Automatic Camera Calibration Method for Distributed Multiple Camera Based Human Tracking System

Hirotake Yamazoe[†‡], Akira Utsumi[†], Nobuji Tetsutani[†] and Masahiko Yachida[‡]
[†]*ATR Media Integration & Communications Research Laboratories*
2-2-2 Hikaridai Seikacho Sorakugun, Kyoto 619-0288, Japan
{utsumi,tetutani}@mic.atr.co.jp
[‡]*Department of Systems and Human Science*
Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama-cho Toyonaka-shi Osaka 560-8531, Japan
yamazoe@yachi-lab.sys.es.osaka-u.ac.jp
yachida@sys.es.osaka-u.ac.jp

Abstract

We propose an automatic camera calibration method to determine the position and orientation of a newly installed camera in our human tracking system. Due to the increasing cost of camera calibration, automatic algorithms are required. In our method, both 3D position tracking results and 2D positions and sizes on camera image planes are used. Because we employ a simple 2D to 3D position translation method based on the observed size of the target, the calibration results become stable even if the input data includes large observation errors. Observations by the newly installed camera are taken into account while tracking to increase the reliability of the calibration result. Using this method, the cost of installing new cameras in a human tracking system can be effectively reduced. Experimental results show the effectiveness of the proposed method.

1. Introduction

We have been investigating a vision-based human motion tracking system for a non-contact computer interface [1, 2]. The targets of our motion tracking system include a face image, body height, and shirt color for human identification. By detecting such motion information, the system can be broadly applied, such as for interaction in virtual environments, in information providing systems able to respond to a user's position, and in surveillance systems.

Many human tracking systems using images have been proposed to date [3, 4, 5, 6]. These systems are usually based on one or two viewpoints. In human motion tracking, a restricted number of cameras in a system can often cause problems such as occlusions, a small detection area, and an insufficient accuracy. However, a human tracking system using multi-viewpoint images can reduce occlusions, and can be expected to provide robust detection [7, 8, 9].

Unfortunately, such a system requires many cameras to track human movements in a wide area. This produces several problems. As many vision systems assume simultaneous observations by all cameras for 3D measurement, a synchronous mechanism (for the observations) is required. In synchronous systems, however, the processing efficiency becomes worse with an increasing number of cameras due to the processing time differences among the cameras. Furthermore, redundancy among multiple observations increases with an increasing number of cameras. To reduce these problems, we earlier proposed a system based on nonsynchronous multiple observations [10].

Another problem is that the calibration cost becomes higher as the number of viewpoints (the number of cameras) is increased. We think that the scalability produced by system expansion can be significant in a tracking system using multi-viewpoint images. Accordingly, automatic camera calibration should be a key technology here.

Lee et al. have proposed a method to determine geometrical relations among multiple cameras [11]. This method requires coplanar image features and a non-linear algorithm for the calculations. We have proposed an automatic camera calibration technique using target motion models [12]. We linearized our calibration algorithm based on the observed sizes of targets. The estimations of target motion are given by the tracking system [10]. In this paper, we expand the method to include accuracy evaluations. Using this technique, each newly installed camera can join the tracking process automatically based on an estimated calibration accuracy and we can easily add/delete observation nodes (cameras) to/from our human tracking system.

2. System Configuration

2.1. Process Flow

This section describes the outline of our multiple camera based human tracking system. The system configuration and process flow are shown in Figure 1. First, the observation nodes perform feature extraction on the input images obtained independently from every camera.

After matching between these features (represented position, head position, foot position, and the body region color), and after the tracking node sends predicted observation positions, the observation nodes send these features and the observation time to the tracking node. The remaining (unmatched) features are sent to the discovering node. Each observation node runs independently.

The discovering node can detect a human that newly appears in the scene using the unmatched information sent from the observation nodes. Each new human detection result is sent to the tracking node, and tracking is started.

In the tracking node, the new human information and observation information are the initial value and input value, respectively, and the human state (position, direction angle, height, etc.) is updated by using a Kalman filter. Moreover, the face region is detected using the position and direction angle estimation results.

