Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation
July 16-20, 2003, Kobe, Japan

Networked Video Surveillance Using Multiple Omnidirectional Cameras

Shinji Morita
Graduate School of Information Science
Nara Institute of Science and Technology

Kazumasa Yamazawa
Graduate School of Information Science
Nara Institute of Science and Technology
yokoya.aist-nara.ac.jp/~yamazawa/

Naokazu Yokoya
Graduate School of Information Science
Nara Institute of Science and Technology
yokoya.aist-nara.ac.jp/~yokoya/

Abstract
Remote surveillance is widely utilized at bank, shop, office, home and so on. In most conventional remote surveillance systems, fixed or active cameras with a narrow field of view are generally used in order to acquire an image of the remote site. This paper proposes new networked surveillance systems. The proposed surveillance system, which uses multiple omnidirectional cameras and network, is based on a server/client model: the server computers, each of which is connected to an omnidirectional video camera, are placed in the surveillance area and the client computer is placed in a user side. The servers detect moving objects and estimate their directions from sensors. The client estimates object positions from their directions received from the servers and presents object-centered perspective images to the user. In experiments, the implemented proposed system can do those at real-time.

1 Introduction

Remote surveillance is widely utilized at bank, shop, office, home and so on. Although most of the methods use a PTZ (panning, tilting and zooming) camera[1] or multiple cameras[2]. An omnidirectional camera[3,4] has a number of advantages. For example, if we use background subtraction for detecting moving objects, with a PTZ camera, we must synchronize the PTZ of the camera and the PTZ of the background image. When we use multiple cameras, the system becomes complex. On the other hand, an omnidirectional camera, which can acquire an omnidirectional view at a time, is suitable for detecting moving objects in the surrounding environment. However, as for most of conventional methods only one computer manages all omnidirectional cameras. Moreover it suffers from oscillations of intensity, such as fluorescent lights.

This paper describes networked video surveillance using multiple omnidirectional cameras and robustly detecting moving objects with a background subtraction technique based on adaptively updating background images. Our system is server/client model. Each server has an omnidirectional camera, detects moving objects, and estimates the directions of the objects. The client receives the directions and the omnidirectional video from the servers through network, estimates the positions of the objects, and shows the user the positions and the images of the objects.

2 Omnidirectional Camera: HyperOmni Vision

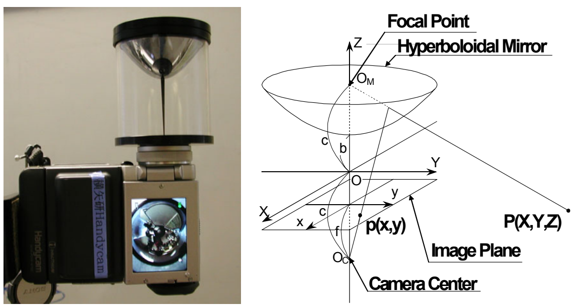
We use HyperOmni Vision[5] as an omnidirectional camera in the present work. HyperOmni Vision is composed of a hyperboloidal mirror and a video camera as illustrated in Fig.1. The camera acquires an omnidirectional scene reflected by the hyperboloidal mirror. The hyperboloidal mirror is constructed of a hyperboloid of two sheets of revolution, which has two focal points (O_M and O_C). The camera lens center is fixed at the focal point O_C. Given a world coordinate (X, Y, Z) and an image coordinate (x, y) as shown in the Fig.1(b), the shape of hyperboloidal mirror and the two focal points are represented as follows.

$$\begin{aligned}&\text{Mirror Shape}: \frac{X^2+Y^2}{a^2}-\frac{Z^2}{b^2}=-1,\\ &\text{Inner focal point } O_M: (0,0,+c), \qquad (1)\\ &\text{Outer focal point } O_C \text{ (Camera lens center)}: (0,0,-c),\\ &\text{where } c=\sqrt{a^2+b^2}.\end{aligned}$$

