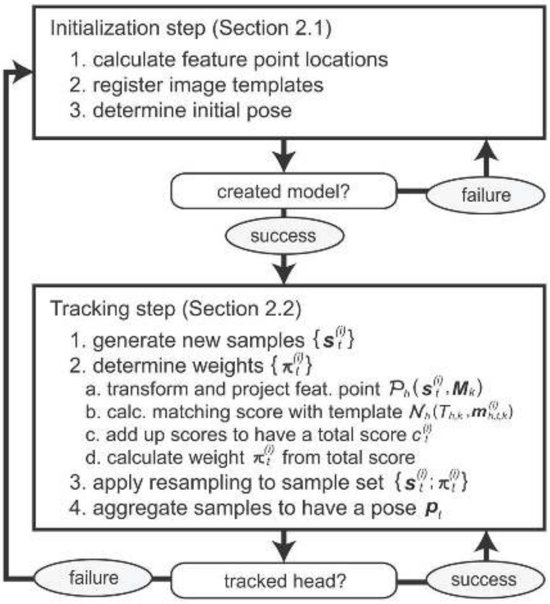

Initialization step (Section 2.1)
1. calculate feature point locations
2. register image templates
3. determine initial pose
created model?
failure
success
Tracking step (Section 2.2)
1. generate new samples $\{s_t^{(i)}\}$
2. determine weights $\{\pi_t^{(i)}\}$
a. transform and project feat. point $\mathcal{P}_h(s_t^{(i)}, M_k)$
b. calc. matching score with template $\mathcal{N}_h(T_{h,k}, m_{h,t,k}^{(i)})$
c. add up scores to have a total score $c_t^{(i)}$
d. calculate weight $\pi_t^{(i)}$ from total score
3. apply resampling to sample set $\{s_t^{(i)}; \pi_t^{(i)}\}$
4. aggregate samples to have a pose p_t
failure
tracked head?
success

Figure 1 Flow of head pose estimation system

library developed by OMRON Corporation[4]. This library is used for detecting a face and 6 facial feature points of the face, i.e., inner and outer corners of both eyes and both corners of the mouth, from input images. The other facial feature points are detected as the distinct features[8] satisfying certain geometrical relations given *a priori*.

We first try to detect those feature points from the left image; we then search for the corresponding points based on epipolar constraints from the right image. After that, the 3D position of each feature point M_k ($k = 1 \ldots K$) is calculated based on triangulation, and the 3D position and image templates $T_{L,k}$, $T_{R,k}$ of each feature point M_k are registered together. In addition, we determine an initial pose for tracking step.

This initialization process is repeated automatically every time a user first appears in input images or the system fails to keep tracking a user's head. Therefore, an arbitrary user can utilize our system reliably without any burdensome constraint and training data for creating a model.

2.2 Tracking step

In the tracking step, we estimate a user's head pose p_t in the t-th image frame. The head pose p_t is represented as a 6D vector $(x_t, y_t, z_t, \phi_t, \theta_t, \varphi_t)^{\mathrm{T}}$ in a 6D state space S where $(x_t, y_t, z_t)^{\mathrm{T}}$ and $(\phi_t, \theta_t, \varphi_t)^{\mathrm{T}}$ are respectively the translation and the rotation from the world coordinate system to the model coordinate system fixed to the user's head. For the pose estimation, we make use of the 3D head model created in the initialization step, and particle filtering.

Particle filtering represents the probability density function (PDF) of a state as a set of many discrete samples; each sample has the corresponding weight. Hence, this sample set can approximate an arbitrary PDF including non-Gaussian ones. Our method uses the sample set $\{(s_t^{(i)}; \pi_t^{(i)})\}$ ($i = 1 \ldots N$) which consists of N discrete samples $s_t^{(i)}$ in the 6D state space S and its corresponding weight $\pi_t^{(i)}$.

Our proposed method determines the head pose p_t in the t-th frame as shown in "Tracking step" of Figure 1. We will describe the details of this step below.

First, we generate N new samples $s_t^{(i)}$ based on the previous sample set $\{(s_{t-1}^{(i)}; \pi_{t-1}^{(i)})\}$ and the motion model by repeating the following process N times.

We choose a base sample s'_{t-1} from $\{(s_{t-1}^{(i)}; \pi_{t-1}^{(i)})\}$ based on the probability proportional to the weight $\pi_{t-1}^{(i)}$. Then, the chosen sample s'_{t-1} is drifted into $s_t^{(i)}$ by assuming uniform straight motion of a user's head between two successive image frames as:

$$s_t^{(i)} = s'_{t-1} + \tau v_{t-1} + \omega \tag{1}$$

where τ is the time interval between frames, v_{t-1} is the previous velocity of the pose, and ω is a system noise added to s'_{t-1}. Note that v_{t-1} is calculated at the end of this tracking step and ω is a 6D Gaussian noise which is adaptively controlled by the method described in Section 2.3.