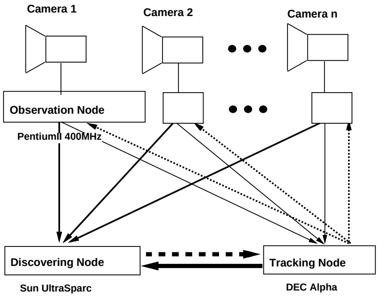


Figure 1. System Diagram

2.2. Observation Node

This section briefly explains the processing of the observation nodes. First, an input image is divided into a human region and a background region [13], and a distance transformation is applied to the human region. Each pixel of the distance transformed image has a certain distance from its nearest boundary pixel (Figure 2 right). We extract those pixels having the maximum value in the transformed image as representative points of the region. Furthermore, the position of a head point and the colors of body parts are extracted from the image (Figure 3).

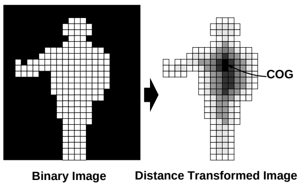


Figure 2. Feature Extraction

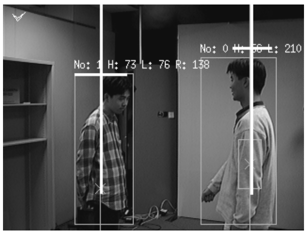


Figure 3. Example of Feature Extraction

Next, we describe the method of matching between the tracking targets (model) already discovered and the extracted features. Linear uniform motions of a human p_j are assumed in the tracking node, and the predicted position of human p_j at time t is expressed by a 2D Gaussian distribution. Here, the average of the distribution is set to $\bar{\boldsymbol{X}}_{p_j,t}$ and the covariance is set to $\bar{\boldsymbol{S}}_{p_j,t}$.

The weak perspective projection of the distribution $N(\bar{\boldsymbol{X}}_{p_j,t}, \bar{\boldsymbol{S}}_{p_j,t})$ of a predicted position to an image i is identical to the following 1D Gaussian distribution $n(\bar{\boldsymbol{x}}_{p_j,t,i}, \bar{\boldsymbol{s}}_{p_j,t,i})$, and this shows the human existence probability on image i.

$$P_i(x_i) = \frac{1}{2\pi \bar{\boldsymbol{s}}_{p_j,i}} exp\left(\frac{-(x_i - \bar{\boldsymbol{x}}_{p_j,i})^2}{2\bar{\boldsymbol{s}}^2_{p_j,i}}\right) \qquad (1)$$

A feature point with the maximum occurrence probability of an observation to the tracking model is considered to be an observation of human p_j, and is labeled p_j[10]. The labeled feature point is sent to the tracking node as observation information. However, a feature point matched with two or more humans is judged to be occluded at the time of the observation, and is not sent.

The observation information (positions, times) of unmatched feature points is sent to the discovering node.

2.3. Discovering Node

The discovering node can detect a human that newly appears in the scene. And, a matched model is added to the tracking node. Since observation information is acquired asynchronously, the usual stereo matching can not be used. Instead, a matching (discovery) technique using Kalman filtering according to series information is used [10].

Information observed at four different times is selected from the unmatched observed information sent from the observation nodes. The Kalman filter is then updated. If the errors between the predicted trajectory and each observed information are less than a threshold, this information becomes the feature point set belonging to the new human. The latest estimated position of the human is sent to the tracking node as the initial discovery position of the human.

2.4. Tracking Node

Here, the human models under tracking are updated using the image features matched with the models in each observation node [10].

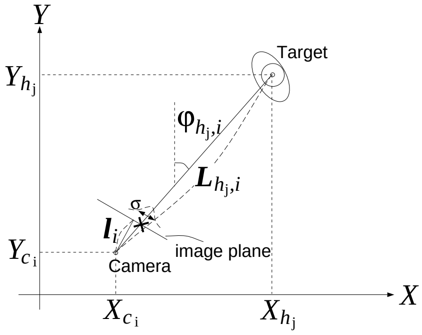


Figure 4. Observation Model

Let us consider uniform movements and express the state of human p_j as $\boldsymbol{X}_{p_j,t} = [X_{p_j,t} Y_{p_j,t} \dot{X}_{p_j,t} \dot{Y}_{p_j,t}]'$ in the world coordinates (X, Y)(Figure 4). $\dot{X}_{p_j,t}$ is the speed of direction X. The initial state is determined by the new model information sent from the discovering node.