A ray going from the point $P(X, Y, Z)$ in 3D toward the inner focal point O_M is reflected by the mirror and passes through the outer focal point O_C intersecting the image plane at the point $p(x, y)$. Therefore, the projection of HyperOmni Vision is not common planar perspective, but satisfies the single viewpoint constraint. The relationship between $P(X, Y, Z)$ and $p(x, y)$ is given by:

$$x = \frac{f\left(b^2 - c^2\right)X}{\left(b^2 + c^2\right)(Z - c) - 2bc\sqrt{X^2 + Y^2 + (Z - c)^2}},$$
$$y = \frac{f\left(b^2 - c^2\right)Y}{\left(b^2 + c^2\right)(Z - c) - 2bc\sqrt{X^2 + Y^2 + (Z - c)^2}}. \qquad (2)$$

By using the above equation, an omnidirectional input image can be converted to a common perspective image at the viewpoint O_M [6] (see Fig.2). In this study, we generate common perspective images of detected moving objects.



(a) Overview (b) Construction

Fig. 1 HyperOmni Vision

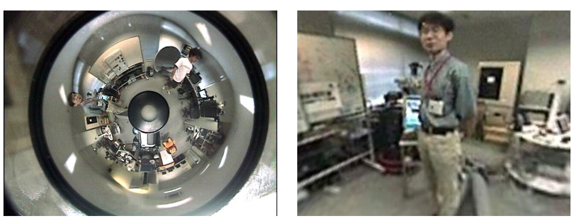


Fig. 2 Input image (left) and generated image (right)

3 Networked Surveillance System

3.1 Outline of the method

Our surveillance system is server/client model. The each server is at surveillance area and has an omnidirectional camera. The client is at the user side.

The process of each server is the following steps:

1. It acquires omnidirectional image at video-rate.
2. It detects moving objects and estimates the directions of the objects.
3. It sends the omnidirectional video and the directions to the client through network.

The process of client is the following steps.

1. It receives the video and the directions from the servers.
2. It estimates the positions of the objects.
3. It shows the positions and the images of the objects to the user.

If we have multiple servers and multiple clients, real-time surveillance is realized by distributing processing as the above processing.

3.2 Moving Object Detection

The proposed moving object detection method is based on background subtraction. The background subtraction is carried out at each pixel.

We model the intensity of background pixel as follows.

$$I = Iave + \sigma \sin(2\pi * \omega * t) + k * noise, \qquad (3)$$

where

I : intensity of background pixel,
$Iave$: average of I in time,
σ: amplitude of oscillation of intensity,
ω : frequency of oscillation of intensity,
t : time,
k : -1, 0, or +1,
$noise$: noise which is dependent only on the video camera.

The term $\sigma \sin(2\pi * \omega * t)$ in Eq.(3) means a flicker of a fluorescent light, CRT, etc. The term $k * noise$ means a noise which is dependent only on the video camera due to gain-up etc. Therefore, the intensity has the range of:

$$Iave - \sigma - noise \leqq I \leqq Iave + \sigma + noise \qquad (4)$$

If the intensity I of the pixel is outside the range, we decide that the pixel is a part of a moving object. First, we predetermine the parameter *noise* which is dependent only on the video camera setup. Next, we consider the slow change of the background. When a pixel is determined to represent the background, the parameters $Iave$ and σ are updated by the following equation.

$$I'ave = Iave * (n-1)/n + I * 1/n,$$
$$\sigma' = \sigma * (n-1)/n + 2 * (I - Iave) * 1/n, \qquad (5)$$

where

n : parameter of the update speed .
(When n increases, updating speed decreases.)

Eq. (5) is computed for all the pixels determined as the background on each frame. We use $I'ave$ and σ' as $Iave$ and σ in the next frame.

For a pixel estimated at a part of moving object, we do not calculate Eq. (5). However, in order to handle cases that an object is put or taken and the rapid change of background, when the pixel is estimated as a part of moving objects, the parameter σ is updated using the following equation.

$\sigma' = \sigma*(m-1)/m + 2*(I-Iave)*1/m,$ (6)

where

m: parameter for the fading out speed of the detected object. (When *m* increases, the speed of fading out decreases.)

Normally, we determine *m* larger than *n*. Therefore, if the detected moving object stays at a position, σ increases, and then the detected moving object fade out. If we set *m* infinity, the detected new objects do not fade out.

