Intelligent Visual Sensor Networks for Person Tracking

Toshiki Iso and Shoji Kurakake
Network Laboratories
NTT DoCoMo, Inc.
Kanagawa, Japan, 239-8536
Email: {iso, kurakake}@netlab.nttdocomo.co.jp

Abstract—**We propose a method for autonomously capturing the best image of people in real time from just sensor data. The method can automatically detect the positions and directions of people and frontal images of them in real-time. it offers excellent performance because it allows for variation in the disparity information collected from a number of cameras with fisheye lenses and uses new criteria that consider not only the relative position between cameras and people, but also the optical properties of the lenses. We describe the method's algorithm, and show that our method is effective by testing a prototype system.**

I. Introduction

The recent generation of sensors offer powerful computing and communication abilities. They are also becoming so small and cheap that it is now possible to use them as ubiquitous sensors. We emphasize the importance of visual sensors because vision provides the most detailed information possible. Surveillance and tel-monitoring applications are especially popular among the applications based on visual sensors. For example, a visual sensor network that can automatically match what the user wants to see with the user's current location allows us to watch our house from remote sites or to receive views of sightseeing spots from another direction without explicit user interaction. Moreover, if our house has a variety of sensors, we can visually communicate with a friend while walking around the house. (We call it the "Hands free video phone" (Fig.1).) In order to realize these systems, we built an intelligent visual sensor network for automatic visual tracking. It is simpler than conventional methods [1], [2], [3], [4]. In this paper, we will describe the algorithm and show the results of experiments conducted on a prototype system.

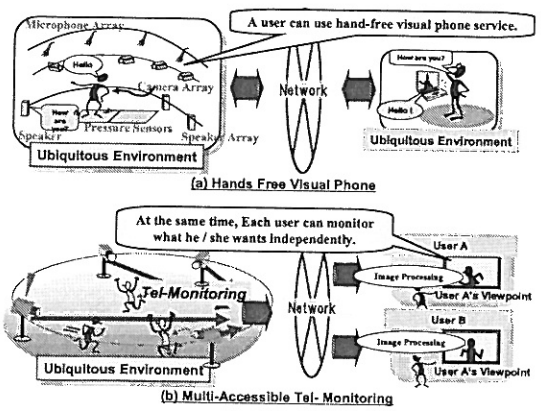

Fig. 1. Ubiquitous applications using person tracking

II. Cameras as Visual Sensors

A. Fisheye Lens

In order to track people walking around freely, many visual sensors are necessary. Since the data must be processed in real-time, we should use methods that are as simple as possible. Our sensor network uses cameras with fisheye lenses($f\theta$ lens) as the visual sensors. This is because such cameras can capture the widest view and minimize user input and access contention. For example, if a normal lens is used, real-time control of camera direction is needed to capture the desired view; only one user can drive the camera at a time so the other user must wait for their turn. An omni-directional lens has a dead spot created by the mirror, so it is impossible to use the whole image.

While the image captured by a fisheye lens seems distorted, simple image processing can recover normal-looking images from arbitrary views. As a result, the use of fisheye lenses allows a person tracking system to be implemented with fewer cameras than is possible with other lenses.

B. Plane Image Transform

To track a person, we need to transform the fisheye image into a plane image by applying the following formula;

$$(x_{image}, y_{image}) = (f \sin\theta \cos\phi, f \sin\theta \sin\phi) \quad (1)$$

$$\begin{cases} \cos\theta = \dfrac{Z}{\sqrt{X^2+Y^2+z^2}} & \cos\phi = \dfrac{X}{\sqrt{X^2+Y^2}} \\ \sin\theta = \dfrac{\sqrt{X^2+Y^2}}{\sqrt{X^2+Y^2+z^2}} & \sin\phi = \dfrac{Y}{\sqrt{X^2+Y^2}} \end{cases} \quad (2)$$

where $q(X, Y, Z)$ and $p(x_{image}, y_{image})$ are the same point in world coordinates and image plane coordinates, respectively, θ and ϕ are incident angles, and f is the focal distance of the fisheye lens2. This formula allows us to generate an image from an arbitrary view simply by specifying the position or incident angle desired.