Let us consider the situation where the target is observed one time with camera i. With the information sent from observation node i, this observation is expressed as follows.

$$\boldsymbol{H}\boldsymbol{R}^{-1}_{\varphi_{p_j,i}}\boldsymbol{C}_i = \boldsymbol{H}\boldsymbol{R}^{-1}_{\varphi_{p_j,i}}\boldsymbol{X}_{p_j} + e \qquad (2)$$

Here, $\boldsymbol{C}_i$ is the position of the camera, and $\boldsymbol{R}_{\varphi_{p_j,t,i}}$ is the clockwise rotation of angle $\varphi_{p_j,t,i}$, which is made by an epipolar line and the Y axis. Here, e is the observation error.

The above observation model constitutes a Kalman filter and the state of human p_j is updated.

The updating process and prediction of the human state are performed for every camera independently. The state prediction of human p_j at time $t+1$ is expressed by a Gaussian distribution, the mean is $\bar{\boldsymbol{X}}_{p_j,t+1}$, and the covariance matrix is $\bar{\boldsymbol{S}}_{p_j,t+1}$. The result of the state prediction is calculated and sent on demand to the observation nodes, and is utilized for the feature point matching stated above. A human model moving out of a detection range is deleted, and the tracking of the human is stopped.

An example of a position tracking result is shown in Figure 5. The dashed line shows the setting value and the solid line shows the estimated result. Generally, the results showed stable tracking. performed.

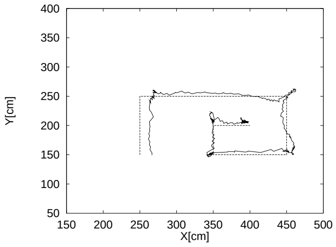


Figure 5. Tracking Result

3. Camera Calibration

Here, let us consider the case that an uncalibrated camera (i.e., the position and direction of the camera are unknown) is added to the system. We assume that people walk around in the scene, and also that the motions of the people are observed by several cameras. The observation node that has the uncalibrated camera can obtain 3D information of the people from the tracking node. In addition, the position and size information of feature points regarding a human image can be observed on an image plane of the uncalibrated camera itself. In this section, we describe a camera calibration algorithm using such information.

3.1. Camera calibration algorithm

We assume that $\boldsymbol{X}_1 \dots \boldsymbol{X}_n$, the 3D positions of feature points of the tracking target, $\boldsymbol{x}_1 \dots \boldsymbol{x}_n$, the observed position in the image, and $w_1 \dots w_n$, the sizes of the observed target in the image, are obtained (the absolute 3D size of the target is unknown).

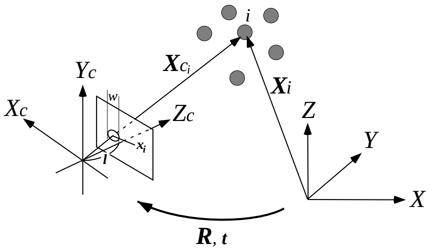


Figure 6. Feature Point Projection

We also assume that the intrinsic camera matrix of a camera $\boldsymbol{A}$ is given. Then, the relationship of $\boldsymbol{X}$, the 3D positions of feature points in the world coordinates, and $\boldsymbol{x}$, the observed positions in the image, is expressed as follows. Rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ express the position and direction of the camera in the world coordinates.

$$\begin{aligned} s\begin{bmatrix}\boldsymbol{x}\\1\end{bmatrix} &= \boldsymbol{A}\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\end{bmatrix}\begin{bmatrix}\boldsymbol{X}\\1\end{bmatrix} \\ &= \begin{bmatrix}\boldsymbol{a}_1\\\boldsymbol{a}_2\\\boldsymbol{a}_3\end{bmatrix}\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\end{bmatrix}\begin{bmatrix}\boldsymbol{X}\\1\end{bmatrix}. \qquad (3)\end{aligned}$$

Here, $\boldsymbol{a}_3 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$.