Next, we estimate the directions of the objects as follows.

1. Compute a histogram with respect to the longitude direction θ as illustrated in Fig.3.
2. Detect blobs whose values are more than 0 in the histogram as the ranges of the longitude of the moving objects.
3. Compute the maximum and minimum latitude for each blob as the direction of the object.
4. Send the directions of the objects to the client.

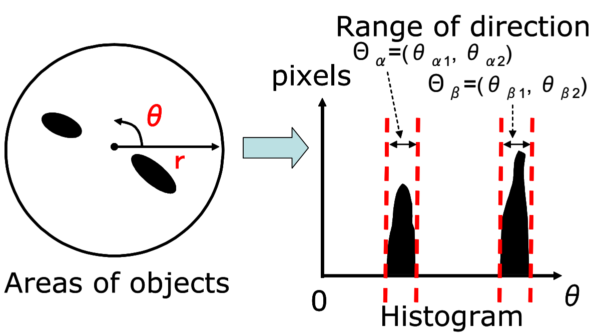


Fig. 3 Generating a histogram using polar-coordinates conversion

3.3 Omnidirectional Video Transport

Our system uses Digital Video Transport System (DVTS)[7] for transportation of omnidirectional video through the network. The DVTS is an application for transportation of IEEE1394 (FireWire) Digital Video stream through the network and has the following characteristics.

1. DVTS consumes about 33Mbps as network bandwidth to transport the high quality digital video stream (720x480 pixel, 30fps).
2. DVTS also transports auditory information synchronized with the visual scenes.
3. DVTS compress images to IEEE1394 Digital Video Stream and develops it with a DV recorder being the product for the consumer market.
4. DVTS can change the bandwidth adaptive for the end to end network condition.
5. Digital Video stream can be transported to anywhere through the network.

3.4 Object Position Estimation

The client receives the directions of the objects and estimates the positions of the objects as illustrated in Fig.4. We estimate the polygonal area where the directions overlap as the position of the objects. At last, a perspective image is computed from the received omnidirectional video and shown to the user. The computation of perspective images focusing on detected objects is important in some surveillance applications and can be carried out in real time by using the technique in [6].

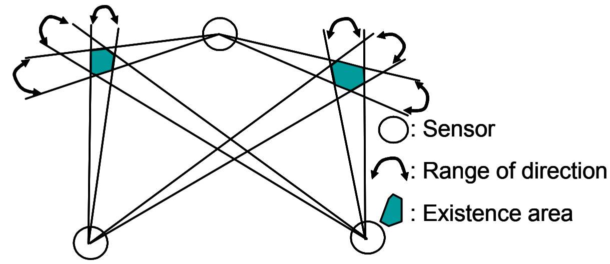


Fig. 4 Existence areas of focused objects

4 Experiment

We have prototyped a remote surveillance system using the proposed method. The system configuration and its hardware specification are illustrated in Fig.5 and Table.1. In the experiment, the surveillance area is room B308 [the second floor] of our institute, and the client is placed in room B211 [the first floor]. The surveillance area is 9.2m x 9.2m. Three omnidirectional cameras are placed at (0.0m, 0.0m), (4.5m, 0.0m), and (3.0m, 4.5m). The height of each omnidirectional camera from the floor is 1.7m. We set the parameters as follows.

$0 \le I \le 255$, *noise*=4, *n*=360, and *m*=1080.

The parameters are determined empirically. However, the selected parameters have been proven to be suitable for different indoor environments such as lobby and meeting rooms.

In this experiment, there were two walkers and a putted object under the condition that there were fluorescent lights and incoming lights from windows. The displayed image to the user is illustrated as Fig.6. Fig.7 illustrates the acquired omnidirectional images and the results of the detection on the each server. Fig.8 illustrates the perspective images which are shown to the users on the client. In the client side, the estimated position was refreshed at 0.2sec. The perspective image was refreshed at 0.05sec. The time from the acquisition of the omnidirectional images to the display of the perspective images is 1sec.