After we obtain new samples $\{s_t^{(i)}\}$, we compute the weight $\pi_t^{(i)}$ for $s_t^{(i)}$ by evaluating it based on the current input image. Here, we determine $\pi_t^{(i)}$ through the four steps as shown in Figure 1.

Given a sample $s_t^{(i)}$, we can translate and rotate the head model depending on the pose represented by $s_t^{(i)}$. We project the feature points M_k of the transformed head model to the points $m_{h,t,k}^{(i)}$ ($h \in \{L, R\}$) on the image plane by the projection function $\mathcal{P}_h(s_t^{(i)}, M_k)$. Then, we calculate a matching score between the neighboring region of $m_{h,t,k}^{(i)}$ and its corresponding template $T_{h,k}$ by normalized correlation-based function $\mathcal{N}_h(T_{h,k}, m_{h,t,k}^{(i)})$.

For each sample $s_t^{(i)}$, we apply the normalized correlation $\mathcal{N}_h$ to $2K$ projected image points $m_{h,t,k}^{(i)}$ and add up those results to have a total score $c_t^{(i)}$. We finally calculate the weight $\pi_t^{(i)}$ from the total score $c_t^{(i)}$ by using Gaussian function.

$$c_t^{(i)} = \sum_{k=1}^{K} \sum_{h \in \{L,R\}} \mathcal{N}_h(T_{h,k}, \mathcal{P}_h(s_t^{(i)}, M_k)) \tag{2}$$

$$\pi_t^{(i)} \propto \exp\left(-\frac{\left(2K - c_t^{(i)}\right)^2}{2\sigma^2}\right) \tag{3}$$

where σ is the standard deviation of Gaussian function and it is empirically set to 3.0. Each weight $\pi_t^{(i)}$ is normalized so that the sum of $\pi_t^{(i)}$ is equal to 1.

In this way, we can obtain the new sample set $\{(s_t^{(i)}; \pi_t^{(i)})\}$ in the t-th image frame. However, this sample set is somewhat rough approximation of the PDF due to the limitation of the number of samples. Hence, we apply a resampling technique similar to that of [6] to $\{(s_t^{(i)}; \pi_t^{(i)})\}$ to make the accuracy improved.

Finally, we calculate the estimated pose p_t from $\{(s_t^{(i)}; \pi_t^{(i)})\}$. In this calculation, we aggregate only the neighborhood of the sample with the maximum weight by the following equation:

$$w_t^{(i)} = \begin{cases} 1 & \text{if } \left\| s_t^{(i)} - s_t^{(M)} \right\| \le d \\ 0 & \text{else} \end{cases} \tag{4}$$

$$\boldsymbol{p}_t = \frac{\sum_{i=1}^{N} \boldsymbol{s}_t^{(i)} \pi_t^{(i)} w_t^{(i)}}{\sum_{i=1}^{N} \pi_t^{(i)} w_t^{(i)}} \tag{5}$$

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

Moreover, we calculate the estimated pose's velocity $\boldsymbol{v}_t$ for the estimation of the next frame:

$$\boldsymbol{v}_t = \frac{\boldsymbol{p}_t - \boldsymbol{p}_{t-1}}{\tau} \tag{6}$$

2.3 Adaptive diffusion control

In this section, we describe the main contribution to improve the performance for estimating the pose of a user's head.

Generally, tracking methods based on stochastic filtering have difficulties when the target's motion differs significantly from the given motion model; it often generates severe overshoot and loses track in the worst case.

One remedy for this problem is to increase the diffusion factor in the motion model so that the pose of the target is well contained in the range of predicted poses obtained from the motion model. However, in the case of particle filtering, the increase of the diffusion factor results in a sparser set of predicted samples around the true pose and, consequently, the deterioration of estimation accuracy.

To overcome this problem, our method controls the diffusion factor of a motion model adaptively. In other words, our method is designed to increase the diffusion factor only when necessary. This is done by increasing or decreasing the diffusion factor represented by the 6D noise vector $\boldsymbol{\omega}$ (Equation (1)) depending on the velocity of the user's head motion.