Let $\boldsymbol{X}_c$ be the 3D positions of feature points in the camera coordinates.

$$\boldsymbol{X}_c = \begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\end{bmatrix}\begin{bmatrix}\boldsymbol{X}\\1\end{bmatrix}. \qquad (4)$$

$$s\begin{bmatrix}\boldsymbol{x}\\1\end{bmatrix} = \boldsymbol{A}\boldsymbol{X}_c. \qquad (5)$$

From the 2D positions of the observed features on image plane $\boldsymbol{x}$ and the size of feature points $\boldsymbol{w}$, $\boldsymbol{X}_c$ can also be expressed as follows.

$$\begin{aligned} \boldsymbol{X}_c &= s\boldsymbol{A}^{-1}\begin{bmatrix}\boldsymbol{x}\\1\end{bmatrix} && (6)\\ &= s\begin{bmatrix}\boldsymbol{a}_{m1}\\\boldsymbol{a}_{m2}\\\boldsymbol{a}_{m3}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\1\end{bmatrix} && (7)\end{aligned}$$

Then, Z_{x_c}, the z elements of $\boldsymbol{X}_c$, can be expressed as follows.

$$Z_{x_c} = s\boldsymbol{a}_{m3}\begin{bmatrix}\boldsymbol{x}\\1\end{bmatrix} \tag{8}$$

Assuming a weak perspective transformation about the image feature, Z_{x_c} can also be expressed with the width of feature points w,

$$Z_{x_c} = \frac{k}{w} \qquad \text{(k is constant)} \tag{9}$$

Therefore,

$$s = \frac{1}{\boldsymbol{a}_{m3}\begin{bmatrix}\boldsymbol{x}\\1\end{bmatrix}} \times \frac{k}{w} \tag{10}$$

$\boldsymbol{a}_{m3}\begin{bmatrix}\boldsymbol{x}\\1\end{bmatrix}$ is constant from $\boldsymbol{a}_3 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$,

$$s = \frac{m}{w}. \tag{11}$$

From the observed x and w, $\boldsymbol{X}_c$ can be computed as follows.

$$\boldsymbol{X}_c = \frac{m}{w}\boldsymbol{A}^{-1}\begin{bmatrix}\boldsymbol{x}\\1\end{bmatrix} = m\boldsymbol{X}_f \tag{12}$$

When the pair of $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$、 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$、 $w_1, \ldots, w_n$ is given, $\boldsymbol{X}_{f_1}, \ldots, \boldsymbol{X}_{f_n}$ can be calculated from $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$、 $w_1, \ldots, w_n$ using Equation (12)

$$\begin{aligned} m\boldsymbol{X}_{f_1} &= \boldsymbol{R}\boldsymbol{X}_1 + \boldsymbol{t}\\ &\cdots\\ m\boldsymbol{X}_{f_n} &= \boldsymbol{R}\boldsymbol{X}_n + \boldsymbol{t}. && (13)\end{aligned}$$

Therefore,

$$m\bar{\boldsymbol{X}}_f = \boldsymbol{R}\bar{\boldsymbol{X}} + \boldsymbol{t}. \tag{14}$$

(but $\bar{\boldsymbol{X}}_f = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_{f_i}$ $\bar{\boldsymbol{X}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i$.) From Equations (13) and (14),

$$\begin{aligned} m\left(\boldsymbol{X}_{f_1} - \bar{\boldsymbol{X}}_f\right) &= \boldsymbol{R}\left(\boldsymbol{X}_1 - \bar{\boldsymbol{X}}\right)\\ &\cdots\\ m\left(\boldsymbol{X}_{f_n} - \bar{\boldsymbol{X}}_f\right) &= \boldsymbol{R}\left(\boldsymbol{X}_n - \bar{\boldsymbol{X}}\right). && (15)\end{aligned}$$

A rotation matrix that satisfies Equation (15) is calculated as $\boldsymbol{R}$, which minimizes the following equation using singular value decomposition.

$$\sum_{i=1}^{n}\left\|\frac{\left(\boldsymbol{X}_{c_i} - \bar{\boldsymbol{X}}_c\right)}{\left\|\boldsymbol{X}_{c_i} - \bar{\boldsymbol{X}}_c\right\|} - \boldsymbol{R}\frac{\left(\boldsymbol{X}_i - \bar{\boldsymbol{X}}\right)}{\left\|\boldsymbol{X}_i - \bar{\boldsymbol{X}}\right\|}\right\|^2 \to \min \tag{16}$$

