# Geometrical and Temporal Calibration of Multiple Cameras by Using LED Markers for Image Synthesis

**Hirotake Yamazoe[1,2], Akira Utsumi[1], Kenichi Hosaka[1] and Masahiko Yachida[2]**
[1]ATR Media Information Science Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
[2]Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama-cho, Toyonaka-shi, Osaka 560-8531, Japan
*yamazoe@atr.jp*

**Abstract**

In this paper, we propose a method to achieve position and pose of multiple cameras and temporal synchronization among them by using LED markers. In the proposed method, each IR marker transmits its own ID as a signal pattern. We can estimate camera positions and poses by using the 3D positions of multiple markers. In addition, these markers also transmit common synchronization signals and can be used in temporal synchronization among cameras. We confirmed through experiments the feasibility of marker detection and tracking by using normal CCD cameras, recognition of IDs, and synchronization information.

**Key words**: LED Marker, Camera synchronization, Camera position and pose estimation, Image synthesis.

## 1. Introduction

Recently, consumer video cameras such as DV/DVD cams, mobile phone cameras, etc. have become more popular, making video recording easy for individuals. Now we have more opportunities to take videos of daily life than ever before. At tourist spots and sporting events, people gather and use their video cameras. Therefore, often an identical scene is recorded by multiple video cameras independently.

In this paper, we consider the integration of independently recorded video sequences and apply 3D image processing for arbitrary view synthesis [1, 2, 3, 4]. To perform such processes, we have to determine the geometrical and temporal relationship among multiple video streams. In independent camera observations, it is difficult to set up fixed camera geometry and synchronization among multiple cameras beforehand. Therefore, we considered an approach that embeds geometrical and temporal information in individual video signals by using the blinking patterns of visual markers. This approach allows video camera users to freely record an event anytime and from any angle. Integration of multiple camera observations becomes a fully offline process. Most CCD color cameras also have sensitivity in IR (infra-red) rays. Since infra-red rays are invisible to humans, these markers will have less visual impact on humans than visible LED markers.

Systems employing similar approaches have already been proposed [5, 6, 7, 8]. For instance, Matsushita et al. proposed a method to capture images and marker IDs by using a high-speed image sensor [9]. However, this method requires a special camera device and cannot be applied with normal camera systems. Aoki et al. proposed a shape-based search method that uses a set of LED markers placed in a rectangle that identifies markers in video images [10]. This method can be applied to normal camera systems, but the shape-based search restricts the number of markers and camera locations. Our method encodes both timecode and marker IDs as blinking patterns of LED markers. Such information is recorded in video signals as a part of normal video images. We can determine 3D camera geometries and synchronize multiple cameras by using that stored information.

In the next section, we describe an overview of our system. In section 3, we focus on the image processing algorithm used for marker detection. Section 4 gives some experimental results and section 5 concludes the paper.

## 2. System Overview

Here, we briefly describe our system. Figure 1 shows the system's configuration. Our system consists of multiple LED markers and a server machine that controls the blinking patterns of these markers. They are distributed around the target scene where independently operating multiple video cameras observe and record the target scene as sequential image data that include the markers' blinking.

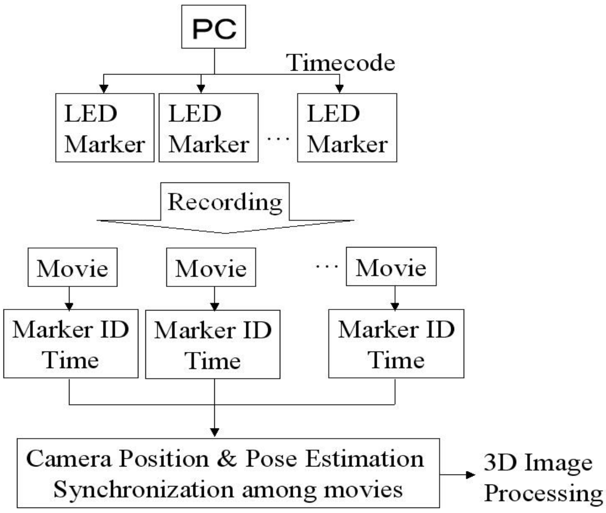


**Figure 1: System Configuration**

Figure 2 shows a prototype module of the IR marker. As described above, markers transmit both their own ID and timecode information shared within the entire system through blinking patterns. The interval of the blinking changes is set to 29.97 Hz to match the video signals.

