A panorama-based technique for annotation overlay and its real-time implementation.

Masakatsu Kourogi†, Takeshi Kurata‡, Katsuhiko Sakaue‡ and Yoichi Muraoka†

†Graduate School of Science and Engineering, Waseda University

‡Eletrotechnical Laboratory

Abstract

The panorama-based annotation method described in this paper uses a panoramic image as the source of information about the positions of the annotations. It finds image alignment parameters between an input frame and the panoramic image and then maps the positions of annotations from the panoramic image to the input frame and displays the input frame overlaid with those annotations. Camera movement from place to place is made possible by preparing a set of panoramic images in advance. The panoramic image that gives the least mean squares error of the image alignment is selected automatically and is appropriately switched as the camera moves around. The position of the camera can be tracked by monitoring the switching of selected panoramic images. Experimental results show that this method can find image alignment parameters, display input frames overlaid with the annotations, and switch the panoramic image appropriately in real-time.

1 Introduction

Annotation overlay on a live video is an essential feature of augmented reality (AR), since it makes possible various kinds of such applications as augmented memory, touring assistance, and amusements [1][2]. Most previous work based on computer vision techniques, has used artificial markers (fiducials) that are placed on a real-world environment [3][4][5]. However, since these markers must be physically placed on every object to be annotated, it is generally difficult to cover a large-scale environment. And because the annotations need to be large enough to be detected in the image, fiducials cannot be effectively used to annotate large objects in an outdoor environment, such as mountains and buildings. Other previous works [6] have achieved annotation overlay in an outdoor environment by using a set of dedicated location/orientation sensors such as the satellite GPS and gyroscope. But the use of these sensors is restricted to appropriate environments. The GPS signals, for example, are blocked by buildings and thus cannot be used indoors.

This paper describes a novel method based solely on vision for annotation overlay without fiducials. It uses panoramic images for image alignment between positions of annotations and input video frames. The proposed method uses (1) a set of panoramic images acquired at various points in the environment, (2) annotations attached to the panoramas and (3) neighborhood relationships between panoramas as prior knowledge about the environment. The method estimates image alignment parameters between an input frame and each of the panoramic images. Then it selects the panorama that gives the least mean squares error of image brightness by the alignment, and overlays the annotations of the selected panorama on the frame. Experimental results show that this method can, with low-cost PCs, locate and orient input frames and display the frames overlaid with the annotations in real-time.

2 Panorama-based annotation overlay

Our approach uses a panoramic image to which annotations are manually attached as the source of information. When an input frame is given, it is aligned with the referred panoramic image. Then, we can map the positions of annotations from the panorama to the frame, and thus can create the frame overlaid with the annotations as shown in Figure 1.

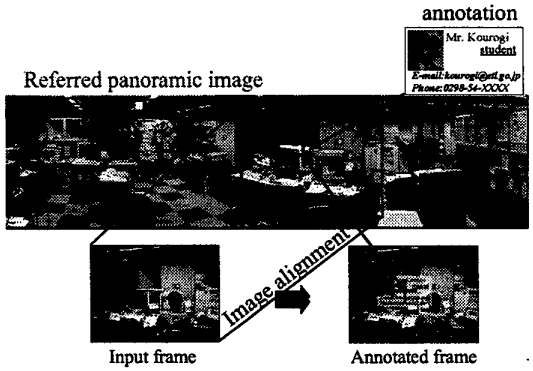

Figure 1: Panorama-based annotation overlay.

Two-dimensional image alignment, however, cannot handle the motion parallax caused by translational displacement of the viewpoint of a camera. If the viewpoint from which video frames are captured is translated apart from the one the panoramic image is acquired, it is in principle impossible to align the video frames with the panorama. To handle the displacement, we use a set of panoramic images acquired at various points in the environment in advance. Video frames are then aligned with the panorama acquired at the viewpoint nearest the one at which the video frames are captured. The referred panoramic image will be switched if necessary as the camera moves around. The neighborhood relationship between panoramas are given to enable the switching to occur. By tracking which panorama is referred, we

0-7803-6536-4/00/$10.00 (c) 2000 IEEE

can also estimate the position and trajectory of the user. An overview of the panorama-based approach is shown in Figure 2.