Then, m is calculated from Equation (15) using a least square method,

$$m = (\boldsymbol{X}_F'\boldsymbol{X}_F)^{-1}\boldsymbol{X}_F'\boldsymbol{R}\begin{bmatrix}\boldsymbol{X}_1 - \bar{\boldsymbol{X}}\\\cdots\\\boldsymbol{X}_n - \bar{\boldsymbol{X}}\end{bmatrix} \tag{17}$$

Here,

$$\boldsymbol{X}_F = \begin{bmatrix}\boldsymbol{X}_{f_1} - \bar{\boldsymbol{X}}_f\\\cdots\\\boldsymbol{X}_{f_n} - \bar{\boldsymbol{X}}_f\end{bmatrix}. \tag{18}$$

$\boldsymbol{t}$ is determined by subsututiting m, $\boldsymbol{R}$ in Equation (14).

3.2. Error Estimation

By using the above algorithm, the rotation matrix $\boldsymbol{R}$ and position $\boldsymbol{t}$ of each newly added camera can be determined. We have to evaluate the accuracy of $\boldsymbol{R}$ and $\boldsymbol{t}$ and have to obtain the variance σ_e of the error e of Equation (2) to compose the new observation node with these camera. From Equation (3), the variance σ_e can be calculated with the variations $\Delta\boldsymbol{R}$ and $\Delta\boldsymbol{t}$.

$$\begin{aligned} \sigma_e^2 &= \boldsymbol{b}(\boldsymbol{A}E[\Delta\boldsymbol{R}\boldsymbol{X}\boldsymbol{X}'\Delta\boldsymbol{R}']\boldsymbol{A}'\\ &\quad + \boldsymbol{A}E[\Delta\boldsymbol{t}\Delta\boldsymbol{t}']\boldsymbol{A}')\boldsymbol{b}'L_{p_j,t,i} && (19)\\ &\simeq \boldsymbol{b}(\boldsymbol{A}E[\Delta\boldsymbol{R}\boldsymbol{X}\boldsymbol{X}\Delta\boldsymbol{R}']\boldsymbol{A}'\\ &\quad + \boldsymbol{A}E[\Delta\boldsymbol{t}\Delta\boldsymbol{t}']\boldsymbol{A}')\boldsymbol{b}'\bar{L}_{p_j,t,i} && (20)\end{aligned}$$

Here,

$$\boldsymbol{b} = [1\ 0\ 0]' \tag{21}$$

Since $L_{p_j,t,i}$ the distance between camera $\boldsymbol{C}_i$ and human $\boldsymbol{X}_{p_j,t}$ is unknown, it is approximated by $\bar{L}_{p_j,t,i}$ calculated using predicted position $\bar{\boldsymbol{X}}_{p_j,t}$.

$E[\Delta\boldsymbol{t}\Delta\boldsymbol{t}']$ can be calculated as follows.

$$\begin{aligned} E[\Delta t\Delta t'] &= m^2E[\Delta\boldsymbol{R}\bar{\boldsymbol{X}}_f\bar{\boldsymbol{X}}_f'\Delta\boldsymbol{R}']\\ &\quad + \sigma_m^2\boldsymbol{R}^{-1}\bar{\boldsymbol{X}}_f\bar{\boldsymbol{X}}_f(\boldsymbol{R}^{-1})' && (22)\end{aligned}$$

Here, σ_m^2 is the variance of m based on Equation (17).

The error variance of rotation matrix $\boldsymbol{R}$ is calculated by an algorithm given in literature [14, 15], and the variable of the above equation is determined.

3.3. Implementation

We applied the proposed camera calibration algorithm to our human tracking system, as follows. When a human region was obtained at an observation node having an uncalibrated camera, the observation node sent the observation time to the tracking node, and obtained the 3D head and foot positions and shirt color of the target. The shirt color was also detected in the observed image at the observation node locally. Then, a similarity evaluation was performed between the tracking model and the image feature using the color information. Because it satisfied a threshold, that observation was used for the camera calibration.

