

# Vision-Based Face Tracking System for Large Displays

Yasuto Nakanishi\* Takashi Fujii\* Kotaro Kiatjima\*

Yoichi Sato\*\* Hideki Koike\*

\* Graduate School of Information Systems, Univ. of Electro-Communications1-5-1  
Chofugaoka, Chofu-City, Tokyo 182-8585, Japan  
{naka, fujii, kita, koike}@vogue.is.uec.ac.jp  
\*\* Institute of Industrial Science, Univ. of Tokyo  
4-6-1 Komaba, Meguro-Ku, Tokyo 153-8505, Japan  
ysato@iis.u-tokyo.ac.jp

**Abstract.** In this paper, we present a stereo-based face tracking system which can track the 3D position and orientation of a user in real-time, and the system's application for interaction with a large display. Our tracking system incorporates dynamic update of template images for tracking facial features so that the system can successfully track a user's face for a large angle of rotation. Another advantage of our tracking system is that it does not require a user to manually initialize the tracking process, which is critical for natural and intuitive interaction. Based on our face tracking system, we have implemented several prototype applications which change information shown on a large display adaptively according to the location looked at by a user.

## 1 Introduction

Recently, large displays such as plasma displays or LCD projectors that can project images to a large area have become popular. They are often used in public places (e.g., train stations or shopping malls) for showing information. However, most of this type of information generally consists of pictures or movies, and it is only repeated and is not interactive, especially in public areas. Although the display equipped with a touch sensor will realize the human computer interaction, it needs the positive action of a user to do so. In ubiquitous computing environments that might contain many large displays, the perceptual user interface that shows information according to a natural activity of a user or to the situation of the place might be desirable. Using the eyes or the face as a source of input in advanced user interfaces has long been a topic of interest to the human computer interaction field. Tracking faces of users who look at various parts of the screen would be a fundamental tool for a variety of perceptual user interface applications in ubiquitous computing environments.

To realize interaction styles that are non-contact, passive, robust, accurate and real-time, there are several commercial products and much research based on computer vision techniques [2,3,5,10]. However, most of the previously developed face-tracking systems were designed to be used by a user sitting in front of a monitor; therefore, they are not suitable for applications with a large display such as a large projection on a wall.

Haro presented a real-time pupil detector and tracker that utilized a probabilistic framework [2]. They used an infrared lighting camera to capture the physiological properties of eyes, Karman trackers to model eye/head dynamics, a probabilistic-based appearance model to represent eye appearance. Kawato proposed an approach that tracks a point between the eyes and then locates the eyes [3]. It utilizes an image filter i.e., the circle-frequency filter to detect “between-eyes”, and stores the small area around it as a template for template matching. Stiefelhagen presented an eye tracker without special lights that employed neural networks to estimate a user’s eye gaze using the images of both of the user’s eyes as input [10]. They trained several neural networks to estimate a user’s eye gaze on a computer screen using the eye images obtained with their eye tracker. However, most of these systems utilize a monocular image and it is very difficult to compute the full 3D locations and orientation of a face or to detect the eye gaze direction accurately and robustly. The most relevant work to us is by [5]; that work employs the template matching method for detecting the edges of eyes and a mouth by using a stereo camera pair. Their system tracks the 3D coordinates of the facial features and it aims to utilize them as a visual human interface for a cooperative task with a robot. These studies, however, assume that a user sits down in front of a computer monitor. Our purpose in this research is to develop a face-tracking system not for a personal display, but rather for a large display.

There has been some research in developing public intelligent environments that utilize image-processing techniques. Chistian developed the Digital Smart Kiosk that detects and tracks prospective clients and conveyed this awareness via an animated talking face [1]. PosterCam is a presentation system with a distributed vision system which monitors the presence of people in a distributed physical space [7]. It detects faces of people who visit it, and transmits the detected faces to back-end servers for clustering. Sawhney also developed an exploratory responsive display projected within a shared workspace [9]. It monitors users’ movements and tracks whether they are looking at the display. It changes the level of detail in showing articles according to people’s movements, and captures people’s faces as they browse articles they are interested in. However, it locates and identifies only people’s faces; it does not detect their facial directions.

In this paper, we describe our face tracking system, which has been developed for the use with applications on a large display. First, we briefly present the overview of our face tracking system, and then we introduce several prototype applications based on our face tracking system.

