MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28-30, 2000, The University of Tokyo, Japan

3—14 Real-Time Camera Parameter Estimation from Images for a Wearable Vision System

Takashi OKUMA, Takeshi KURATA and Katsuhiko SAKAUE
Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Ibaraki 305-8568, JAPAN

Abstract

In this paper, we propose a method of estimating camera parameters for annotating 3D virtual objects with a wearable vision system. The proposed method tracks natural feature points using a doubled color histogram matching and template matching. The model-view matrix calculation method consists of three algorithms for PnP problems and uses each algorithm selectively. The experimental system using the proposed method works at 5 to 10 frames per second.

1 Introduction

This paper describes a method of estimating position and orientation of a camera from captured image sequences in real time. Recently, wearable computing attracts much attention with the progress of miniaturization of computers[1][2][3]. Enhancement of user's ability, which is one of the most important applications of the wearable computing, is implemented using information collection function and information display function that depend on the user's context.

On the other hand, mixed reality (MR) attracts much attention, because it enables us to add information on the real environment[4][5]. MR environment can be defined as an environment in which the virtual and real environments are composed in real-time. Some applications of wearable computing in combination with MR, such as 3D poster using annotation with 3D virtual object, require precise registration between virtual objects and real environments. In order to realize this geometric registration, three-dimensional (3-D) coordinates of the real and virtual environments should be aligned properly with respect to each other. For this purpose the user's viewing position and orientation must be precisely measured.

In section 2, we introduce the overview of the wearable vision system that we are developing. Then, implementation of the mixed reality system on the wearable vision system is examined. In section 3, camera parameter estimation method based on the natural feature tracking is proposed. In section 4, we describe the prototype mixed reality system using proposed method and show their output images.

2 Wearable Vision System

2.1 Overview of the system

We are developing a wearable vision system that works actually in order to examine merit, demerit and applications in the situation which users always wears the cameras in[2][3]. This system consists of wearable equipments, remote PC cluster and wireless LAN system. Figure 1 shows the appearance of the user who wears wearable equipments of the system.

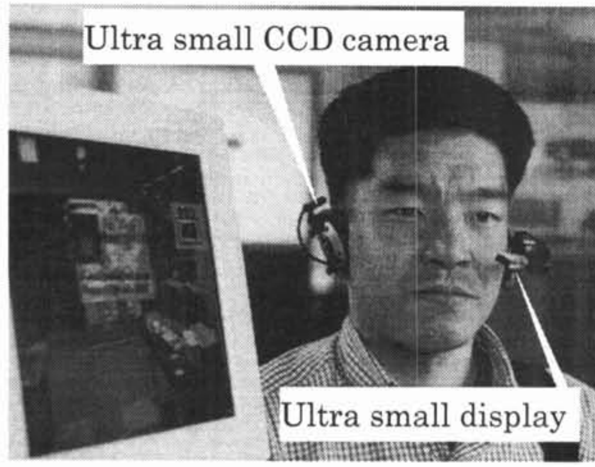

Figure 1: Appearance of a user of the system

The headset that users wear consists of ultra small CCD camera and ultra small display. This headset takes a style that doesn't hide eyes and face of the user. This style enables the system to capture the user's view and to display output images at the same time.

2.2 Mixed Reality

It is necessary to adjust a normal coordinate system of the real environment to a normal coordinate system of the virtual environment, in order to synthesize virtual objects in appropriate position at appropriate orientation. This adjustment can be realized by acquiring position and orientation of the user's viewpoint at normal coordinate system of the real environment. Virtual objects rendered with this viewpoint information are overlaid on the real world. Therefore, acquirement of the viewpoint information is necessary to implement mixed reality. In general, there exist the following two major methods to acquire the position and orientation of user's viewpoint.

- A method that uses 3-D position sensors, such as magnetic, ultra sonic and mechanical sensors, that used in virtual reality systems.
- A method that estimates the user's viewpoint by analyzing camera images taken at the viewpoint. This type of method is sometimes referred to as vision-based tracking or registration[5][6][7].

