Cooperative Surveillance System with Fixed Camera Object Localization and Mobile Robot Target Tracking

Chih-Chun Chia, Wei-Kai Chan, and Shao-Yi Chien

Media IC and System Lab

Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan

b93901158@ntu.edu.tw, r3943041@ee.ntu.edu.tw, sychien@cc.ee.ntu.edu.tw

Abstract. This paper presents a cooperative surveillance system. It presents a cooperation scheme between fixed cameras and a mobile robot. The fixed cameras detect the objects with background subtraction and locate the objects on a map with homography transform. At the same time, the information of the target to track, including the position and the appearance, is transmitted to the mobile robot. After Breadth First Search in a map of boolean array, the mobile robot finds the target in its view by use of a stochastic scheme with the information given, then the mobile robot will track the target and keep it in the robot's view wherever he or she goes. By proposing this system, the dead spot problem in typical surveillance systems with only fixed cameras is considered and resolved.

1 Introduction

Recent approaches in surveillance systems typically include the use of static cameras along with the content analysis algorithm [1] [2]. The drawback is that dead spot can not be covered and intruders can try to avoid the fixed camera's sight, which results in less robustness for the surveillance system. Systems employing PTZ cameras or omni-directional camera system can increase the covering range [3] [4] [5]. However, there may still be dead spots and the covering area still depends on the cameras' positions decided in the deployment phase. Besides, several object tracking algorithms that are capable of tracking targets with a mobile camera are developed in recent years [6] [7]. The critical issue in the use of these algorithms is the initialization of the track, which is usually manually selected without automatic initialization scheme.

We now propose and construct a cooperative prototype system which consists of fixed cameras and a mobile robot as shown in Fig. 1. In this system, we propose a cooperation scheme between fixed cameras and a mobile robot. The fixed cameras can do object detection [8] [9] and feature extraction automatically. Then, the object localization in the environment is achieved with homographic relations [10], which is constructed in the camera calibration phase, between the

fixed cameras and a global map. After these fixed camera operations, the information of the object's location in the map and its appearance is provided to the mobile robot, in which a target finding algorithm and a target tracking algorithm are implemented. The target finding is done robustly with a stochastic scheme presented in this paper. After finding the target, we use a particle filter based method to track the target and the robot will follow the target throughout the entire environment and keep it in the center of the robot's view. Note that, by proposing this system, the restriction of dead spots in typical surveillance systems is resolved with a mobile robot and the intruders will be tracked wherever he or she goes.

The remaining sections are organized as follows. Section 2 describes the process of intruder detection and localization. Section 3 presents the algorithm on target finding and tracking with mobile robot. Section 4 presents the implementation of our prototype system and gives the experimental results. Finally, Section 5 concludes this paper.

2 Target Detection and Localization

In this section, we introduce the *Target Detection and Localization* subsystem in Fig. 1 (b). In this subsystem, two localization mechanisms, Zigbee Localization and Vision Localization, are presented and integrated.

2.1 Zigbee Localization

Zigbee is a specification for a suite of high level communication protocols using small, low-power digital radios based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). It is generally targeted at radio-frequency applications that require a low data rate, long battery life, and secure networking.

Being widely deployed in wireless monitoring applications with high-reliability and larger range, Zigbee transmitters spread around the environment. We assume that all in-comers who are authorized to come in should be wearing a Zigbee receiver so that their locations can always be monitored. The control center can receive information about the number of authorized visitors and their rough locations [11].

2.2 Vision Localization

Our vision Localization is based on homography transform [10]. After background subtraction, which is a technique widely employed in surveillance applications [2] [8] [9], and appropriate denoising, the segmentation result can be used to detect and localize the objects in the view of each fixed camera. Here, we use the bottom-centroid of segmented object blob as the object's location in the view of each fixed camera. In order to localize the object in a global coordinate system with map, a homography transform [10] is employed to build the correspondences between fixed camera views and a global map. Homography