## **2 Our Face Tracking System**

### **2.1 Overview**

We have developed a vision-based face tracking system which can track the position and orientation of a user in real-time (30 frames/sec) [4]. The configuration of our

tracking system is similar to the one proposed by Matsumoto et al. [5], but our tracking system is capable of tracking a user's face for wider angles of rotation by introducing dynamic update of template images as explained in Section 2.3. Our system runs on a PC (Pentium3-866MHz, Linux OS) equipped with a HITACHI IP5010 image-processing board, which is used while being connected to two NTSC cameras. It is equipped with 40 frame memories of 512 x 512 pixels. In order to reduce the processing time of face tracking, we use the lower resolution image whose size is 256 x 220 pixels. We use a camera unit that consists of two 3CCD black-and-white cameras and two near-infrared lights; the disparity of the two cameras is 16 cm (Figure 1). The cameras are equipped with infrared filters. These filters transmit only the light whose wavelength is close to infrared rays. By using this filter, the camera takes only the infrared light that reflects in the face of the user, thereby enabling us to eliminate the background images.



Figure 1. The stereo camera unit our face-tracking system.

## 2.2 Stereo Tracking Algorithm

In order to search facial features from the camera images, we first select the region of the face. This is done by binarizing an input image from each camera while changing the threshold of binarization iteratively. Then, within this extracted facial region, we identify the location of pupils with the algorithm proposed by Stiefelhagen [10]. We search for the pupils by looking for two dark regions that satisfy the creation of anthropometric constraints and lie within a certain area of the face. After the pupils are located in the camera image, we identify the location of the mouth based on histogram projection in two orthogonal directions.

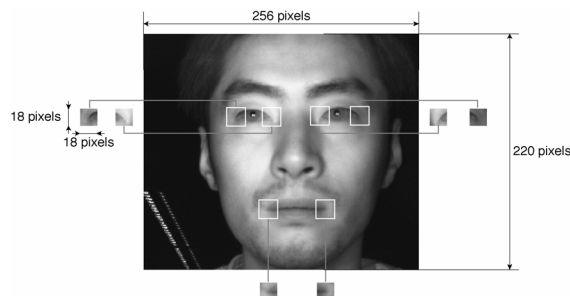


Figure 2. Samples of obtained initial templates.

After storing the template images, we perform the template matching with four template images of eye edges and with two template images of mouth edges for each camera image. This search process using template matching is computationally expensive. Therefore, search areas are defined in our method and the eye edges and the mouth edges are searched for only within these areas instead of over an entire region of the user's face. In this process, each feature is assumed to have a small motion between the current frame and the previous one. We perform the template matching only in the areas around the eye and mouth locations that were found in the previous frame. The areas of a fixed size, e.g.,  $48 \times 38$  pixels in our current implementation, are set so that they include the locations of the edges of the eyes and the mouth obtained at the previous frame. We utilize a function of normalized correlation equipped in the image-processing board in template matching, and six 2D locations are found for each camera image. Then the 3D coordinate of each feature is determined based on triangulation (Figure 3).

The locations of the eye and mouth edges found in template matching are obtained independently, and the provided 3D coordinates do not always correspond to the model of the face registered at the initialization. There might be the case that multiple candidates exist for matching and that inappropriate points are detected, and it would not be appropriate that we utilize those locations. We utilize the 3D model of the face stored at the initialization to cope with this problem. We revise the coordinates provided in template matching so that they retain the nature of the rigid body model. We use the algorithm that lets a rigid body model fit the last state in the previous frame using the virtual springs proposed in [5].

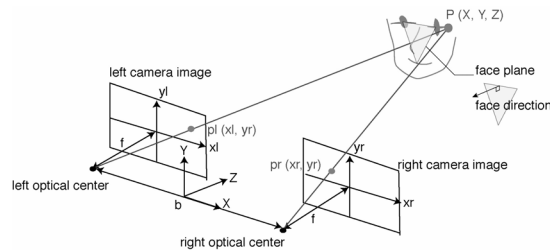


Figure 3. Coordinate system for face tracking

### 2.3 Dynamic Update of Template Images of Facial Features

In order to extend the limit of rotation angles of a user's face which can be tracked with our face tracking system, we incorporate dynamic update of template images of facial features. In our face algorithm for large displays, we dynamically store new template images according to the situation, and we utilize those images and the template images registered at the initialization together. When a user turns his/her face to the right, we use the template images registered at the initialization in the template matching for the right camera image, and use the new template images

obtained from the right camera image in the template matching for the left camera image. When obtaining new template images from the right camera, we store images that are at 2D locations corresponding to the 3D model of the previous frame from the current right camera image (Figure 4). We utilize the 3D face model in the previous frame for switching the mode storing new template images; the modes are “right”, “center” and “left”. Table 1 shows what kinds of template images are used at each mode.