The camera calibration algorithm described in the previous section was then performed with the 2D information as well as the 3D position obtained from the tracking node, and the calibration error was estimated. For stable tracking purposes, the fixed number of observations at the newly installed camera was initially used for calibration only. After that period, the newly installed cameras will joined the human tracking process automatically.

The process flow is shown in Figure 7.

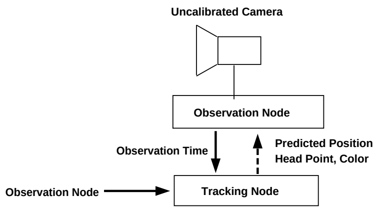
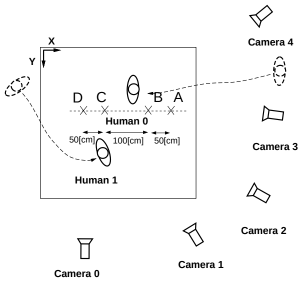


Figure 7. Process Flow

4. Experiments

To confirm the availability of this algorithm, the following experiments were performed.

Five cameras (cameras 0-4) were arranged as shown in Figure 8. Each camera was connected to a PC (Pentium II 400MHz). The camera and PC composed an observation node. All input images were processed at the observation nodes. The processing speed was about 5-6 frames/sec.



Top View

Figure 8. Camera Arrangement

Each PC was connected to the same network and the internal clock was synchronized by NTP (Network Time Protocol). The discovering node and tracking node were also connected to the network.

Each camera except cameras 3,4 had already been calibrated, and each observation node sent camera parameters and observed information (observation time, position, and color of a feature point) to the discovering node and tracking node. In this experiment, a human walked in the scene, and the human was tracked using cameras 0-2. The observation nodes having uncalibrated cameras 3,4 acquired estimated 3D positions of the head and foot of the target from the tracking node, and then calibrated their cameras using this information as well as 2D features (2D head position, 2D foot position, and height of the human region) obtained from their own images. Figure 9 shows the distribution of 3D head positions and 3D foot positions used in the experiment.

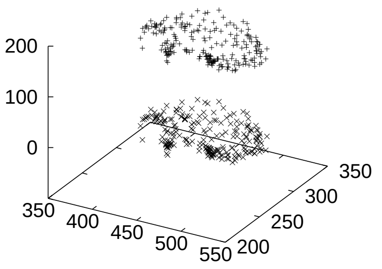


Figure 9. Sample Points (Head Positions and Foot Positions)

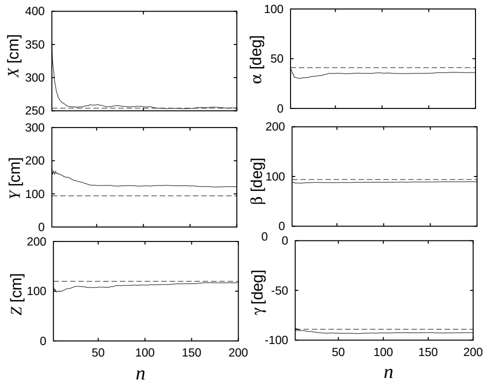


Figure 10. Calibration Results

Figure 10 shows estimated results of the camera parameters for camera 3 (the horizontal axis shows the observation count, and the dashed line denotes the parameter value calculated by hand. In this example, this camera joined the tracking after 60 observations). Figure 11 shows estimated errors of camera parameters. As can be seen, the estimation errors decrease with increasing number of observations.

Next, we measured changes in the accuracy of human tracking before and after cameras were added. In this experiment, first, three calibrated cameras tracked human motions, and then two new cameras were added to the system. The latter cameras were automatically calibrated with our algorithm. We compared the tracking accuracy between the

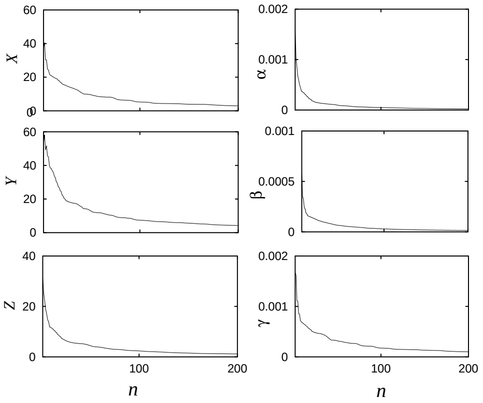


Figure 11. Calibration Errors

Table 1. Human Tracking Accuracy(cm^2)

	Point A	Point B	Point C	Point D
3 Cameras	NA	NA	74.837	82.168
5 Cameras	157.41	189.51	22.325	42.491

two cases. We measured the tracking accuracy at four fixed positions as shown in Figure 8.