The hybrid of above two also improve the accuracy and the stability of geometric registration.

We noticed the latter method, because the equipments of the wearable vision system include a small CCD camera that captures images at user's viewpoint. The vision-based method estimates the viewing position from the 2-D positions on the image

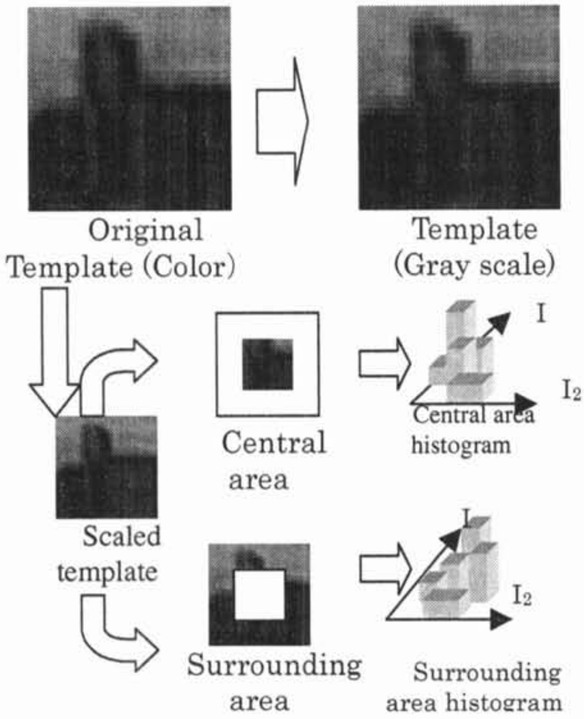

Figure 2 : Features used in matching process

plane (the screen coordinates) of known feature points. There is potentially no limitation in its measurement range.

Fiducial markers were set in the scene as known feature points in order to reduce the load on calculation in conventional vision-based registration method for the MR systems. It is necessary to enlarge the area in which fiducial markers are set in order to enlarge the measurement range with these systems. It is desirable that feature tracking method that doesn't use fiducial markers in order not to limit the measurement range. However, it is difficult to acquire the screen coordinates of the known feature points accurately because of noise, change of lighting condition and so on. It is also difficult to track the feature points stably because they are often occluded or go out of the view.

In this paper, we propose a technique to match features with doubled color histogram in order to track features without markers. We also propose the hybrid of doubled color histogram matching and template matching. The following section describes details of proposed method and a prototype system for annotating with 3D virtual objects on wearable vision systems using the proposed method.

3 Annotation with 3D Virtual Objects on Wearable Vision Systems

Our method estimates the camera parameters by the following four steps:

1. **[Preparation of a feature point information list]** A list that has sets of feature point information is prepared. A set of feature point information contains the color information and the 3-D position in the world coordinate system. Color information consists