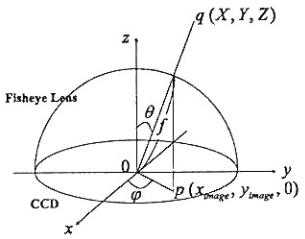

Fig. 2. Fisheye lens

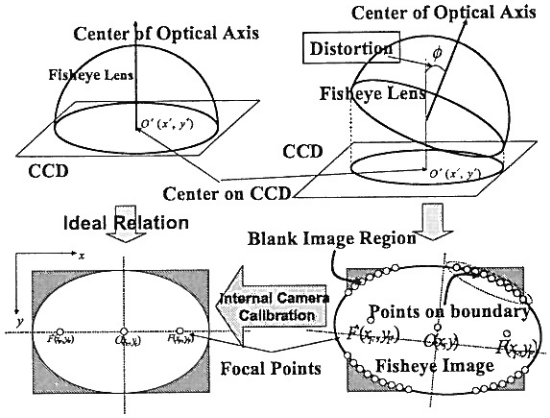

Fig. 3. Inner camera calibration for fisheye lens

C. Fisheye Lens Calibration

To track a person across a wide area we need to integrate several cameras into a sensor network. This demands camera calibration, but no method has been published for the calibration of cameras with fisheye lenses.

We propose two simple calibration methods. At first, our method for internal camera calibration finds the true optical axis by calculating the ellipse equation of the boundary on the effective image plane. It is based on the assumption that the refractive index distribution exhibits point symmetry. In practice, after detecting feature points on the boundary of the effective image plane by the SUSAN edge detector [5], we use them to calculate the two focus points F and F_i of the ellipse equation (Fig. 3); the center point between them is taken as the true position of the optical axis.

Our method for external camera calibration corrects the position of each camera by adjusting standard markers that are located accurately on a flat board.

III. Sensing Person Movement

This section describes how to detect a person's movements. Our method basically uses changes in disparity [6]. Disparity is the difference between pairs of stereo images and represents depth in space; changes in disparity are changes in depth in space, and can indicate that some object has entered the field of view.

Background subtraction and frame subtraction are commonly used to detect people in camera images because of their simplicity. Unfortunately, they are weak to sudden changes in illumination. Moreover, detection can fail if the person is stationary for more than a few minutes. On the other hand, as our method uses disparity differences to identify the existence of people, it ignores sudden changes in illumination and stationary users can still be detected.

By applying the method to all cameras in a sensor network, we can obtain sequences of changes in disparity. These sequences can be analyzed to extract spatial-temporal information from which we can detect the movement of people. Our algorithm that realizes this is described below.

A. Estimating person region by two criteria based on spatial-temporal changes in disparity

We assume cameras C_i and C_j share an area (Fig. 4) and both have been calibrated. By projective transform, the relation of pixel $P_{C_i}(x_{C_i}, y_{C_i})$ to $P_{C_j}(x_{C_j}, y_{C_j})$ on each image plane can be described using equation(3). (where $H_{C_iC_j}$ is the projection transform matrix between C_i and C_j and λ is a coefficient.)

$$\lambda \begin{pmatrix} x_{C_i} \\ y_{C_i} \\ 1 \end{pmatrix} = H_{C_iC_j} \begin{pmatrix} x_{C_j} \\ y_{C_j} \\ 1 \end{pmatrix} \tag{3}$$

If the main surface in the common-view area is flat, the cameras' pixel values are the same. If the surface is uneven, their pixel values are different because disparity is generated by the unevenness. Therefore, when a person enters the common-view area, the disparity changes drastically. This characteristic is used to detect people.

$$DM^{(i,j)}(x,y,t) = \begin{cases} 1 & \text{if } |N_{Ci}(x,y,t) - N_{Cj}(x,y,t)| \geq threshold \\ 0 & otherwise \end{cases} \tag{4}$$

Let $DM^{(i,j)}(x,y,t)$ denote the disparity map at (x,y) on C_i image at time t where $N_{Ci}(x,y,t)$ and $N_{Cj}(x,y,t)$ are normalized to the same size by linear interpolation.