Such control of the noise vector $\boldsymbol{\omega}$ contributes to improving the robustness against sudden abrupt motion and maintaining the accuracy of estimation at the same time. $\boldsymbol{\omega}$ is the Gaussian noise with a zero mean; its covariance matrix is the 6D diagonal matrix which has β_x^2, β_y^2, β_z^2, β_ϕ^2, β_θ^2, β_φ^2 as the diagonal elements. Here, we define $\boldsymbol{\delta}_t = (\beta\beta\beta\beta\beta\beta_{x\,y\,z},\ {}_{\phi\theta\varphi}\ ,\ )^{\mathrm{T}}$ as "diffusion control vector", and control the diffusion factor of sample through controlling the vector $\boldsymbol{\delta}_t$.

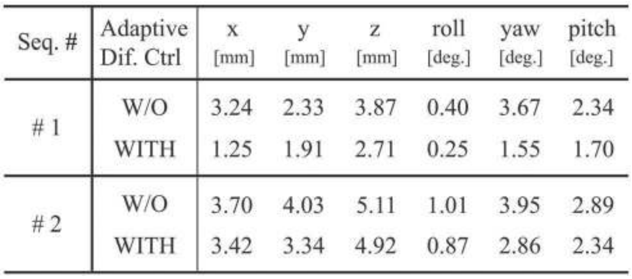
In this work, we assume that the uncertainty of the predicted pose is proportional to the magnitude of the pose change. Therefore, $\boldsymbol{\delta}_t$ increases linearly with an increase of the absolute value of corresponding velocity:

$$\boldsymbol{\delta}_t \boldsymbol{\Gamma} = + \hat{\boldsymbol{v}}_{t-1} \quad \boldsymbol{\gamma} \tag{7}$$

where $\hat{\boldsymbol{v}}_{t-1}$ is a 6D vector whose elements are the absolute values of the corresponding elements of $\boldsymbol{v}_{t-1}$. $\boldsymbol{\Gamma}$ and $\boldsymbol{\gamma}$ are a 6×6 matrix and a 6D vector respectively, which are updated iteratively based on the accumulated tracking results.

Similar control of diffusion factor is also made advantage of in the method of Dornaika et al.[2] which was developed independently around the same time as our work. In their method, 3D deformable mesh model of human face is utilized instead of our simple model acquired automatically, and diffusion factor is controlled based on the error of registering the model to each input image frame. Although their tracking method works well, it was not evaluated how much improvement was achieved by their diffusion control. In contrast, the performance improvement by our adaptive diffusion control has been evaluated via experiments in Section 3.

Table 1 RMS error of the method WITHOUT and WITH Adaptive Diffusion Control



Seq. #	Adaptive Dif. Ctrl	x [mm]	y [mm]	z [mm]	roll [deg.]	yaw [deg.]	pitch [deg.]
# 1	W/O	3.24	2.33	3.87	0.40	3.67	2.34
	WITH	1.25	1.91	2.71	0.25	1.55	1.70
# 2	W/O	3.70	4.03	5.11	1.01	3.95	2.89
	WITH	3.42	3.34	4.92	0.87	2.86	2.34

3 Experimental Results

We have conducted experiments to evaluate the performance of our estimation 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 by IEEE-1394. Each image was captured at a resolution of 640×480. The size of image templates for normalized correlation was set to 16×16, and a set of 1000 samples was used for particle filtering. Our proposed method runs at 30 frames per second with this configuration.

In order to evaluate the tracking performance of our proposed method under different conditions, we used two pre-recorded sequences of a user moving his head. In one image sequence (Sequence #1), a user moved his head relatively slowly and held still occasionally. In the other sequence (Sequence #2), a user moved his head relatively fast. Each sequence was 20 seconds long and therefore contained 600 frames, which did not include severe change of illumination or occlusion with the user's hands. The pose of the user's head was also measured directly by using an electromagnetic sensor called the Polhemus FASTRAK to provide ground truth for evaluating the accuracy of pose estimation by our method.

In this experiment, we stabilize $\boldsymbol{\Gamma}$ and $\boldsymbol{\gamma}$ in Equation (7) to 6D identity matrix and $(1,1,1,1,1,1)^{\mathrm{T}}$ respectively in order to eliminate any bias due to the difference of the accumulated tracking results. On the other hand, in the experiment without adaptive diffusion control, we set the diffusion control vector $\boldsymbol{\delta}_t$ in Section 2.3 to $(5,5,5,5,5,5)^{\mathrm{T}}$. This vector gave the best performance among those used in our experiments without adaptive diffusion control.

Table 1 shows the root mean square (RMS) error of pose estimation of Sequence #1 and Sequence #2. As we can see clearly from these results, adaptive diffusion control in our method is effective for increasing accuracy and robustness of pose estimation.