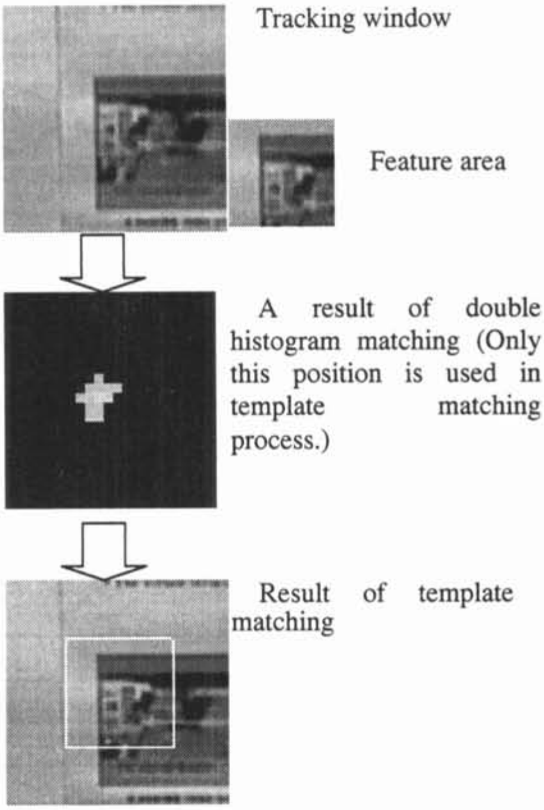

Figure 3 : Matching process

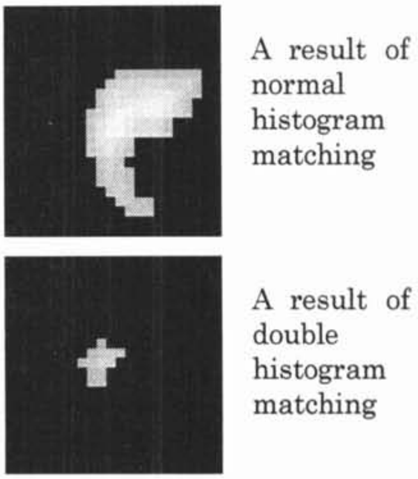

Figure.4 Comparison of results of normal and double histogram matching

of central area histogram and surrounding area histogram (see Figure 2).

2. **[Tracking of feature points]** Feature points of known positions are extracted and tracked based on template matching. For the sake of reducing the load on template matching calculation, candidates of matching positions are reduced by histogram matching using both of central area histogram and surrounding area histogram (see Figure 3). This "doubled histogram matching" reduces more candidates than normal histogram matching. (See Figure 4)

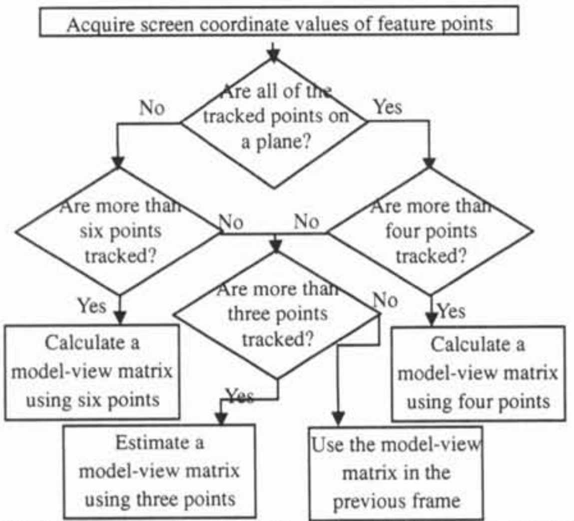

Figure 5 : A process of model-view matrix calculation.

3. **[Calculation of the model-view matrix]** A calculation method for a PnP (Perspective n-Points) is selected according to the number of tracked feature points. Then the selected algorithm calculates a model-view matrix that represents the relationship between the camera and world coordinates.
4. **[Estimation of the screen coordinates of untracked feature points]** The camera coordinates of the untracked feature points are calculated using the computed model-view matrix. Then the screen coordinates of the feature points are calculated using perspective projection of the camera coordinates.

The following sections describe these processes in more detail.

3.1 Natural Feature Tracking

It is necessary to acquire the screen coordinates of known feature points in the input images in order to estimate camera parameters. A lot of methods for feature point matching have been proposed. However, most of those only use the intensity values of pixels and not to use color information. We consider that the color information of the feature points is also important. Therefore, we notice the color histogram matching[8]. Color histogram matching is low calculation cost and stable. However, its accuracy of matching is not so high compare with that of the template matching, because color histogram matching is not to use the positional relation of the pixel.

Therefore, we use color histogram matching as preprocessing of the template matching. In additional, we use "doubled color histogram" for the color histogram matching. "Doubled color histogram" consists of the central area histogram and the surrounding area histogram (see Figure 3). We use the $I_1I_2I_3$ color space[9] to make each color histogram.

When the similarity (histogram intersection) of doubled color histogram are larger than threshold, template matching is done at the position.