We can find blobs $B_{C_i}(t)$, which are extracted by label processing on $DM^{(i,j)}(x,y,t)$, as uneven surface regions. This means that each surface shape has a different degree of disparity. Therefore, a change in disparity means a change in surface shape. Based on the change in disparity, we can identify the regions in which people exist. In detecting the change in disparity, we use two criteria: temporal change $\Delta DM^{(i,j)}_{tem}(x,y,t)$ from the difference between disparity across frames and spatial change $\Delta DM^{(i,j)}_{spa}(x,y,t)$ from the difference between the initial disparity and the disparity at time t.

$$\Delta DM^{(i,j)}_{tem}(x,y,t) = DM^{(i,j)}(x,y,t) - DM^{(i,j)}(x,y,t-1) \tag{5}$$

$$\Delta DM^{(i,j)}_{spa}(x,y,t) = DM^{(i,j)}(x,y,t) - DM^{(i,j)}_{init}(x,y) \tag{6}$$

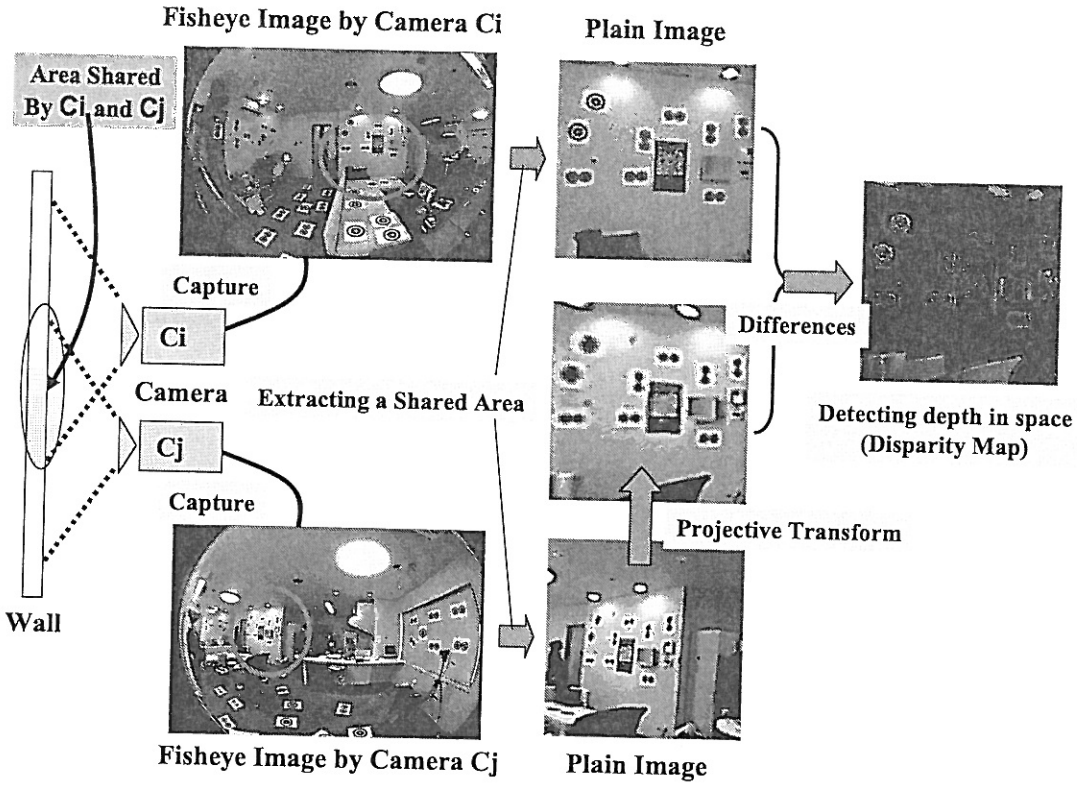

Fig. 4. Detecting depth in space by disparity

The former is robust to illumination changes, but becomes weak if the person is stationary for some time. In this situation, the latter criterion is effective since the disparity from the initial condition is based on statistical data collected under a variety of illumination conditions. Combining these two criteria increases the reliability of the judgment of person existence.