Fig. 1. (a) The proposed cooperative system; (b) the proposed cooperation scheme

is defined as a mapping between a point on a ground plane as seen from one camera, to the same point on the ground plane as seen from the second camera. Here, we extend it to build the point correspondence between fixed camera views and a global map. As shown in Fig. 2(a), two views O_l and O_r are viewing the

Fig. 2. (a) Two views viewing the same point; (b) shot of the test environment taken by one of the fixed cameras and a simple global map. The colored points are the four corresponding pairs of points chosen.

same point P . The coordinate of P in the left view's ground plane is denoted as P_l , while P_r represents that of the right view. Through the mathematical definition of homography, which is given by

$$P_l = \begin{pmatrix} X_l \\ Y_l \\ 1 \end{pmatrix}, P_r = \begin{pmatrix} X_r \\ Y_r \\ 1 \end{pmatrix}, H_{lr} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \quad (1)$$

Then, we will have :

$$\omega * P_r = H_{lr} P_l \quad (2)$$

In the above equation, ω is a scaling factor [10]. According to [10], if we have four pairs of corresponding points with coordinates denoted by (X_{lc}, Y_{lc}) and (X_{rc}, Y_{rc}) (where $1 \leq c \leq 4$, l and r stands for the two views), we can obtain a system of eight equations with eight unknowns while letting H_{33} constant as 1. The matrix H_{lr} can be solved by:

$$\begin{pmatrix} X_{l1} & Y_{l1} & 1 & 0 & 0 & 0 & -X_{l1} * X_{r1} & -Y_{l1} * X_{r1} \\ X_{l2} & Y_{l2} & 1 & 0 & 0 & 0 & -X_{l2} * X_{r2} & -Y_{l2} * X_{r2} \\ X_{l3} & Y_{l3} & 1 & 0 & 0 & 0 & -X_{l3} * X_{r3} & -Y_{l3} * X_{r3} \\ X_{l4} & Y_{l4} & 1 & 0 & 0 & 0 & -X_{l4} * X_{r4} & -Y_{l4} * X_{r4} \\ 0 & 0 & 0 & X_{l1} & Y_{l1} & 1 & -X_{l1} * Y_{r1} & -Y_{l1} * Y_{r1} \\ 0 & 0 & 0 & X_{l2} & Y_{l2} & 1 & -X_{l2} * Y_{r2} & -Y_{l2} * Y_{r2} \\ 0 & 0 & 0 & X_{l3} & Y_{l3} & 1 & -X_{l3} * Y_{r3} & -Y_{l3} * Y_{r3} \\ 0 & 0 & 0 & X_{l4} & Y_{l4} & 1 & -X_{l4} * Y_{r4} & -Y_{l4} * Y_{r4} \end{pmatrix} \begin{pmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \end{pmatrix} = \begin{pmatrix} X_{r1} \\ X_{r2} \\ X_{r3} \\ X_{r4} \\ Y_{r1} \\ Y_{r2} \\ Y_{r3} \\ Y_{r4} \end{pmatrix} \quad (3)$$

Note that we use the global map as one view in the homography transform and a fixed camera's view as the other view. One example of four-points correspondence to solve matrix H_{lr} is shown in Fig. 2(b) After solving the matrix in the camera calibration phase, the object localized in each fixed camera's view can be given a global coordinate on the map. The vision localization gives information about all objects, including authorized and unauthorized, when they are detected with background subtraction.

2.3 Integration for Identification

Identification (Intruder detection) can be easily done by comparing the results from Zigbee Localization in Subsection 2.1 and those from Vision Localization in Subsection 2.2. From the Zigbee Localization, the system may receive information about the number of authorized visitors and their rough locations. At the same time, fixed cameras can find the objects(people), including authorized and unauthorized, and locate them in terms of global coordinates. Intruders who have no authority to come in can be detected by comparing the object information from the fixed cameras with those from the Zigbee.

3 Mobile Robot for Tracking

From section 2, the coordinate of the intruder can be inferred. A template of the intruder can also be obtained from segmentation. Information about object's location and appearance will then be transmitted immediately to the mobile robot to start tracking. In this section, we will introduce our *Mobile Robot for Tracking* subsystem in Fig. 1 (b).