Figure 3 shows an example of matching process. In this example, the number of times to do template matching is reduced to 2%. Figure 4 shows the position where template matching is done with the doubled color histogram and with the normal color histogram. In this example, the number of position where template matching is done with doubled color histogram is 13% of that with normal color histogram.

At the first frame, all of the pixels in the frame buffer are processed to find known feature points. Then, tracking windows are set around the positions of the found feature points.

In the second and subsequent frames, pixels in each tracking window are processed to track the feature point. Then, screen coordinates of the feature point are updated. When feature points are not extracted in the tracking window, the feature point is judged to be occluded or go out of the view. Then, the screen coordinates of these feature points are estimated later, using the model-view matrix that is calculated at the following process.

3.2 Calculation of the model-view matrix

The camera pose is expressed by a 4×4 transformation matrix that converts the world coordinate system to the camera coordinate system. This matrix is called a model-view matrix. Among a model-view matrix (M), a position of a point in world coordinate (w) and its position in the camera coordinate (c), the following equation stands.

$$\mathbf{c} = \mathbf{M}\mathbf{w}\,. \qquad (1)$$

Our method estimates the camera pose using screen coordinates of feature points of known 3-D positions[5]. It employs a combination of three calculation methods of PnP problems: The first is based on known six points, the second is based on known four points on a plane and the third is based on known three points as shown in Figure 5.

Step 3-1. If all of the tracked feature points are not on a plane, the process moves to Step 3-2, or else the process moves to Step 3-3.

Step 3-2. If more than five feature points are tracked, the process moves to Step 3-5, or else the process moves on to step 3-4.

Step 3-3. If four feature points on a plane are tracked, the process moves to Step 3-6, or else the process moves to Step 3-4.

Step 3-4. If more than two feature points are tracked, the process moves to Step 3-7, or else the process moves to Step 3-8.

Step 3-5. The model-view matrix is calculated using any six of the tracked feature points[10].

Step 3-6. The model-view matrix is calculated using the four feature points[11].

Step 3-7. Candidates of the model-view matrix are calculated using the three feature points[12], and then the model-view matrix is selected from them.

Step 3-8. The model-view matrix is set as the model-view matrix of the previous frame

3.3 Estimation of the screen coordinate of untracked feature points

After the model-view matrix is calculated, the screen coordinates of untracked feature points are calculated. First, the camera coordinates of untracked feature points are calculated with Eq. (1). The screen coordinates are then calculated using perspective projection of the camera coordinates onto the image plane. Note that the

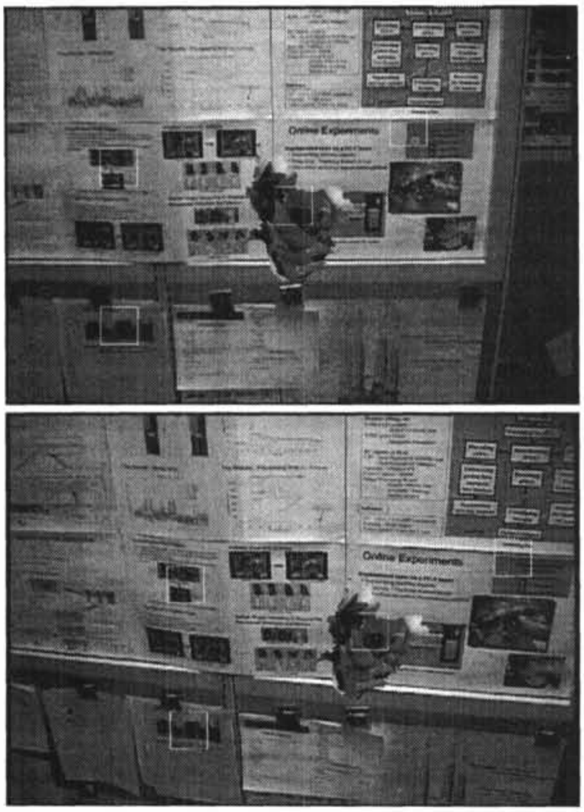

Figure 6 : Examples of output images (stuffed toy is CG)

screen coordinates may be out of view. Finally, the tracking windows are set around the calculated screen coordinates. This process makes it possible to start/restart tracking the feature points that has been out of the view and has been occluded.