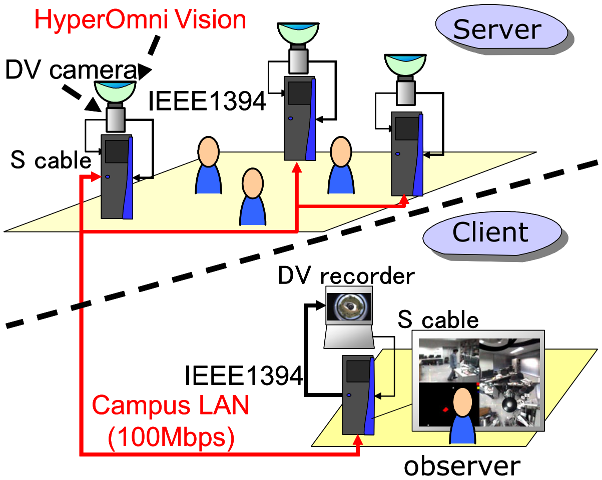


Fig. 5 Configuration of surveillance system

Table 1 Hardware configuration of surveillance system

Computer (Sever, Client)	CPU: Pentium4 2GHz Memory: 512MB OS: Redhat 7.2
Network	100Base-T
Image transport system	DVTS Ver.0.9α22

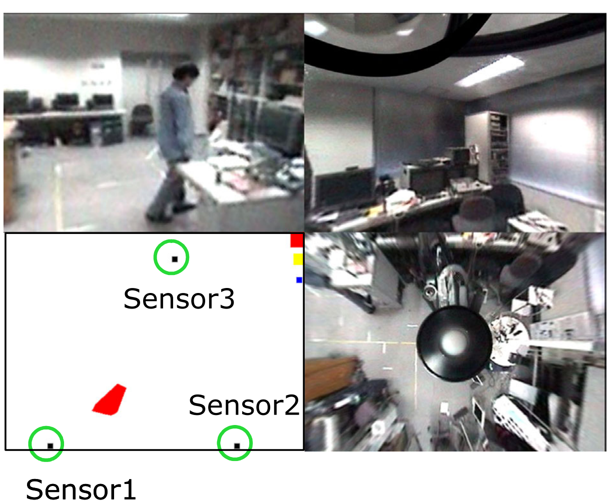


Fig. 6 Displayed image to the user

5 Conclusions

This paper describes networked video surveillance using multiple omnidirectional cameras and robustly detecting moving objects with a background subtraction technique based on adaptively updating background images. Our system is server/client model. Each server has an omnidirectional camera, detects moving objects, and estimates the directions of the objects. The client receives the directions and the omnidirectional video from the servers through network, estimates the positions of the objects, and shows the user the positions and the images of the objects. In the experiment, the system has been successfully demonstrated in door environments.

The future work includes the improvement of updating the areas of detected moving objects and the tracking of detected moving objects between adjacent frames.

References

[1] T.Wada and T.Matsuyama. Appearance sphere: Background model for pan-tilt-zoom camera, In *Proc. ICPR96*, vol.A, pages 718-722, 1996.

[2] Mikic, S.Santini, and R.Jain. Tracking Objects in 3D using Multiple Camera Views, In *Proc. ACCV2000*, pages 234-239, 2000.

[3] R.Miki, N.Yokoya, K.Yamazawa, and H.Takemura. A Real-time Surveillance and Monitoring System Using Multiple Omnidirectional Video Cameras, In *Proc. ACCV2000*, pages 528-534, 2000.

[4] T.Sogo, H.Ishiguro and M.M.Trivedi. Real-time target localization and tracking by N-ocular stereo, In *Proc. IEEE Workshop on Omnidirectional Vision*, pages 153-160, 2000.