3.1 Target Model and Similarity Measure

Since the camera on the robot may be different from the fixed cameras, specific camera calibration techniques, including those for cameras with different lighting conditions and orientations, can be useful for the processing and analysis [12]. In this paper, the template of the intruder will be analyzed into a color histogram. The object, or model image, is represented by an ellipse. The sample points of the model image are denoted by x_i and $h(x_i)$, where x_i is the 2D coordinates and $h(x_i)$ is the corresponding color index of the histogram. The number of color indexes used denoted as β . As long as we have segmented out the suspected intruder's image, we can construct the p.d.f of the object's color distribution as follows [7]:

$$p(u_j) = \sum_{i=1}^I k(\|\frac{x_i - c}{\sigma}\|) \delta[h(x_i) - u_j], 0 \leq j \leq \beta \quad (4)$$

where I is the number of pixels in the region, u_j is the color index. σ is the bandwidth in the spatial space and c is the center of this space. δ is the well-known Kronecker delta function(1 if $h(x_i) = u_j$, 0 otherwise). To increase the

reliability of the color distribution, smaller weights are assigned to pixels farther away from the center (denoted as c), which are more likely to belong to the background. The weighting function we use is the well-known Epanechnikov kernel : $k(u) = \frac{3}{4}(1 - u^2)$, $|u| \leq 1$.

Here we adopt the commonly used similarity measure called Bhattacharyya coefficient. We can estimate the p.d.f. of the model image and that of the candidate image (denoted by p_x and p_y , respectively), then measure the dissimilarity between these two distributions as below [7]:

$$B(I_x, I_y) = \sqrt{1 - \rho(p_x p_y)} \quad (5)$$

where function ρ is defined as

$$\rho(p_x, p_y) = \int \sqrt{p_x(u)p_y(u)} du \quad (6)$$

3.2 Target Finding with Mobile Robot

The mobile robot will receive information about the intruder, including its coordinate and appearance (color distribution). We assume that the robot stays at a fixed location when no event occurs. With the target coordinate given, the mobile robot may go to an initial location that is convenient to find the target. The initial location and the initial direction of the robot's view can be decided according to different environments in advance with respect to any target location. The mobile robot can go to any location, including the initial location, by Breadth First Search with a 2D boolean array, in which the array element stores 1 or 0 indicating whether reachable or not [13].

After arriving at the initial location and turning to the initial direction with the help of a compass module on the robot, the robot have to find the target within its view. In order to do so, we employ a stochastic scheme here. First, we randomly choose different regions to make hypotheses about the target's location and size. Regions are again represented by an ellipse, with randomly chosen center. However, the ratio of the two axis is fixed into 3.5:1 (which approximately stands for a person's ratio in height and width).

The number of regions chosen is denoted by N_r . Every region is analyzed into color histogram, and we estimate the Bhattacharyya distance (denoted as B_i , $1 \leq i \leq N_r$) between those hypothetical regions and the model image.

With the Bhattacharyya distance for every hypothetical regions, we can further calculate the mean μ_r and standard deviation σ_r of B_i ($1 \leq i \leq N_r$). Thus we can select the regions with their Bhattacharyya distance less than $\mu_r - 2\sigma_r$. That is to say, if we assume that the distances between hypothetical regions and the model image is approximately Gaussian distributed, this would indicate 95% confidence level that the selected region is almost our target. Finally, we estimate the target's location and size by averaging the chosen regions' location and size.

3.3 Target Tracking with Mobile Robot

We use particle filter with color-based features to do target tracking with mobile robot. Particle filter, also known as Sequence Monte Carlo method or Sampling-Importance-Resampling (SIR) filter, is a state estimation technique based on simulation [14] [15]. Here, the state of target is described by the center of the ellipse and a scale factor representing the length of axis, since the ratio of the two axes is fixed.

The idea of particle filter is to evaluate the probability of all the particles and thus estimate the location of our target. We use a particle sample set $S = \{s^{(n)} | n = 1 \dots N\}$, each sample s is a hypothetical state of the target.