4 Experiment and Discussion

The proposed algorithms are implemented on a PC (SGI VW540 Pentium III Xeon 500Hz 4CPUs). A video see-through function using the proposed algorithm is realized in the following experiment. Each digitized image consists of 720×486 pixels. It has been found that an experimental system that is based on the proposed method can seamlessly merge 3-D virtual objects into a 3-D real environment at right position in real-time. An example of output images is shown as Figure 6. In this experiment, four feature points are set in the list and they are tracked. Therefore, the feature points must be placed on a plane in order to calculate the model-view matrix at the first frame. This experimental system can synthesize virtual and real images at 5 to 10 frames per second.

5 Conclusion

We have proposed a method of estimating camera parameters for annotating 3D virtual object with the wearable vision system. The proposed method tracks natural feature points using a doubled color histogram matching and template matching. The model-view matrix calculation method consists of three algorithms for PnP problems and uses each algorithm selectively. The method also estimates the screen coordinates of untracked feature points that are occluded or are out of the view, so that the method allows users to look around an area in which feature points are set beforehand. The experimental system using the proposed method works at 5 to 10 frames per second.

In the future work, we should make error analyses of the proposed method and improve in accuracy and robustness. We should also investigate a method for selecting feature points and getting the feature points information automatically.

Acknowledgments

This work was supported by the Real World Computing (RWC) Program.

References

[1] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, W. R. Picard, A. Pentland, "Augmented Reality Through Wearable Computing," Technical Report 397, MIT Media Lab. Perceptual Computing Section, 1997.

[2] M. Kourogi, T. Kurata, K. Sakaue, and Y. Muraoka, "Improvement of Panorama-based Annotation Overlay Using Omnidirectional Vision and Inertial Sensors," In Proc. ISWC 2000, 2000.

[3] T. Kurata, J. Fujiki, M. Kourogi, K. Sakaue, "A Fast and Robust Approach to Recovering Structure and Motion from Live Video Frames," Proc. CVPR 2000, pp.528-535, 2000.

[4] Y. Ohta and H. Tamura Eds. , *Mixed Reality – Merging Real and Virtual Worlds,* Ohmsha & Springer-Verlag, 1999.

[5] T. Okuma, K. Kiyokawa, H. Takemura and N. Yokoya, "An Augmented Reality System Using a Real-time Vision Based Registration," *Proc. ICPR 98*, pp.1226-1229, 1998.

[6] U. Neumann and Y. Cho, "A Self-tracking Augmented Reality System," *Proc. VRST 96*, pp. 109-115, 1996.

[7] K. N. Kutulakos and J. R. Vallino, "Calibration-free Augmented Reality," *IEEE Trans. Visualization and Computer Graphics*, Vol.4, No.1, pp.1-20, 1998.

[8] M. J. Swain, and D. H. Ballard, "Color Indexing," International Journal of Computer Vision, Vol.7, No.1, pp.11-32, 1991

[9] Y. Ohta, T. Kanade, T. Sakai, "Color Information for Region Segmentation," Computer Graphics and Image Processing, Vol.13, pp.224-241, 1980.

[10] K. Deguchi, "A Unified Approach to PnP Camera Calibration Problem by Projective Geometry," Proc. IPSJ Symposium, Vol.90, No. 20, pp.41-50, 1990. (in Japanese.)

[11] Y. Nakazawa, S. Nakano, T. Komatsu and T. Saitou, "A System for Composition of Real Moving Images and CG Images Based on Image Feature Points," The Journal of the institute of image information and television engineers, Vol.51, No.7, pp.1086-1095, 1997. (in Japanese.)

[12] R. M. Haralick, C.-N. Lee and K. Ottenberg, "Analysis and Solutions of the Three Point Perspective Pose Estimation Problem," *Proc. CVPR '91*, pp.592-598, 1991.