Candidate regions (people) are given by the boundary lines, which are externally tangent to blobs $B_{C_i}(t)$ extracted by the above criteria.

B. Analyzing region and movement direction

At first, we describe a method that can detect regions containing people. As described above, a pair of cameras can detect candidate regions. However, the detected candidate regions have an uncertainty because the disparity contains the two differences between each of the images. In order to sharpen the region considered, we need more information of disparity from other views. Then, in the similar way of thinking, another cameras can be selected as pairs of cameras and used to detect each of the candidate area of people from the boundary lines which are externally tangent to blobs $\overrightarrow{B_{C_i}}(t)$. After detecting the overlapping region in each of the candidate areas of people, we can obtain the blob region $\vec{B}(t)$ indicating that people exist.

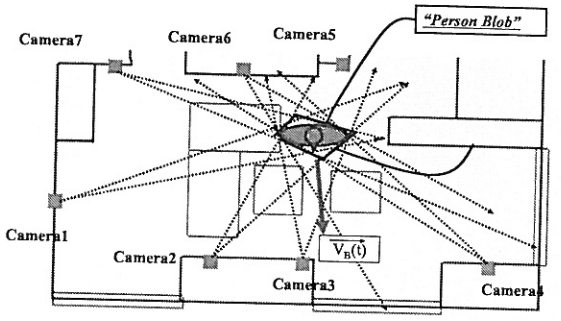

Fig. 5. Detecting a candidate region of a person

Next, we describe a method that can detect the direction of person movement. Basically, the method tracks the the center

of gravity of the person's region. Since the tracked positions are not always stable, we propose the following method which detects movement vector $\overrightarrow{v_B}(t)$ by using two criteria, equation (8) and (9) as follows;

$$\overrightarrow{v_B}(t) = w(t)(\overrightarrow{B}(t) - \overrightarrow{B}(t-1)) \quad (7)$$

if equation (8) and (9) are satisfied as follows;

$$E_{vec}(t) = \frac{\overrightarrow{v_B}(t) \cdot \overrightarrow{v_{B_{ave}}}(t)}{|\overrightarrow{v_B}(t)||\overrightarrow{v_{B_{ave}}}(t)|} < threshold \quad (8)$$

$$and$$

$$E_{pos}(t) = |\overrightarrow{B}(t) - \overrightarrow{B}(t-1)| < threshold \quad (9)$$

where

$$w(t) = 1 - \frac{|\overrightarrow{v_B(t)} - \overrightarrow{v_{B_{ave}}(t-1)}|}{|\overrightarrow{v_B(t)} + \overrightarrow{v_{B_{ave}}(t-1)}|} \quad (10)$$

$$\overrightarrow{v_{B_{ave}}}(t) = \frac{1}{N}\sum_{n=1}^{N} \overrightarrow{v_B}(t - \Delta t_n) \quad (11)$$

$$\overrightarrow{B}(t) = (x_g(t), y_g(t)) \quad (12)$$

On condition that the person performs only natural motion, we define $w(t)$ and $E_{pos}(t)$ as the weight that prevents the movement vector from undergoing extreme changes at $t-1$ just before t. $E_{vec}(t)$ prevents the movement vector from undergoing extreme changes over the time range from $t - \Delta t_N$ to now $t - \Delta t_1$ (fig. 6). $\overrightarrow{v_{B_{ave}}}(t)$ represents the average velocities in the very short time period from $t - \Delta t_N$ to $t - \Delta t_1$. By using these criteria, we can obtain the stable person movement vector $\overrightarrow{v_B}(t)$.

