ROBUST PEOPLE DETECTION AND TRACKING IN A MULTI-CAMERA INDOOR VISUAL SURVEILLANCE SYSTEM

Tao Yang[1,2] *Francine Chen*[1] *Don Kimber*[1] *Jim Vaughan*[1]

[1] FX Palo Alto Laboratory, Palo Alto, CA, 94304, USA
[2] Northwestern Polytechnical University, Xi'an, 710072, China
yangtaonwpu@163.com {chen,kimber,vaughan}@fxpal.com

ABSTRACT

In this paper we describe the analysis component of an indoor, real-time, multi-camera surveillance system. The analysis includes: (1) a novel feature-level foreground segmentation method which achieves efficient and reliable segmentation results even under complex conditions, (2) an efficient greedy search based approach for tracking multiple people through occlusion, and (3) a method for multi-camera handoff that associates individual trajectories in adjacent cameras. The analysis is used for an 18 camera surveillance system that has been running continuously in an indoor business over the past several months. Our experiments demonstrate that the processing method for people detection and tracking across multiple cameras is fast and robust.

1. INTRODUCTION

Many commercial surveillance systems target a small number of susceptible areas in a business and often make use of "hotspots" or "tripwires" to identify activities of interest. Such systems include those offered by Panasonic, Vidient, Verint, Vistascape, NiceVision and ObjectVideo. Although intelligent use of these techniques enables many useful functions, including tailgating detection, counting people passing through a region, or detecting an intruder in a forbidden area, other surveillance functions are needed to secure an office building.

In contrast to these systems, we are developing a multi-camera surveillance system for monitoring an office building, where one goal is to track people of interest. Eighteen AXIS 210 network cameras are located to cover the public spaces, where the system automatically detects and tracks people. Fig. 1 is an overview block diagram of the system. The video recorder server receives video data from multiple remote cameras through a local network and is responsible for sending live video to the "video tracker PCs" for analysis. The tracking result of each single camera is saved into the video content analysis database. The multiple camera information fusion PC receives object trajectories and features from the database and integrates

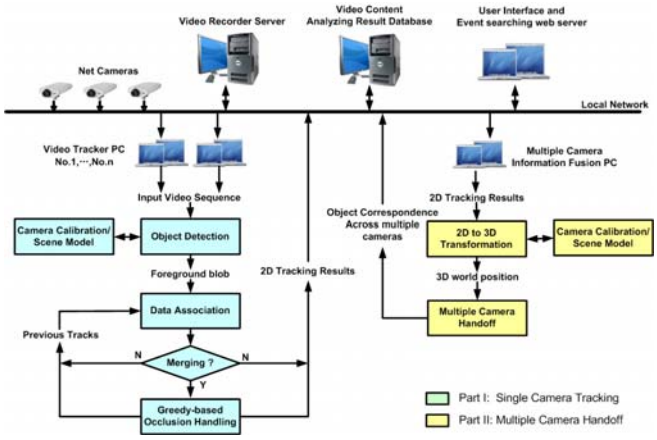

Fig. 1. Overview of the multi-camera surveillance system

them together with a 3D geometry scene model to estimate the final object correspondence across multiple cameras.

When a person enters the floor from the elevator, the person's face is detected in the video of a camera pointed toward the elevator. The system matches each detected face to a track of a person leaving the elevator. Security personnel can then browse the recorded faces and select a person. A description of the system's user interface is given in Girgensohn et al. [1]. The interface shows where the selected person is walking on a floorplan (see Fig. 5 for a sample floorplan), a bank of cameras and four larger views that includes the best view of the selected person.

In order to track people in real time over extended periods, across multiple cameras, possibly without overlapping views, we focus on two key analysis components in this paper:

1. Real-time and reliable people segmentation, tracking, and occlusion handling in a single camera
2. Consistent labeling of people across multiple cameras

2. SEGMENTING MULTIPLE HUMANS

The first step in processing is segmenting people from the background. We use the popular Gaussian mixture model approach for pixel level background modeling (Stauffer et al.[2]), and convert color images to grayscale prior to

computing the model, so that either color or grayscale images are handled. The model is adaptively updated with information from new frames.

For foreground segmentation that is robust to shadows, similar colors, and illumination changes in the background, many methods [3,4] mainly adopt two basic processing steps: (1) using pixel to pixel subtraction to get a rough foreground image, and (2) using a post processing model to remove false alarm pixels of detected foreground regions. When a foreground pixel is not detected by pixel level subtraction due to similar color, those methods [3,4] will not classify this pixel as foreground. In addition, training data is needed to detect shadows [4].