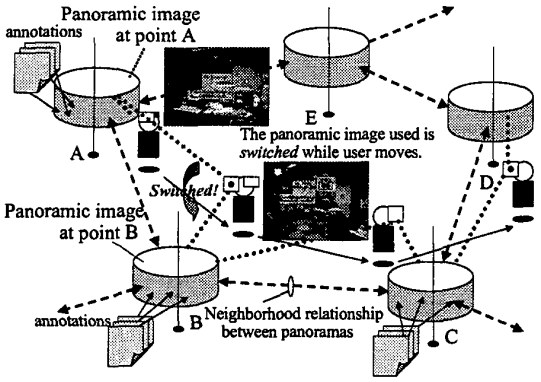

Figure 2: Overview of panorama-based annotation.

2.1 Image alignment between frame and panorama

We estimate image alignment parameters between an input frame and a referred panoramic image, by using a fast and robust gradient-based method [7] that can find affine or projective parameters of image alignment between images. We use affine model because of its stability of estimation. The affine transformation matrix $\boldsymbol{A}$ consists of 6-parameters, and can be written as follows:

$$\boldsymbol{A} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Since the method [7] is empirically known to have difficulties to stably estimate the parameters if translational component (a_3, a_6) is large, we give multiple initial estimates to the method and then estimate the parameters respectively from the initial estimates. Then, we select the estimated result that gives the best match from them.

Our method evaluates the results of image alignment in terms of the mean squares error (MSE) of image brightness between the images. Suppose that a point (x, y) on the frame is transformed to the point (x', y') on the referred panorama by the affine transform matrix $\boldsymbol{A}_{f \to P}$. Let $I_f(x, y)$ and $I_P(x', y')$ respectively be image brightnesses of the the frame and the panorama, let S be a set of points in the frame, and let N_S be the number of points in S. The MSE can be calculated by using the following equation:

$$MSE = \frac{1}{N_S} \sum_{(x,y) \in S} (I_f(x, y) - I_P(x', y'))^2. \quad (2)$$

Since MSE calculation is computationally expensive, the number of the pixels N_S is reduced by selecting those with large absolute gradients of image brightness.

2.2 Scale factor of image alignment

When the frame is aligned with the referred panorama by translation, rotation, and scaling, the scale factor s can be calculated as

$$s = \sqrt{\det \boldsymbol{A}_{f \to P}}. \quad (3)$$

This factor represents the ratio of the size of an object in the frame to the size of the object in the panorama. Let f and f_P be the focal lengths of the cameras that capture frames and the panorama, respectively. If the scale factor is close to f_P/f, it can be assumed that the position where the frame is captured is close to the one at which the panorama is acquired.

2.3 Evaluation criterion of image alignment

The MSE of image brightness is not necessarily by itself a good measure of image alignment between a frame and a set of panoramas, since the MSE can be small even when the scale factor is far from f_P/f. Our method therefore evaluates the result of image alignment as follows. If more than one results give MSEs smaller than a threshold, the method selects the one that gives the scale factor nearest f_P/f.

2.4 Neighborhood relationship between panoramas

To allow a user's camera to move around in the environment, neighborhood relationships are given to any two panoramic images if the positions where the images are acquired are proximate. We define the neighborhood relationship between two panoramas I_{P_i} and I_{P_j} by a data set as consisting of the following elements:

1. A region R_k in the panorama I_{P_i}.
2. The adjacent panorama I_{P_j}.
3. The affine transform parameters $\boldsymbol{A}_{R_k:P_i \to P_j}$ for the transformation from I_{P_i} to I_{P_j} on the region R_k.

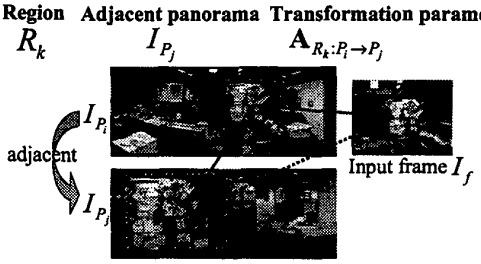

Figure 3: Neighborhood relationship.