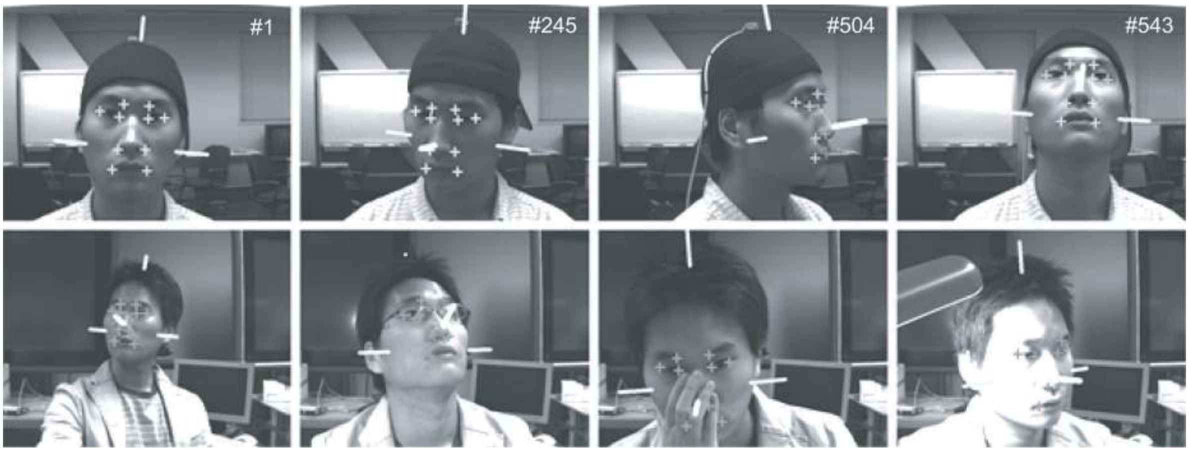


Figure 2 Resulting images (The top is the resulting images of Sequence #2, and the bottom shows the tracking results in some challenging situations.)

Figure 2 shows the resulting images of pose estimation by our method. The top of the figure is the ones for several input image frames of Sequence #2. On the other hand, the bottom shows the tracking results for challenging situations such as when the user is wearing eyeglasses, his face is partially occluded, and dynamically changing illumination and background image. Even in these challenging situations, our method was able to estimate the pose of the user's head reliably. You can see the movies of our system on our Web page (http://www.hci.iis.u-tokyo.ac.jp/~oka/MVA2005.html).

4 Conclusion

In this work, we have proposed a new system for estimating the 3D pose of a user's head reliably from input images from multiple cameras in real-time.

Our system consists of two steps: initialization step and tracking step. The initialization step can create the 3D model of a user's head with multiple feature points automatically. Hence, an arbitrary user can utilize our system reliably without any burdensome constraint and training data for creating a model.

In the tracking step, our system tracks the head pose in consecutive image frames based on particle filtering. The key component of this tracking step is adaptive control of diffusion factors in a motion model of a user's head motion used in particle filtering. This contributes significantly to improving the following performance simultaneously: the robust tracking against abrupt head motion and the accurate pose estimation when the user is staring at a point in a scene. The performance of our method has been successfully demonstrated via experiments.

Further studies include extension of our method in several directions. In particular, we are planning to incorporate 3D deformable model of a user's head in our current method in order to increase the accuracy of the pose estimation even further.

Acknowledgements

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 the initialization step of our system.

References

[1] B. Braathen, M. Bartlett, G. Littlewort, E. Smith, and J. Movellan: "An approach to automatic recognition of spontaneous facial actions," *Proc. FG2002*, pp.360-365, 2002.

[2] F. Dornaika and F. Davoine: "Head and facial animation tracking using appearance-adaptive models and particle filters," *Proc. Workshop on Real-Time Vision for Human-Computer Interaction*, 2004.

[3] M. Isard and A. Blake: "Condensation- conditional density propagation for visual tracking", *Int. J. Computer Vision*, vol.29, no.1, pp.5-28, 1998.

[4] S. Lao, T. Kozuru, T. Okamoto, T. Yamashita, N. Tabata, and M. Kawade: "A fast 360-degree rotation invariant face detection system," *Demo session of ICCV2003*, 2003.

[5] L. Lu, X. Dai, and G. Hager: "A particle filter without dynamics for robust 3D face tracking," *Proc. Workshop on Face Processing in Video*, 2004.

[6] J. MacCormick and M. Isard: "Partitioned sampling, articulated objects, and interface-quality hand tracking," *Proc. ECCV2000*, pp.II-3-19, 2000.

[7] J. Sherrah and S. Gong: "Fusion of perceptual cues for robust tracking of head pose and position," *Pattern Recognition*, vol.34, no.8, 2001.

[8] J. Shi and C. Tomasi: "Good features to track," *Proc. CVPR'94*, pp.593-600, 1994.