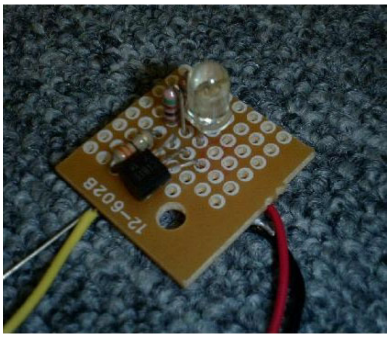

**Figure 2: LED marker**

Figure 3 shows the process flow of marker detection. First, we detect marker candidate points from each frame of an image sequence by using a template matching method. Then we track trajectories of multiple markers by assuming a linear motion of cameras in a short sequence. Since the LED markers have characteristic in blinking patterns, we can omit the false signals.

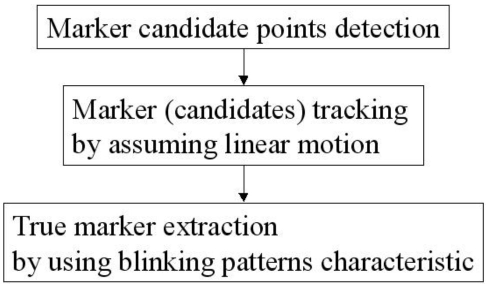


**Figure 3: Process flow**

After that, we extract a timecode signal from the blinking patterns of multiple markers. As shown in Figure 4, all markers transmit common timecode signals periodically. Consequently, we can achieve synchronization among multiple image sequences based on the time code.

Other parts of the blinking patterns contain the ID of each marker. If each marker has a known 3D position, we can easily determine the 3D camera motions from the observed marker positions in the camera images. In the next section, we show some results of camera position estimation.

In the next section, we describe the details of each process.

## 3. Position and Pose Estimation by Using LED Markers

### 3.1 Encoding of ID and timecode information

In this section, we describe how to encode marker ID and timecode information in the marker's blinking pattern.

In the current implementation, we encode marker ID and timecode information to 30-bit code (1 [sec]). Figure 4 shows the format of the encoded data. We use 20 bits for ID information and 10 bits for timecode information.

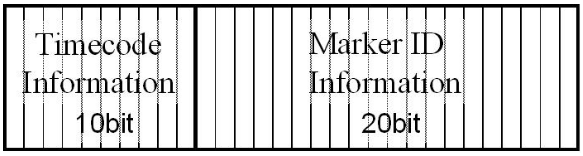


**Figure 4: Format of encoded data**

We convert time information (UNIX time) to a hexadecimal number (4 digits). Each digit is encoded to 10-bit redundant code, and it takes 4 seconds to send all timecode information. Since all markers blink at the same pattern in the timecode part and each camera observes multiple markers in each image, we can extract the time code part from blinking patterns by detecting the part in which multiple markers blink with the same pattern. After that, we can determine the marker ID information part and extract the marker ID of each marker (Figure 5). We use 256 LED markers (8-bit IDs) and marker IDs are encoded to 20-bit redundant codes.

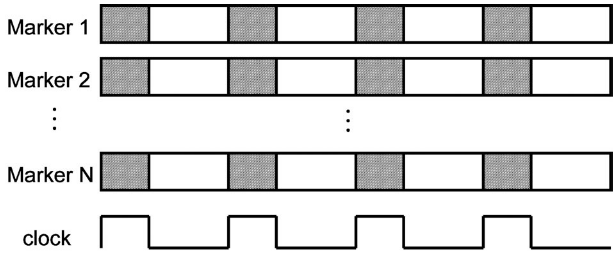


**Figure 5: Time information**

To distinguish LED markers from other blinking pixels (false candidates) in input images, the blinking rate of each marker $p_{on}$ should be within the defined range ( $p_L < p_{on} < p_H$ ). In addition, each marker should blink at least once within n frames (n=3) for robust marker tracking. Both ID information and timecode information are encoded in the hamming codes satisfying the conditions described above.

### 3.2 Marker candidate detection

Here, we describe the marker candidate detection process from input images observed in scenes including LED markers.