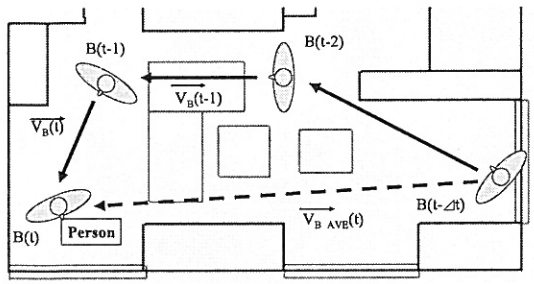

Fig. 6. Detecting the movement vector of person's blob

IV. Capturing the best shot image of person

A. Criteria for selecting the optimal cameras

In order to track people for applications such as tel-monitoring, it is necessary to capture them from the optimal camera angle. Usually, the best camera angle is that which

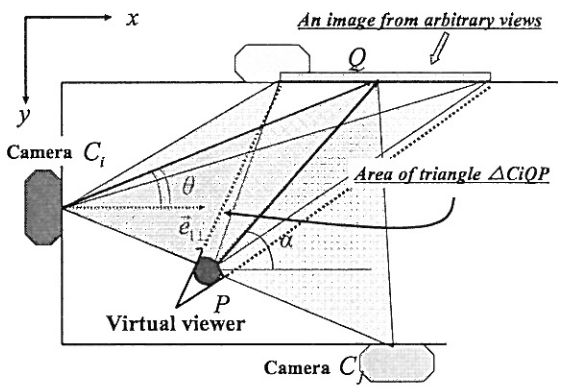

Fig. 7. Criteria for selecting the optimal cameras

captures a frontal view of them. Traditional criteria for normal lenses use similarities between view angle and camera angle or distances between the person's position and camera position. Unfortunately, they are not suitable for fisheye lenses. This is because the resolution distribution varies widely on the image plane of the fisheye lens. In other words, its resolution depends on the angular deviation from the optical axis. If the deviation becomes large, the resolution becomes worse, so we need to consider the optical properties of the lens in selecting the best criteria. We define below new criteria which can be used for a variety of cameras.

$$E_{cam}(i) = \int_{\theta_s}^{\theta_e}\int_{\phi_s}^{\phi_e} \{\frac{1}{w(\theta,\phi)} \cdot \frac{1}{2}|\overrightarrow{C_iQ}| \cdot |\overrightarrow{PQ}| \cos(\theta - \alpha) d\theta d\phi\} \quad (13)$$

where $w(\theta, \phi)$ is defined as follows;

$$w(\theta,\phi) = \{\frac{\partial R(\theta,\phi)}{\partial \theta}\}^2 + \{\frac{\partial R(\theta,\phi)}{\partial \phi}\}^2 \quad (14)$$

C_i, Q, and P represent position of camera i, center of view points, and virtual viewer, respectively.

The element of integral of equation (13) represents the area of triangle C_iQP. The best camera is the one that minimizes the area of triangle C_iQP. Moreover, as we use cameras with fisheye lens in this system, it is necessary to consider optical properties for better image resolution. $R(\theta, \phi)$ represents the optical resolution distribution of the image captured by the fisheye lens, θ and ϕ are the camera angles expressed as rotations around the Z-axis and X-axis, respectively , α and β are the view angles expressed as rotations around the Z-axis and X-axis, respectively.

For a fisheye lens, $R(\theta, \phi)$ can be define as follows;

$$R(\theta,\phi) = f \sin\theta \quad (15)$$

Therefore, we obtain the following criterion;

$$w(\theta, \phi) = \{f \cos\theta\}^2. \quad (16)$$

With a normal lens, the optical resolution distribution is linear to incidence angle θ, so $R(\theta, \phi)$ is nearly constant.

When equation (13) is minimum, we can obtain the optimal camera C_i.

$$i = \arg\max_i \{E_{cam}(i)\} \quad (17)$$

The above criteria allows us to select not only the optimal cameras, but also to capture better quality images from cameras with fisheye lenses (fig. 7).