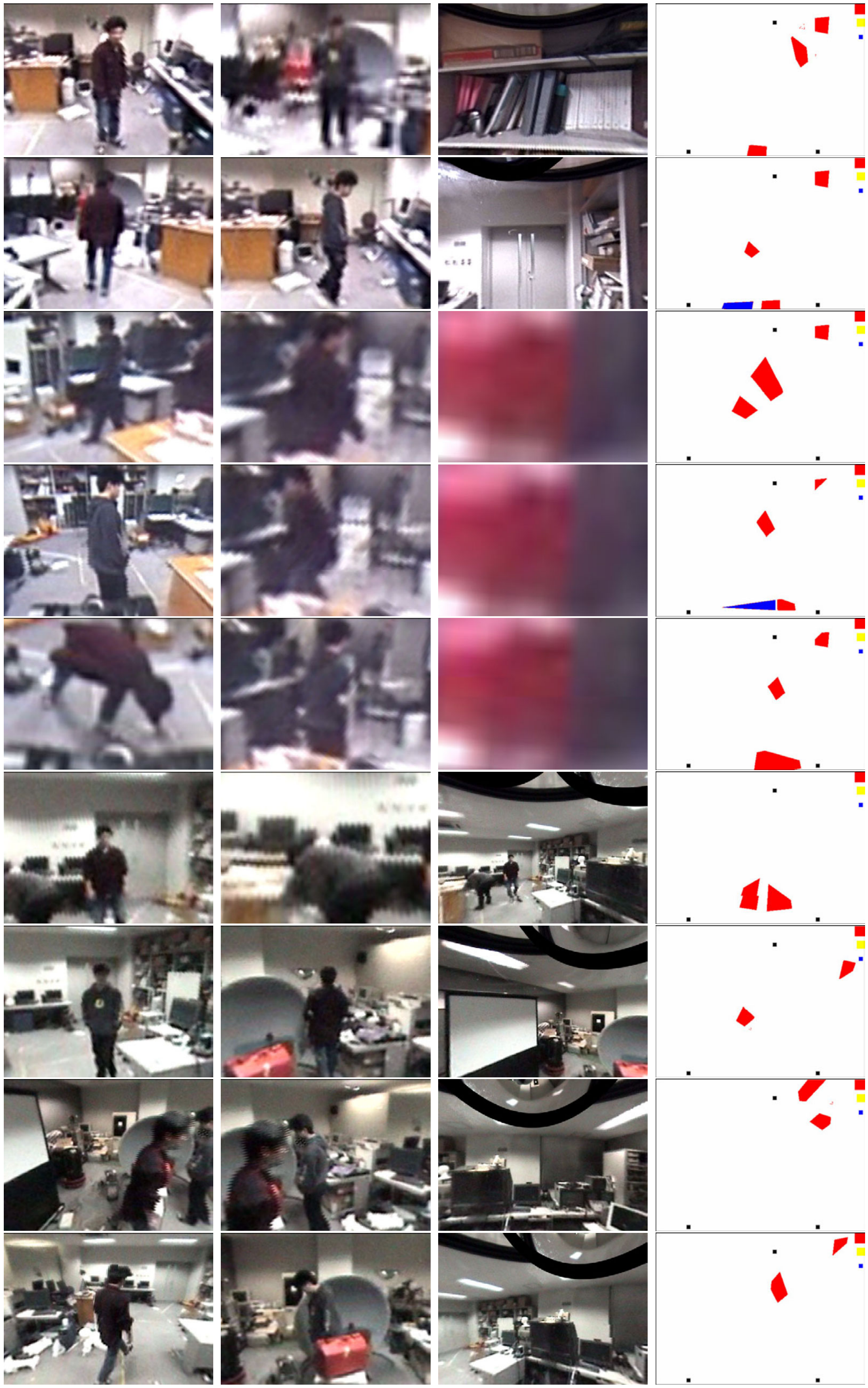
[5] K.Yamazawa, Y.Yagi, and M.Yachida. Obstacle Detection with Omnidirectional Image Sensor HyperOmni Vision, In *Proc. Int. Conf. on Robotics and Automation*, pages 1062-1067, 1995.

[6] Y.Onoe, K.Yamazawa, H.Takemura, and N.Yokoya. Telepresence by Real-Time View-Dependent Image Gen-eration from Omnidirectional Video Streams, *Computer Vision and Image Understanding*, 71(2): 154-165, 1998.

[7] A.Ogawa, K.Kobayashi, O.Nakamura, and J.Murai. Design and Implementation of DV Stream Over Internet, In *IWS Internet Workshop*, 1999.

Camera1 Camera2 Camera3

Fig. 7 Input images and detected objects



Upper left Upper right Lower right Estimated position

Fig. 8 Displayed image sequence