We assume that LED markers have high pixel values compared with surrounding pixels in the input images. First, we scan the luminance value $V_t$ of the input image $I_t$ observed at time t by using the template T shown in Figure 6. Then, we detect marker candidate points as 2D positions (x, y) that have a larger score in Equation 1.

$$\sum_{i=-m}^{m}\sum_{j=-n}^{n} V(x+i, y+j)\cdot T(i,j) > \varepsilon . \qquad (1)$$

Here, $\varepsilon$ is the threshold luminance value for marker detection. In the current implementation, we employ two different sizes of templates, $5\times 5$ and $7\times 7$ .

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6: Templates for marker detection.**

In the next section, we describe the marker tracking process by using the marker's blinking patterns.

### 3.3 Marker Tracking

We need to extract real marker trajectories from multiple marker candidates observed in input images. First, we track marker candidates in each frame. Next, we omit false marker sequences by using the marker's blinking characteristics as follows.

Since we assume that the 2D linear motion of each marker in images in short sequences, real marker trajectories should satisfy the conditions below.

Let $x_p^{(t)}$ and $v_p^{(t)}$ be the 2D observed position and velocity of real marker p at time t respectively. From the arbitrary observation set up to time t, we can estimate $x_p^{(t)}$ and $v_p^{(t)}$ . If these parameters belong to a real marker, a marker candidate $x^{(t+\Delta t)}$ that satisfies equation (2) should be obtained in the next frame (time $t+\Delta t$ ) (Figure 7).

$$x^{(t+\Delta t)} - x_p^{(t)} + \Delta v_p^{(t)} < \varepsilon_x . \qquad (2)$$

When no marker candidate satisfies equation (2), we assume that the sequence is not a real marker or the marker p does not blink at time $t+\Delta t$ . According to the marker's blinking pattern, no marker is off for more than n frames. If a marker cannot be observed for more than n frames, it should not be a marker.

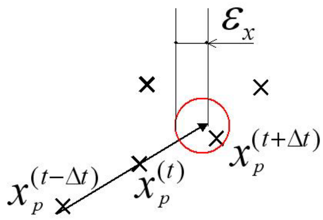


**Figure 7: Marker tracking**

### 3.4 Decoding of marker ID and time information

First, we extract the timecode information part in which all of the observed markers blink with the same patterns from the signal sequences. Since the timecode parts are transmitted periodically, we can easily extract them. Next, we extract the part between two timecodes as the marker ID information.

### 3.5 Camera position and pose estimation

From the obtained information, we can estimate the positions and poses of multiple cameras. In the current implementation, we employ a homography-based method to estimate relative positions and poses of multiple cameras [11]. In this section, we briefly describe the process.

We assume that 3D points $P_1,\ldots,P_N$ (i.e. LED markers) are distributed on a plane and these points are observed with both cameras $C_k$ and $C_L$. $\boldsymbol{x}_{k,1},\ldots,\boldsymbol{x}_{k,N}$ and $\boldsymbol{x}_{L,1},\ldots,\boldsymbol{x}_{L,N}$ are the projected 2D positions in $C_k$ and $C_L$ respectively. Here, we define observation vectors $\boldsymbol{m}_{k,i}$ and $\boldsymbol{m}_{L,i}$ as follows (Figure 8).

$$\boldsymbol{m}_{K,i} = \frac{1}{f_K}\begin{bmatrix} x_{K,i} \\ y_{K,i} \\ f_k \end{bmatrix}, \boldsymbol{m}_{L,i} = \frac{1}{f_L}\begin{bmatrix} x_{L,i} \\ y_{L,i} \\ f_L \end{bmatrix} \tag{3}$$

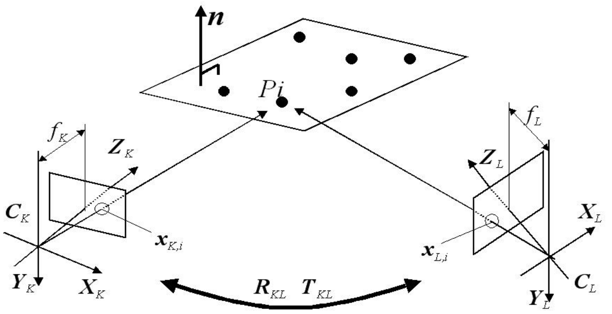


**Figure 8: Homography**

Using these 2D positions, we can obtain the following equation.