In contrast, our method directly builds foreground images by a feature level subtraction approach without any post processing module. First, we use the foreground density around each pixel to determine a candidate set of foreground pixels. If more than 10% of the pixels in a local region are labeled foreground based on pixel level background subtraction [2], the pixel centered in the region is considered as a candidate foreground pixel. Then, instead of comparing the difference in intensity value of each pixel, we compute the difference of the neighborhood image. In general, any metrics that can measure the similarity between two images can be used here. In the current system, for each candidate pixel $I_{xc,yc}$, the normalized cross correlation (1) is selected to compare the similarity of the local region R centered at (*xc,yc*) and the integral image method is used to compute (1) in real time [5]:

$$\rho(R^B, R^I) = \frac{\sum_{(x,y)\in R}((R^B(x,y)-\overline{R}^B)(R^I(x,y)-\overline{R}^I))}{\sqrt{\sum_{(x,y)\in R}(R^B(x,y))-\overline{R}^B)^2}\sqrt{\sum_{(x,y)\in R}(R^I(x,y))-\overline{R}^I)^2}} \quad (1)$$

where R^B and R^I denote the local region on the reference background and input image respectively. By subtracting the corresponding mean value $\overline{R}^B$ or $\overline{R}^I$, and normalizing the image vector to unit length, equation (1) is robust to changes in image amplitude such as illumination changes and shadows. In addition, based on camera calibration information, the size of the region *R* may vary due to different distances. Examples of segmentation under similar color and shadows are given in Fig. 3.

3. TRACKING HUMANS THROUGH OCCLUSION

Techniques for tracking multiple humans under occlusion can be categorized as either merge-split (MS) or straight-through (ST). In the MS approach, overlapping objects are encapsulated into a new group. When one of the objects splits from the group, the identify of that object is re-established using appearance features [6,7] such as color, texture and shape. When the number of objects in a group is larger than two, the MS method frequently fails because it's hard to tell how many objects are inside each splitting blob.

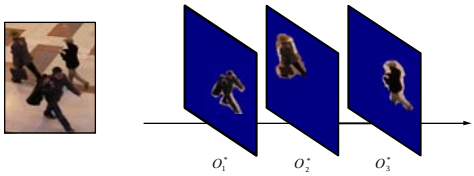

Fig. 2. Object visible ratio during occlusion

In the ST method, individual objects are tracked through occlusion. Many ST methods [8] adopt bottom-up segmentation and use an appearance model to assign each pixel to a certain tracker.

We pose the problem of object tracking through occlusion as a track-based segmentation problem in the joint object space. First, a *data association module* classifies complex interactions between moving blobs. We follow previous works [7,8] in the use of a correspondence matrix to classify object interactions into five classes: 1) Continuation, 2) Appear, 3) Disappear, 4) Merge and 5) Split.

Next, a *greedy search based occlusion handling module* decodes the best configuration of occluding objects in the group *G* in real-time. Appearance features that are computed during tracking are used to estimate the matching scores in a Bayesian framework. When a merge occurs, the objects involved in occlusion are identified. The search for the most probable configuration of a group O_g^* becomes a maximum likelihood estimation problem where:

$$O_g^* = \arg\max_{O_g} P(O_g \mid G) \quad (2)$$

Since objects may occlude each other, each object O_i is NOT conditionally independent of every other object O_j for $i \neq j$. Using conditional probability, $P(O_g|G)$ can be written as:

$$P(O_g \mid G) = P(O_1 \mid G)\prod_{i=2}^{N} P(O_i \mid G, O_1, ..., O_{i-1}) \quad (3)$$

For a configuration $O_g^* = \{O_1^*, ..., O_N^*\}$, we can order the *N* objects into *N* layers according to their visible ratios (Fig. 2), computed as the fraction of the object model. Usually the object with higher visible ratio in the group will have a higher observation probability. Thus we can directly find the object O_1^* in the first stage by (4):

$$O_1^* = \arg\max_{O_i} P(O_i \mid G) \quad (4)$$

After that, we can find the position of objects in other stages by searching the maximum probability in each stage:

$$O_m^* = \arg\max_{O_i} P(O_i \mid G, O_1^*, ..., O_{m-1}^*) \quad (5)$$

where $i = 1,...N, O_i \notin \{O_j^*\}, j = 1,...,m-1$.