B. Generating images from arbitrary views

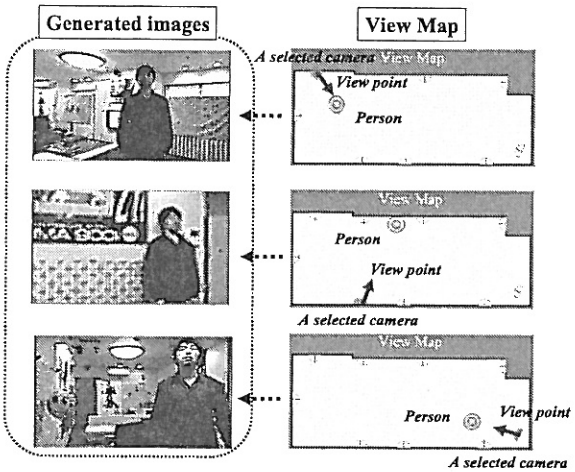

Fig. 8. Generating images of a person from arbitrary views

For surveillance or hands free visual phones, it is necessary to generate images from arbitrary views in real-time. Several conventional methods that provide real-time performance exist such as IBR [7]. Unfortunately, they are basically suitable for normal cameras, not fisheye cameras. This is because they require perfect external camera calibration for combining different camera images; moreover, the image resolution of cameras with fisheye lenses is not uniform. We achieve real-time processing by applying affine transform; it is a simple method that can generate images from arbitrary views. Calculating the affine transform parameters requires the relative position between the optimal camera and the angle of the arbitrary view. As we can select the optimal camera images using the criteria describe above, we can easily generate images from arbitrary views. Typical examples of arbitrary views are shown in Fig. 8.

V. Experimental Prototype

We constructed an experimental intelligent visual sensor network. Many sensors were set up in a room of a house and connected to each other via personal computers over a local area network. Fig. 9 shows the system's processing flow.

Experimental conditions are described below.

- Visual sensors
  Seven CCD cameras($720[pixel] \times 480[pixel]$) with Fisheye lenses($f \sin\theta$)
- Sensor network's computing resource
  Ten Personal Computers(Desktop type, CPU:3.2$[GHz]$, RAM1$[GB]$)
- Viewer's Computing resource
  Three Personal Computers(Notebook type, CPU:1$[GHz]$, RAM1$[GB]$)

Figures 10 show the best images of a person moving in the sensor network's cover area. Its tracking speed is $8[fps]$.

VI. Conclusion

We proposed an intelligent visual sensor network for autonomously tracking people using only image processing techniques. It offers robustness with regard to illumination changes and does not need any special equipment. We constructed a prototype system based on our algorithm, and described the potential of ubiquitous applications such as tel-monitoring and hands free visual phones.

We plan on developing a method of external camera calibration for fisheye lenses in order to improve the accuracy our algorithm for detecting people and the quality of the generated images. We will apply the method to realize tel-monitoring and hands free visual phone services.

Acknowledgment

The authors would like to thank Dr. H. Nakano, NTT DoCoMo Multimedia Labs, and Mr. K. Imai, NTT DoCoMo Network Labs for their encouragement and our colleague for many discussions and their support.

References

[1] R. T. Collins et al., *A System for Video Surveillance and Monitoring*, CMU VSAM Report, CMU-RI-TR-00-12,2000.

[2] T. Kanade et al., *Cooperative multisensor video surveillance*, Proceedings of the 1997 DARPA Image Understanding Workshop, volume 1, pp. 3-10, May 1997.

[3] N. Ukita and T. Matsuyama, *Real-time Multi-target Tracking by Cooperative Distributed Active Vision Agents*, AAMAS'02, 2002.

[4] T. Matsuyama, *Cooperative Distributed Vision: Dynamic Integration of Visual Perception, Action, and Communication*, Proc. of 23rd Annual German Conference on Artificial Intelligence (Lecture Notes in Artificial Intelligence 1701, Springer), pp.75-88, 1999.

[5] S. M. Smith, J. M. Brady Jia-Xiong• • SUSAN - A New Approarch to Low evel Image Processing• • •Technical Report(http://www.fmrib.ox.ac.uk/ steve/susan/), 1995.

[6] M. Kiyama, N. Ohta, K. Kanatani, *Detecting Entering Persons using Two Cameras and Homography*, IPSJ Technical Report, 99-CVIM-118-8, pp.53-58. 1999.

[7] S. M. Seitz and C. R. Dyer, *View Morphing*, Proc. of SIGGRAPH'96, pp. 21-30, 1996.