$$\boldsymbol{m}_{L,i} = a_i \boldsymbol{H}_{KL} \boldsymbol{m}_{k,i} \quad , \tag{4}$$

where $\boldsymbol{H}_{KL}$ is called the homography between $C_k$ and $C_L$, and $a_i$ is a constant.

$$\boldsymbol{H}_{KL} = d\boldsymbol{R}_{KL} + \boldsymbol{T}_{KL}\boldsymbol{n}^t \quad . \tag{5}$$

Once we obtain $\boldsymbol{H}_{KL}$ ,we can easily decompose it to relative position $\mathbf{R}_{KL}$ and pose $\mathbf{T}_{KL}$.

## 4. Experiment

To confirm the effectiveness of the proposed method, we performed the following experiments.

First, we placed six IR markers in an experimental scene (Figure 9). Figure 10 shows the experimental environment. Then, we captured an image sequence of the target scene using a CCD camera (ELMO CN-42H) with both translation and rotation motion. We applied the proposed method to these image sequences to estimate camera positions / poses.

Figure 11 shows the extracted marker trajectories and Figure 12 shows the extracted blinking patterns of all observed markers. In Figure 12, the shaded areas denote the extracted timecode part. Marker IDs are obtained in the part between two timecode parts.

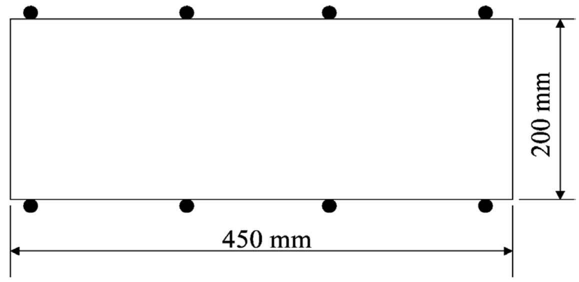


**Figure 9: Marker configuration.**

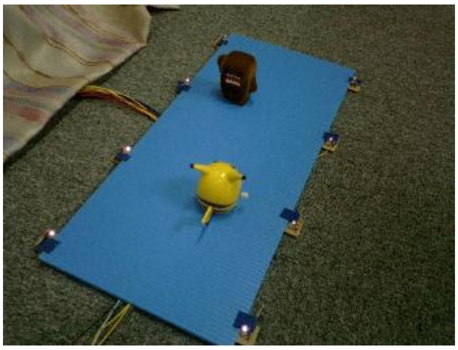

**Figure 10: Experimental environments**

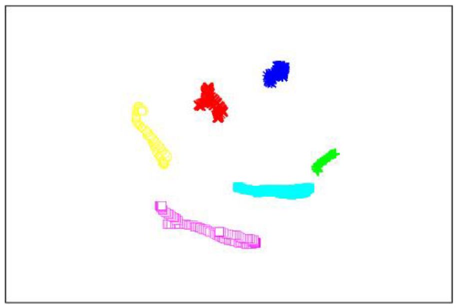

**Figure 11: Extracted marker trajectories**

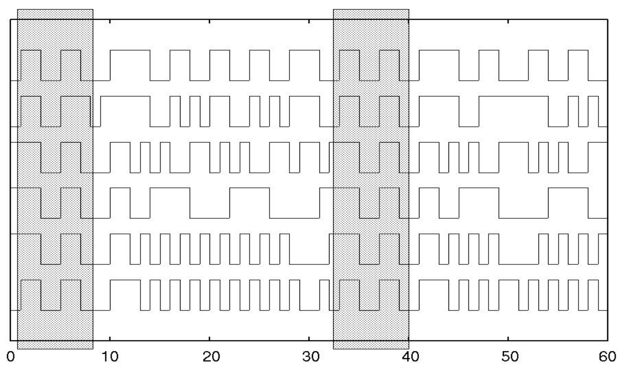


**Figure 12: Marker blinking pattern extraction results**

Next, we performed camera position and pose estimation by using 2D marker positions (here, the intrinsic parameters of all cameras are known). We use two CCD cameras. One camera is fixed and used for reference coordinates, and the other is used for position and pose estimation. Figure 13 shows the relationship between the two cameras. In the initial state, the positions and poses of the two cameras, except for the height, are the same. Then, camera 2 pans and moves along the X-axis

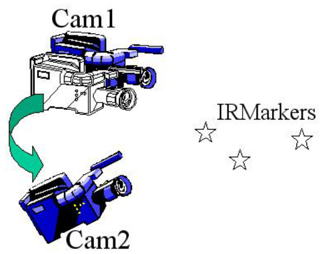


**Figure 13: Camera arrangement**

Figure 14 shows the 3D relative positions and poses between camera 1 and camera 2 as estimated by the 2D marker positions and IDs (here, the intrinsic parameters of the camera are known). As can be seen, we can estimate 3D camera geometries properly by using the proposed method. These results are useful for various 3D graphic applications.