To compute the probability $P(O_i|G, O_1^*, ..., O_{m-1}^*)$ at a stage *m*, we scan each object model over the entire group G, and use equation (6) to estimate the probability:

$$P(O_i \mid G, O_1^*, ..., O_{m-1}^*) = \max(P(O_i \mid F_{xc})), xc \in G \quad (6)$$

where F_{xc} is the covered foreground image inside the object's mask and centered at pixel *xc*, and $P(O_i|F_{xc})$ is computed as the average probability over the pixels (7):

$$P(O_i \mid F_{xc}) = \frac{1}{w \cdot h} \sum_{x_k \in F_{xc}} P(O_i \mid I(x_k)) \quad (7)$$

where $I(x_k)$ is the intensity value of the pixel located at x_k, w and h is the width and height of object O_i .The conditional probability $P(O_i|I(x_k))$ is computed using Bayes' theorem as:

$$P(O_i \mid I(x_k)) = \frac{P(I(x_k) \mid O_i)P(O_i)}{\sum_{s=1}^{N} P(I(x_k) \mid O_s)P(O_s)} \quad (8)$$

where $O_s \notin \{O_j^*\}, j = 1,...,m-1$.

We estimate $P(I(x_k)|O_s)$ based on the color histogram of object O_s, $P(O_s)$ is the comparative size of the objects before occlusion, and the sum of the pixel probabilities in (7) is computed by a two-dimensional integral image in real-time. For pixels covered by a selected object in previous stages, we discount its probability $P(O_s|I(x_k))$ according to its distance to the center of nearest objects selected in previous stages. Some experimental results are given in section 5.

4. MULTIPLE CAMERA HANDOFF

Once objects are tracked in single camera views, multiple camera handoff is used to determine the likelihood that multiple tracks result from the same object. Based on the relations between camera views, the handoff methods can be roughly classified as two classes: with overlapping views [9] and non-overlapping views, or "blind" regions [10].

In our system, we propose a new hand-off function that is based on the spatial-temporal matching ratio for all pairs of tracks in two views. The camera handoff module operates in two steps: initialization and track matching.

In the *initialization step*, camera calibration is performed and the camera streams are synchronized temporally across the local network by a digital video recorder server. The minimum and maximum transition time $\tau_{\min}$ and $\tau_{\max}$ of an object passing from camera i to camera j are automatically learned by single camera tracking results of the training data, in which multiple people freely walk across all cameras.

In the *track matching step,* once a new object b appears in a certain camera j, the handoff module will be triggered. Matching ratios M between track T_b and every track in all connected cameras are computed, the track with maximum matching ratio is selected as a candidate for handoff, and if the maximum matching ratio is larger than a threshold, the two tracks will be labeled as the same object.

- For cameras with overlapping fields-of-view, the matching ratio $M_{a,b}$ is estimated as the fraction of time during the overlap interval [t_{min} , t_{max}] that the tracks are within world distance threshold hd:

$$M_{a,b} = \frac{1}{t_{\max} - t_{\min}} \sum_{ti=t_{\min}}^{t_{\max}} k\left(\left\| \frac{T_a(t_i) - T_b(t_i)}{hd} \right\|\right) \quad (9)$$

where $T_a(t_i)$ and $T_b(t_i)$ are world positions on trajectories of T_a and T_b at time t_i, and k(x) = 1 if $\|x\| < 1$, and 0 otherwise.

- In our indoor surveillance system, people may pass from one camera view to another through "blind" regions in which they cannot be seen. The matching ratio in these cases is computed as:

$$M_{a,b} = \begin{cases} P(q_u(a), q_u(b)) & \Delta t \in [\tau_{\min}, \tau_{\max}] \\ 0 & otherwise \end{cases} \quad (10)$$

where Δt is the time interval between the start of track T_b in camera j and end of track T_a in camera i, and P is the color similarity between objects a and b with color models q_u(a) and $q_u(b)$.

Table 1 Performance evaluation results of IBM datasets

TRDR	FAR	DR	SP	AC
0.998703	0.102534	0.972538	0.922038	0.973263
PP	**NP**	**FPR**	**FNR**	**OER**
0.897466	0.979529	0.077962	0.027462	0.22

5. EXPERIMENTS

We now present sample results from our fully automated system that tracks people across multiple cameras. The system has been running continuously for over half a year. The input image size is 320x240 and four live videos are processed per standard PC at a speed of 7fps. The sample results are for foreground segmentation, occlusion handling, and camera handoff.

Fig. 3 shows segmentation results for three examples. A standard Gaussian mixture model [2] fails to detect many of the foreground pixels (second column) in Fig.3a when the foreground is similar to its background, and erroneously detects shadows in Fig.3b. Using local region information, our proposed feature level segmentation method successfully handles these problems. The third column shows our original segmentation results without post-processing or size filtering. The fourth column contains the final segmented foreground object after morphological filtering.