A point (x_{P_i}, y_{P_i}) in the panorama I_{P_i} is transformed to the point (x_{P_j}, y_{P_j}) in the panorama I_{P_j} by using the affine transform matrix $\boldsymbol{A}_{R_k:P_i \to P_j}$ and the following equations:

$$\begin{bmatrix} x_{P_j} \\ y_{P_j} \\ 1 \end{bmatrix} = \boldsymbol{A}_{R_k:P_i \to P_j} \begin{bmatrix} x_{P_i} \\ y_{P_i} \\ 1 \end{bmatrix} \quad (4)$$

$$= A_{R_k:P_i \to P_j} A_{f \to P_i} \begin{bmatrix} x_f \\ y_f \\ 1 \end{bmatrix}. \quad (5)$$

When image alignment parameters between a frame I_f and the panorama I_{P_i} transforms the central position (x_{f_c}, y_{f_c}) of the frame to a point on the region R_k, the frame I_f is likely to match with the corresponding region in the panorama I_{P_j} as shown in Figure 3. Thus, the method estimates the image alignment parameters between the frame and the panorama I_{P_j} in parallel by using the product of two affine transform matrices $A_{R_k:P_i \to P_j} A_{f \to P_i}$ as an initial estimate. If image alignment between the frame and an adjacent panorama gives a result better than that given by the alignment between the frame and the currently referred panorama, the referred panorama for the next frame will be switched to the adjacent one.

The neighborhood relationships have to be given to any two panoramas on which switching can occur by the user's movement. Since the user is most likely to move in the direction he is looking, at least one neighborhood relationship needs to be given to the regions in the panoramas that the user can move to.

2.5 Searching for the panorama

To find the panoramic image that includes an input video frame, we give multiple initial estimates to each of the panoramas in the set, so that at least one of the initial estimate will be sufficiently close to the true parameters. Then we select the best combination of parameters and panorama by using the criterion described in Section 2.3. Once the best panorama-parameters pair is found, the parameters are used as the initial estimate for image alignment of the next frame.

3 Experiments

We implemented our method as software running on a PC cluster and evaluated it in experiments in which a user was equipped with a wearable display, a wearable camera, and a wearable computer.

3.1 Implementation

The software of our implementation uses the Parallel Virtual Machine (PVM) library [8] for data distribution and collection among PCs so that it can run independently from architecture and operating system of targeted computer. For high-performance computation, we used parallel computation based on multi-thread programming model: the POSIX thread for UNIX and the Win32 thread for Windows NT/95.

The unit of processing is the estimation, from one initial estimate, of the affine parameters of image alignment between a frame and a panorama.

In this software, the process of estimation is implemented as a thread code so that it can be speeded up by increasing the number of CPUs and PCs without having to rewrite the code.

As shown in Figure 4, we used a small CCD camera to capture input video frames and used a head-worn display (HWD) to show the output frames overlaid with annotations. A mobile PC held by the user captured and compressed the video data by a factor of 20 with JPEG encoding and transmitted it to the PC cluster via a wireless network complying with the IEEE 802.11 standard. The PC cluster consisted of four conventional PCs (CPU: two Dual PentiumII-450MHz, two Dual PentiumIII-500MHz, OS: Linux-2.2.11 SMP supported) connected via 100-M ethernet.

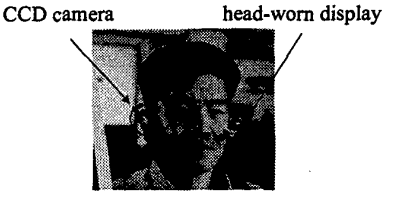

Figure 4: User with a HWD and a camera.

3.2 Creation of the prior knowledge

Before the experiments, a set of four panoramic images were acquired at point A–D in Figure 5 by panning a camera 360 degrees. These panoramas were created automatically by mapping the successive video frames to a cylindrical plane by using the method [7]. The created panoramas are shown in Figure 6. Then annotations were manually attached to the panoramas and neighborhood relationships between panoramas were given.

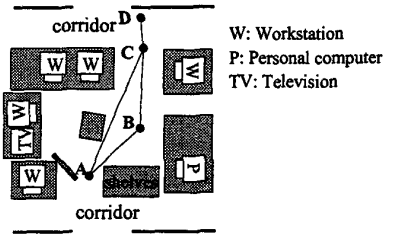

Figure 5: Environmental map.