Finally, we performed a temporal synchronization experiment using a DV camera (Panasonic NV-GS200). In this experiment, we used two DV cameras. Figure 15 shows the synchronized images. As can be seen, the two camera system can be properly synchronized by using the proposed method.

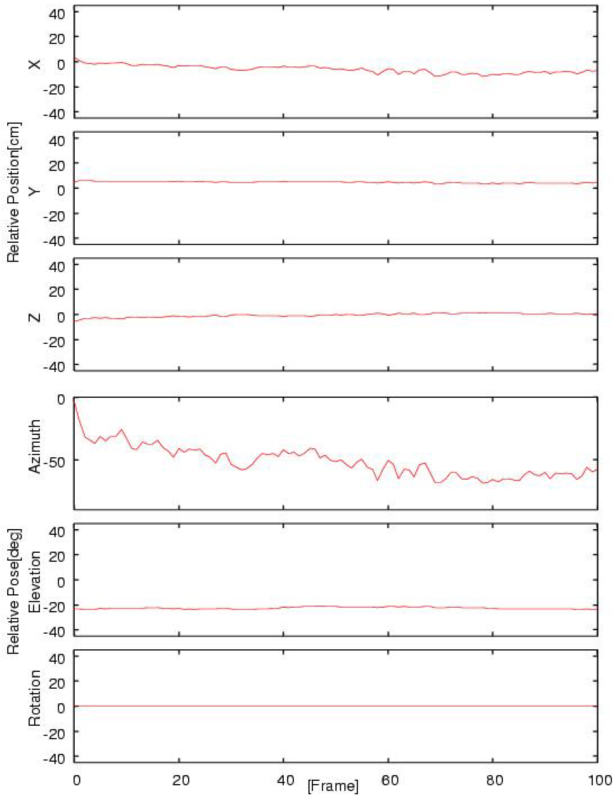


**Figure 15: Camera position and pose estimation results**

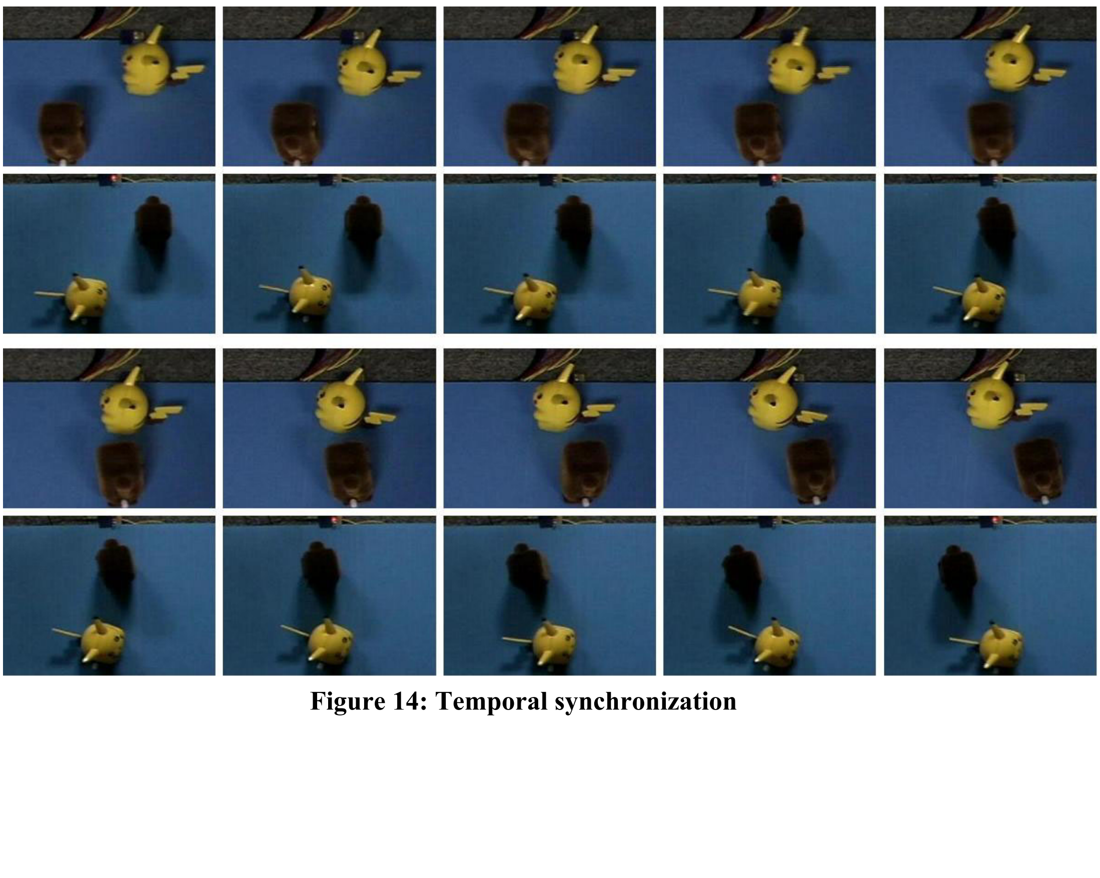

**Figure 14: Temporal synchronization**

## 5. Conclusion

We proposed a method to estimate 3D positions and poses of multiple cameras and to achive time synchronization among cameras by using multiple LED markers. In our system, the blinking patterns of multiple markers are used to transmit both timecode information and their own IDs. Timecode information can be easily extracted as common signal patterns among all markers. The ID number of each marker is extracted from the rest part of the signal patterns. 3D positions of markers, then, become available for the calculation of 3D camera positions and poses. We confirmed the effectiveness of our system through experiments.

Future work includes the implementation of the proposed method by using larger and brighter markers for outdoor scenes. Various multi-view based 3D image processing method using extracted camera geometries should also be addressed.

## Acknowledgement

This research was supported in part by the National Institute of Information and Communications Technology (NICT).

## References

1. R.Azuma. SIGGRAPH95 Course Notes: A Survey of Augmented Reality. 1995.