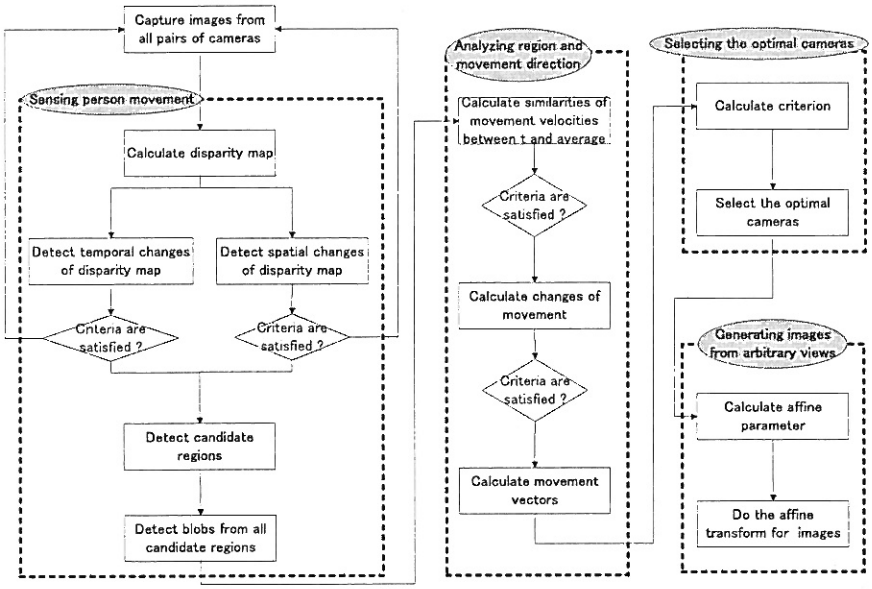

Fig. 9. Processing flow in prototype

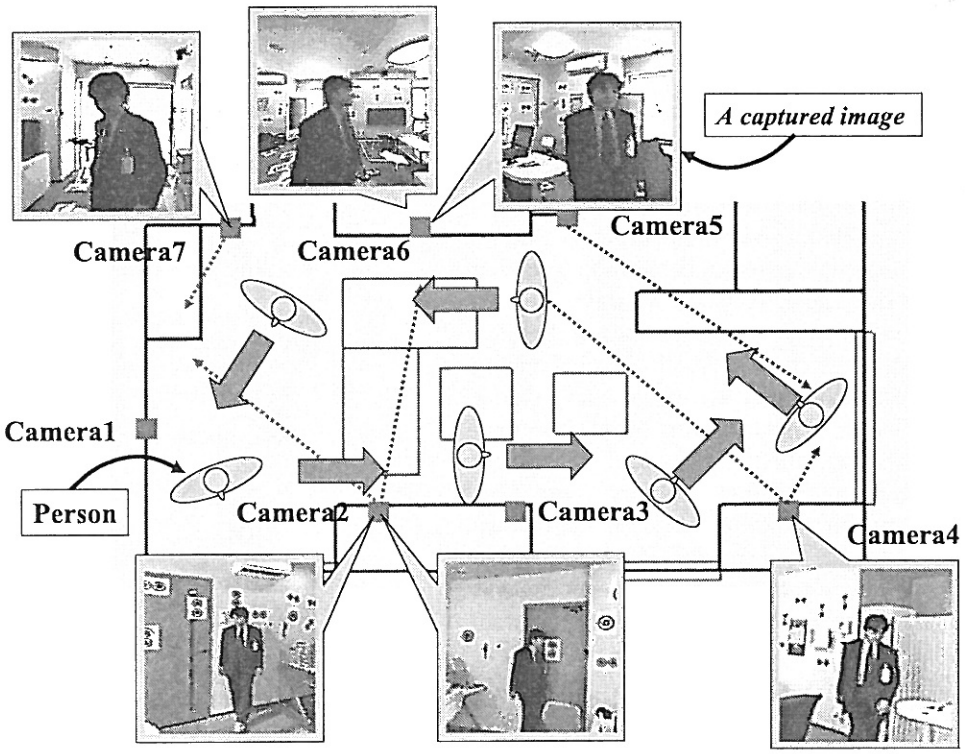

Fig. 10. Autonomous capture of the best image