Here, the estimated 3D positions of a person were recorded for 10 seconds, and the variances (trace of the covariance matrix) of the estimated 3D positions were compared. The results are shown in Table 1. The tracking results at points A and B were not applicable in the three-camera case because the bounds of the tracking area were exceeded. With the addition of the two cameras, these points were included in the tracking area. In addition, as can been seen, the tracking accuracy was improved at points C and D with the new cameras. This suggests that we can easily expand the tracking area and also can improve the tracking accuracy. Using this technique, multiple camera based human tracking systems can be installed in wider area.

5. Conclusion

This paper describes an algorithm for automatic camera calibration. The rotations and the positions of newly installed cameras can be automatically calibrated from human tracking results using non-synchronous multiple observations. The newly installed cameras can join the tracking process automatically according to the estimated calibration errors. This algorithm can be used to expand the tracking area and also increase the tracking accuracy. Our technique reduces the cost of changing camera arrangement. We consider that it can facilitate the construction of a large-scale human tracking system with many cameras.

Future work includes improving the feature extraction and viewpoint selection method to obtain highly accurate results, the automatic determination of the world coordinates in a multiple camera system, and autonomous recovery from sudden changes (errors) in a camera's position/orientation.

References

[1] Hiroki Mori, Akira Utsumi, Jun Ohya, and Masahiko Yachida. Human tracking system using adaptive camera selection. In *Proc. of RO-MAN '98*, pages 494–499, 1998.

[2] Akira Utsumi, Hiroki Mori, Jun Ohya, and Masahiko Yachida. Multiple-view-based tracking of multiple humans. In *Proc. of ICPR'98*, pages 597–601, 1998.

[3] D. M. Gavrila and L. S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *Proc. of CVPR*, pages 73–80, 1996.

[4] Ali Azarbayejani and Alex Pentland. Real-time self-calibrating stereo person tracking using 3-d shape estimation from blob features. In *13th International Conference on Pattern Recognition*, pages 627–632, 1996.

[5] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *SPIE proceeding vol. 2615*, pages 89–98, 1996.

[6] M. Patrick Johnson, P. Maes, and T. Darrell. Evolving visual routines. In *Proc. of Artificial Life IV*, pages 198–209, 1994.

[7] Jakub Segen and Sarma Pingali. A camera-based system for tracking people in real time. In *Proc. of 13th International Conference on Pattern Recognition*, pages 63–67, 1996.

[8] Q. Cai, A. Mitiche, and J. K. Aggarwal. Tracking human motion in an indoor environment. In *Proc. of 2nd International Conference on Image Processing*, pages 215–218, 1995.

[9] Q. Cai and J. K. Aggarwal. Tracking human motion using multiple cameras. In *Proc. of 13th International Conference on Pattern Recognition*, pages 68–72, 1996.

[10] Akira Utsumi and Jun Ohya. Multiple-camera-based human tracking using non-synchronous observations. In *Proc. of ACCV2000*, pages 1034–1039, 2000.

[11] L.Lee, R.Romano, and G.Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Pattern Anal. Machine Intell.*, 22(8):758–767, 2000.

[12] Hirotake Yamazoe, Akira Utsumi, Nobuji Tetsutani, and Masahiko Yachida. Automatic camera calibration method for multiple camera based human tracking system. In *Proc. of IWAIT 2001*, pages 77–82, 2001.

[13] Akira Utsumi and Jun Ohya. Image segmentation for human tracking using sequential–image–based hierarchical adaptation. In *Proc. of CVPR '98*, pages 911–916, 1998.

[14] M.D.Shuster and S.D.Oh. Three-axis attitude determination from vector observations. *J.Guidance and Control*, 4(1):70–77, 1981.

[15] F.Landis Markley. Attitude determination using vector observations and the singular value decomposition. *the Journal of the Astronautical Science*, 36:245–258, 1988.