2. A.State, G.Hirota, D.T.Chen, W.F.Garrett, and M.A.Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. Proc.of SIGGRAPH96, pp.429-446, 1996.

3. J.Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. Proc. Og Asia Pacific Computer Human Interaction 1998, pp.15-17, 1996

4. U.Neumann, S.You, Y.Cho, and J.Park. Augmented reality tracking in natural environments. Y.Ohta and H.Tamura, editors, Mixed Reality – Merging Real and Virtual Worlds, pp.101-130, 1999.

5. Y.Sumi, T.Matsuguchi, S.Ito, S.Fels and K.Mase. Collaborative capturing of interactions by multiple sensors. Proc. of Ubicomp 2003, pp.193-194, 2003.

6. D.J.Moore, R.Want,. B.L.Harrison, A.Gujar, and K.Fishlin. Implementing phicons: Combining computer vision with infrared technology for interactive physical icons. Proc. of UIST'99, pp.66-67, 1999.

7. H.Kato and M.Billingshurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. Proc. of the 2nd Int. Workshop on Augmented Reality (IWAR99), 1999.

8. J.Rekimoto and Y,Ayatsuka. Cybercode: Designing augmented reality environments with visual tags. Proc. of Designing Augmented Reality Environments (DARE2000), 2000.

9. N.Matsushita, D.Hirata, T.Ushiro, S.Yoshimura, J.Rekimoto, and Y.Yamamoto. ID Cam: A smart camera forscene capturing and ID recognition. Proc. of Int. Symp. On Mixed and Augmented Reality (ISMAR03), pp.227-236, 2003.

10. H.Aoki and S.Matsushita. Balloon tag: (in)visible marker which tells who's who. Proc. of ISWC'00, pp.181-182, 2000.

11. J.Weng, N.Ahuja, and T.S.Huang. Motion and structure from point correspondences with error estimation: Planar surfaces. IEEE Trans. Signal Processing, Vol.39, No.12, pp.2691-2717, 1991.