3.3 Experimental results

In the experiments the user moved along the path A → B → C → D (Figure 5). The output video frames overlaid with annotations are shown in Figure 7, and the user's estimated position and orientation are shown in Figure 8 and 9.

It took 2000–3000 msec to search the set of four panoramas for the panoramic image that included the first input frame. The throughput of the annotation overlay was 100–120 msec or 8–10 frames per second and the delay is 600–800 msec.

The results of annotation overlay show that the software implementation of the proposed method can robustly provide video frames overlaid with annotations in real-time.

4 Conclusion

This paper describes an annotation overlay method that uses a set of panoramic images as a source of information. Experimental results show the software implementation of the method can provide video frames overlaid with annotations in real-time.

Acknowledgements

This work was conducted as a part of the Real World Computing (RWC) Program.

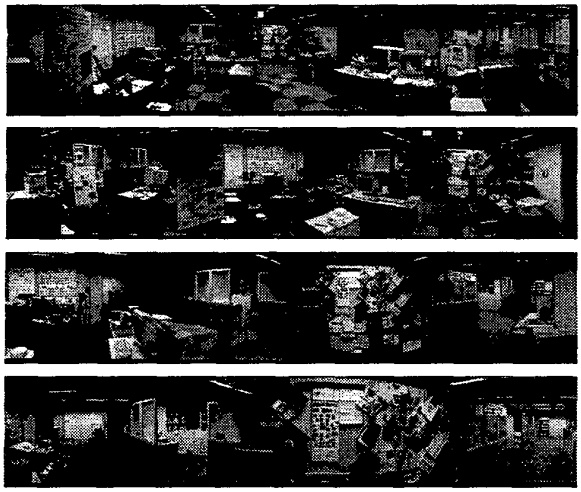

Figure 6: A set of panoramas acquired at point A–D (top to bottom).

Figure 7: Output frames overlaid with annotations.

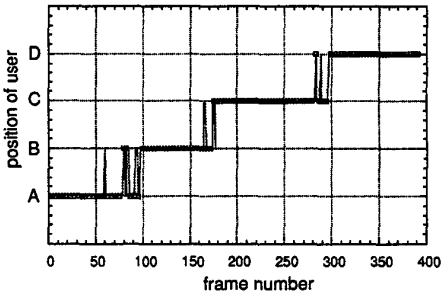

Figure 8: Estimation of the user's position.

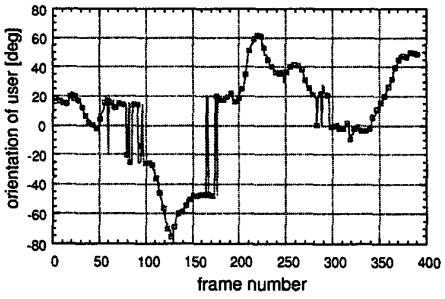

Figure 9: Estimation of the user's orientation.

References

[1] R. Azuma, "A survey of augmented reality," in *Presence*: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355–385, 1997.

[2] J. Rekimoto and K. Nagao, "The world through the computer: Computer augmented interaction with real-world environments," in *Proc. of UIST '95*, pp. 29–36, 1995.

[3] U. Neumann and J. Park, "Extendible object-centric tracking for augmented reality," in *Proc.of VRAIS'98*, pp. 148–155, 1998.

[4] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healy, D. Kirsh, R. Picard and A. Pentland, "Augmented reality through wearable computing," in *Presence*: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 386–398, August 1997.

[5] D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker and M. Tuceryan, "Real-time vision-based camera tracking for augmented reality," in *Proc. of VRST'97*, pp. 87–94, 1997.

[6] S. Feiner, B. MacIntyre and T. Höllerer, "Wearing it out: First steps toward mobile augmented reality systems," in *Mixed Reality - Merging Real and Virtual Worlds*, pp. 363–377, Ohmusha-Springer Verlag, 1999.

[7] M. Kourogi, T. Kurata, J. Hoshino and Y. Muraoka,"Real-time image mosaicing from a video sequence," in *Proc. of ICIP'99*, Vol. 4, pp. 133–137, 1999.

[8] A. Geist, "PVM: Parallel Virtual Machine, A users' guide and tutorial for networked parallel computing," MIT Press, 1994.