When the system judges that a user turns his/her face to the right and uses new template images although he/she turns his/her face to the left, the reliability of the template matching will decrease and that will influence the tracking in the next frame. Such a case happened when a user turned his/her face suddenly to the left from the right. We coped with this problem by checking transition of the modes. When users turn their faces, we suppose that the transition of the modes should contain the mode of the center like as "from right to center" or “from center to left". When the transition is "from right to left", we regard it as an error. When users turn their faces suddenly, such an error might happen. If we catch that error, we obtain the direction of the face using the initial template images.

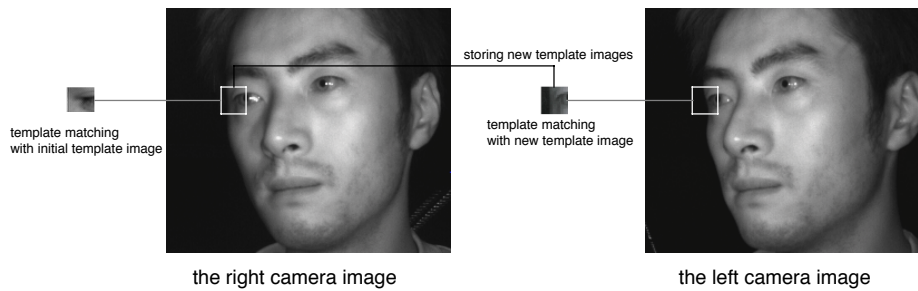


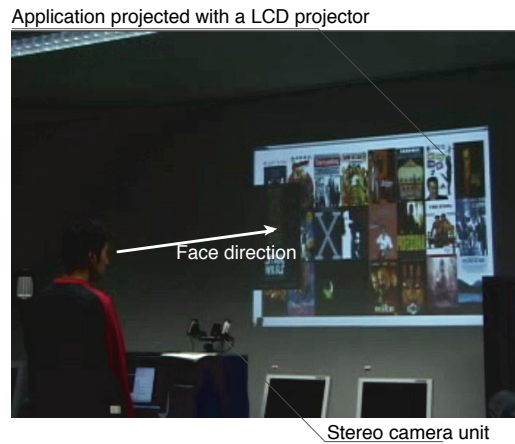
Figure 4. Using the one camera image in the template matching for the other camera image.

Table 1. Template images for each face direction mode.

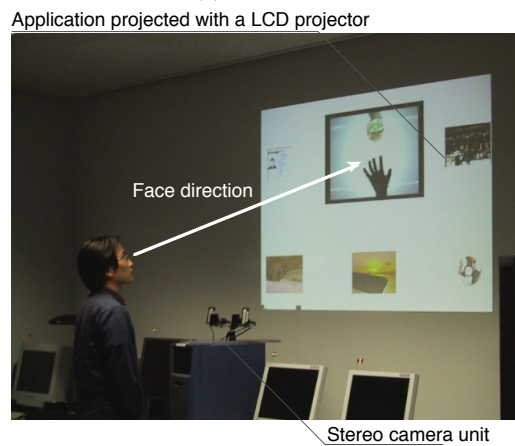
	center	right	left
template matching for the right camera image	initial template images	initial template images	new template images provide from the left camera image
template matching for the left camera image	initial template images	new template images provided from the left camera image	initial template images

With the current implementation of our face tracking system, a user’s face can be tracked for approximately 70 degrees in horizontal rotation and 45 degrees in vertical orientation. With this performance, users can look around an area the width of 140cm when they are 1m away from a display. While we have not yet done any careful optimization of the codes, our system runs at near video-frame rate, i.e., approximately 25-30 frames per second.

### 3 Visual User Interaction



(a)



(b)

Figure 5. Visual interfaces using our tracking system.

Large displays are often used in public spaces, and they are typically highly frequented but under-utilized. It may be that most of the displayed information is generally pictures or movies that are only repeated rather than being interactive. General computerized information kiosks show information interactively. Although users wanting information will utilize them positively, it is difficult to attract the interest of those who pass by them. Large displays in public spaces often show advertisements or information in the form of movies, and they would be catchier than general information kiosks. A face-tracking system would be able to give information to passive users or to those who happened to pass the display. Using a face-tracking

system and the focus+context techniques [6, 8] together would be one method to realize natural human computer interaction for such a case.

Our example applications were projected onto a wall with an LCD projector; the size of the projected area was 160cm x 120 cm. The face-tracking system and an application compose a client server system, and the face-tracking system sends the coordinate of the position in which a user gazed (turned his/her face toward) at the application.