After successfully locating the target according to Section 3.2, we can construct a sample set with all the samples equivalent to the target just found and then start evolution. Evolution of the particles is described by propagating each particle according to a Gaussian noise added to its center.

Many previous approaches may have propagated the particles according to a system model including moving direction and speed in order to model a person who may move in the same way and constant speed most of the time. However, we do not consider moving direction and speed here, since in our case, the tracker robot is also moving.

After the particles have finished propagation, weighting of the sample set can be computed by estimating the Bhattacharyya coefficient. Obviously, we would give larger weight to samples whose color distributions are more similar to the model. The weighting of each sample is given as follows:

$$\pi'^{(n)} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1-\rho(p_{(n)},q)}{2\sigma^2}} \quad (7)$$

Normalizing $\pi'^{(n)}$ with the following equation, we obtain $\pi^{(n)}$.

$$\pi^{(n)} = \pi'^{(n)} / \sum \pi'^{(n)} \quad (8)$$

We can then estimate the location of our target as:

$$E[S] = \sum_{n=1}^N \pi^{(n)} s^{(n)} \quad (9)$$

We determine the scale of the target by testing scales between 0.9 and 1.1 of the original size, since a person obviously have speed limit. Between 0.9 and 1.1, β scales are chosen and their weighting are evaluated from Bhattacharyya distance. A weighted average of these scales is calculated to obtain the final size of the target.

In the last step, i.e. resampling, samples with higher weights will be reproduced more times than the samples with lower weights [14]. When the next frame comes, the whole process can be repeated again to continue tracking.

Note that using mobile robot is a big difference in our tracking job from others. After locating the target in each frame, the robot will judge if the target

Fig. 3. Some key shots from a tracking test sequence

is in the left-side or the right-side of its view in order to decide which way it should turn. At the same time, it makes decision on going forward or backward according to the change in target's scale.

Here, we also present a way to make the mobile robot move more smoothly by using threshold. Since video frames may be taken at times when the robot is turning or when the person is moving irregularly, so make moving judgment every frame rather diminished its tracking ability. Action-threshold is set so that only if there are consecutive α frames making the same judgment will the robot actually move or turn.

Fig. 3 shows some key shots from a tracking result. The tracked target is bounded by a thick black ellipse. We see that no matter the target moved right or left, the robot turns with it to keep the target in the center of its view.

4 System Implementation and Experiment

4.1 System Implementation

All of the processing tasks are implemented on a Core 2 Quad 2.4GHz PC (1066MHz FSB) with C# in Visual Studio 2005. The mobile robot we used is Dr.Robot X80, which offers full WiFi (802.11g) wireless multimedia transmission. Due to the low data rate transmission limit in the robot, we abandoned the use of camera built inside Dr.Robot X80. Instead, Vivotek Network Camera IP 7137, a wireless camera with video streaming function, provides the video for analysis. The IP 7137 Network Camera is placed on the robot and the captured video is transmitted to the PC to analysis. After the video is analyzed, the command to control robot movement is transmitted from the PC to robot. The robot and camera is shown in Fig. 4. The overall system can track the target with the robot at 3-5 fps while the code is not optimized.

Our test environment is the technology exhibition center at Ming-Da building 1st floor of Electrical Engineering Department of National Taiwan University with three fixed camera. The individual views of these three cameras are shown in Fig. 5. The videos captured with these fixed cameras are also transmitted to the PC for analysis with our algorithms.

4.2 Experimental Result

In this section, we present some results of this cooperative surveillance system. In the first place, we setup two different scenarios for testing this system. The

Fig. 4. (a) Dr.Robot X80; (b) Vivotek WLAN Network Camera IP7137

Fig. 5. The environment of Ming-Da Building: (a) A simple map of the environment with the three fixed cameras and their view range marked; (b) the image of camera colored in red; (c) the image of camera colored in blue; (d) the image of camera colored in green

first one(Fig. 6) is that an intruder comes in and goes to dead spots that fixed cameras can not cover. The second scenario(Fig. 7) is that an intruder is going out of the building, which the fixed camera obviously cannot cover. In these two scenarios, our system have shown great robustness in successfully locating and tracking the intruder.