Figure 5 (a) shows one application and includes twenty-four images. The gazed image is magnified a little. If the user keeps gazing at that image, it becomes larger and corresponding music sounds. It is a primitive example application with focus+context facilities. Figure 5 (b) shows another interface that utilizes focus+context techniques more actively. The gazed-at item is magnified with the fish-eye view technique which is one well known method in information visualization [8]. This technique distorts the layout to display both detail and context in a single view, and allows users to specify some information item of current interest, show the specified item in detail, and provide context by displaying the remaining items in successively less detail. This application was developed with Macromedia Director, and shows six QuickTime movies or Macromedia Flash files currently. By using the fish-eye view technique, the gazed item is magnified while the other items are reduced.

Our face-tracking system worked well so long as we used these applications, and it seems reasonable that we operate applications projected on a wall by our face directions. These applications are only examples which combined the face-tracking system with focus+context techniques. However, this combination would work effectively for the ubiquitous computing environment with large displays in public places. We will develop more useful applications in the near future.

## **4 Discussions and Conclusions**

In this paper, we presented a stereo-based face tracking system which can track the 3D position and orientation of a user in real-time, and the system's application for interaction with a large display. Our tracking system incorporates dynamic update of template images for tracking facial features so that the system can successfully track a user's face for a large angle of rotation. Another advantage of our tracking system is that it does not require a user to manually initialize tracking process, which is critical for natural and intuitive interaction. Based on our face tracking system, we have implemented several prototype applications which change information shown on a large display adaptively according to the location looked at by a user.

Currently, a user's gaze direction is determined based on the orientation of the user's face. We should improve our tracking system so that it can detect a user's pupils. If a display becomes large, the number of persons watching it will increase. However, our system tracks the face of a single user and does not work on multiple people simultaneously in the current implementation. We utilized low-resolution images because we gave more priority to tracking one face in real-time than to tracking multiple faces. When there are several users in front of our camera unit, it is

difficult to store high quality template images for each user in the current system. We will improve our system so as to track multiple users by processing more high-resolution images or by adding wide-angle lenses or more camera units. We use some fixed parameters for finding facial features and for processing template-matching method. Therefore, the sweet spot area in which our system tracks a user's face is not so large that he/she can move around in front of the system. By developing a system that knows the position where a user is standing, we will be able to change those parameters dynamically, thereby enlarging the area in which the user can move about.

## References

1. Christian, A.D. and Avery, B.L., Digital Smart Kiosk Project, Proceedings of ACM Human Factors in Computing Systems (SIGCHI '98), pp. 155-163 (1998).
2. Haro, A., M. Flickner, and I. Essa, Detecting and Tracking Eyes By Using Their Physiological Properties, Dynamics, and Appearance, Proceedings IEEE CVPR 2000, pp. 163-168 (2000).
3. Kawato, S. and Ohya, J., Two-step Approach for Real-time Eye Tracking with a New Filtering Technique, Proceedings of Int. Conf. on System, Man & Cybernetics, pp.1366-1371 (2000).
4. Kitajima, K., Sato, Y. and Koike, H., Vision-based face tracking system for window interface: prototype application and empirical studies, Extended Abstracts of 2001 ACM Human Factors in Computing Systems (SIGCHI 2001), pp. 359-360 (2001).
5. Matsumoto, Y. and Zelinsky, A., An Algorithm for Real-time Stereo Vision Implementation of Head Pose and Gaze Direction Measurement, Proceedings of IEEE Fourth International Conference on Face and Gesture Recognition (FG'2000), pp.499-505 (2000).
6. Pirolli, P., Card, S.K. and Van Der Wege, M.M., Visual Information Foraging in a Focus + Context Visualization, Proceedings of ACM Human Factors in Computing Systems (SIGCHI'2001), pp.506-513 (2001).
7. PosterCam, <http://crl.research.compaq.com/vision/interfaces/postercam/default.htm>
8. Sarkar, M. and Brown, M.H., GRAPHICAL FISHEYE VIEWS of GRAPHS, Proceedings of ACM Human Factors in Computing Systems (SIGCHI'92), pp.83-91 (1992).
9. Sawhney, N., Wheeler, S. and Schmandt, C., Aware Community Portals: Shared Information Applicances for Transitional Spaces, Journal of Personal and Ubiquitous Computing, Vol.5, No.1, pp.66-70 (2001).
10. Stiefelhagen, R., Yang, J. and Waibel, A., Tracking Eyes and Monitoring Eye Gaze, Proceedings of Workshop on Perceptive User Interfaces, pp. 98-100 (1997).