Each figure contains one map with four sequential images. The map shows the intruder and the robot's route, where the small black man represents the intruder and a mark (viewpoint) represents the robot and its viewing direction. The four sequential images are taken from the moving camera (i.e. the robot's view), and in each image, the target being tracked is shown with an ellipse surrounding it. In the map, the position of the green mark(robot) and the green-framed small man(target) shows where the first image is taken, while blue for the second

Fig. 6. Scenario one - Dead Spot Experiment, in image (d) and (e), the intruder goes into a dead spot where the fixed cameras cannot see

Fig. 7. Scenario two - Out of Building Experiment, in image (d) and (e), the intruder escaped out of the door where the fixed camera cannot continue tracking

image, red for the third image, and purple for the last image. In scenario 1, we also show the moment that the robot found the target by including Fig. 6(b) taken with the fixed camera.

5 Conclusions

This paper presents a cooperative surveillance system with fixed camera target localization and mobile robot target tracking. Vision localization is based on fixed camera's segmentation combined with the concept of homography. After locating the intruder, the robot will receive the commands, then start target finding and tracking by a stochastic scheme and a particle filter based tracking algorithm. In this system, the dead spot problem in typical surveillance systems with only fixed cameras is overcome with mobile robot target finding and tracking.

References

1. Regazzoni, C., Ramesh, V., Foresti, G.L.: Special issue on video communications, processing, and understanding for third generation surveillance systems. *Proc. IEEE* 89(10), 1355–1367 (2001)
2. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 246–252 (July 1999)
3. Micheloni, C., Foresti, G.L., Snidaro, L.: A Network of Cooperative Cameras for Visual-Surveillance. *IEE Proc. on Visual, Image and Signal Processing*, 152(2), 205–212 (2005)
4. Foresti, G.L., Micheloni, C., Snidaro, L., Remagnino, P., Ellis, T.: Active Video-based Surveillance Systems: the low-level image and video processing techniques needed for implementation. *IEEE Signal Processing Magazine* 22(2), 25–37 (2005)
5. Iwata, K., Satoh, Y., Yoda, I., Sakaue, K.: Hybrid Camera Surveillance System by Using Stereo Omni-directional System and Robust Human Detection. In: Chang, L.-W., Lie, W.-N. (eds.) *PSIVT 2006*. LNCS, vol. 4319, pp. 611–620. Springer, Heidelberg (2006)
6. Maggio, E., Smerladi, F., Cavallaro, A.: Adaptive Multifeature Tracking in a Particle Filtering Framework. *IEEE Trans. Circuits Syst. Video Technol.* 17(10), 1348–1359 (2007)
7. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Trans. Pattern Anal. Machine Intell.* 25(5), 564–577 (2003)
8. Chien, S.-Y., Ma, S.-Y., Chen, L.-G.: Efficient Moving Object Segmentation Algorithm Using Background Registration Technique. *IEEE Trans. Circuits Syst. Video Technol.* 12(7) (July 2002)
9. Chan, W.-K., Chien, S.-Y.: Real-time memory-efficient video object segmentation in dynamic background with multi-background registration technique. In: *Proc. IEEE Multimedia Signal Processing Workshop* (October 2007)
10. Bradshaw, K.J., Reid, I.D., Murray, D.W.: The active recovery of 3D motion trajectories and their use in prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(3), 219–234 (1997)

11. Lorincz, K., Welsh, M.: MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. In: Strang, T., Linnhoff-Popien, C. (eds.) LoCA 2005. LNCS, vol. 3479, pp. 63–82. Springer, Heidelberg (2005)
12. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision. Addison Wesley, Reading (1992)
13. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. McGraw Hill/The MIT Press (2001)
14. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House (Copyright 2004)
15. Nummiaro, K., Koller-Meier, E., Gool, L.V.: An Adaptive Color-Based Particle Filter. *Image and Vision Computing* 21(1, 10), 99–110 (2003)
16. Chang, C., Ansari, R.: Kernel Particle Filter for Visual Tracking. *IEEE Signal Processing Letters* 12(3) (March 2005)