We report our tracking results on the IBM datasets [11] and our office video. We manually labeled ten IBM indoor sequences which contain complex interactions between multiple persons and shadows. The number of frames containing ground truth objects is 3085, out of 7443 frames total. The frame-based metrics defined in Porikli et al.[12] are used for performance evaluation. Table 1 shows the results, in which TRDR is tracker detection rate, FAR is false alarm rate, DR is detection rate, SP is specificity, AC is accuracy, PP is positive prediction, NP is negative prediction, FPR is false positive rate, FNR is false negative rate. Fig.4.a shows comparison of ground truth tracks and our tracking results in five IBM indoor sequences.

To evaluate tracking performance through occlusion, we define the occlusion error rate (OER) per object, that is, the error rate where an event is one object tracked through one occlusion. The total number of occlusion events in the ten IBM indoor sequences is 64. Our system correctly handles 50 events, for an OER of 0.22. Fig. 4.b shows examples of tracking through occlusion in our surveillance system.

We present sample results of the surveillance system in Fig. 5 for a scenario where a person is walking around the building. People tracking and correspondence results are displayed in ten key frames from A to J. For each sample time, images from various cameras that contain the same object ID are displayed. The top left number in a bounding box shows the track ID in each camera, and the bottom right number presents the object ID of the person after camera handoff. Trajectories are shown on the floor map. In the whole sequence, the person passed sixteen cameras. A person's appearance and scale is quite different in different camera views (Fig.5.H). The system successfully handles these problems, and in the whole sequence, the person's object ID changed only once.

6. CONCLUSION

We have presented the key components of a fully automated system that can track people in a multi-camera indoor surveillance scenario. There are a couple issues that should be addressed in the future. The use of color features to compute the probabilistic mask potentially can result in segmentation errors when there are similarly colored objects in a group. Another issue is that the handoff method cannot handle many people waking across the same cameras at the same time. We believe that the above limitations could be decreased in the future by adding local feature information.

7. REFERENCES

[1] A. Girgensohn, F. Shipman, T. Dunnigan, T. Turner, and L. Wilcox. Support for Effective Use of Multiple Video Streams in Security. *Proc. fourth ACM International Workshop on Video Surveillance & Sensor Networks,* Santa Barbara, CA, 2006.

[2] C. Stauffer, W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Volume 22 , Issue 8 , pp.747-757, 2000.

[3] Y.L. Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA, June 2005.

[4] F. Porikli, J. Thornton. Shadow Flow: A recursive method to learn moving cast shadows. *Proc. IEEE Computer Society Conference on Computer Vision*, Vol. 1, pp.891-898, 2005.

[5] J.P. Lewis. Fast normalized cross-correlation. *In Vision Interface*, 1995.

[6] S. McKenna, S. Jabri, Z. Duric, H. Wechsler. Tracking Groups of People. *Proc. Computer Vision and Image Understanding*, 2000.

[7] T. Yang, S.Z. Li, Q. Pan, J. Li. Real-time multiple object tracking with occlusion handling in dynamic scenes. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA, 2005.

[8] R. Cucchiara, C. Grana, G. Tardini. Track-based and object-based occlusion for people tracking refinement in indoor surveillance. *Proc. ACM 2nd International Workshop on Video Surveillance & Sensor Networks*, pp.81-87, USA, 2004.

[9] S. Khan, M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *Proc. IEEE Trans. on Pattern Analysis and Machine Intelligence*, Volume 25, No. 10, pp.1355-1360, October 2003.

[10] F. Porikli, A. Divakaran. Multi-camera calibration, object tracking and query generation. *Proc. IEEE International Conference on Multimedia and Expo*, Vol. 1, pp.653-656, July 2003.

[11] www.research.ibm.com/peoplevision/performanceevaluation.html

[12] F. Bashir, F. Porikli. Performance Evaluation of Object Detection and Tracking Systems. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance,* June 2006.

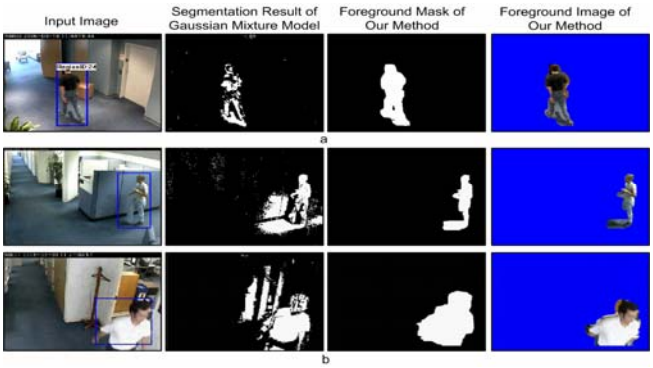

Fig. 3. People segmentation results a) foreground color similar to background b) with shadows

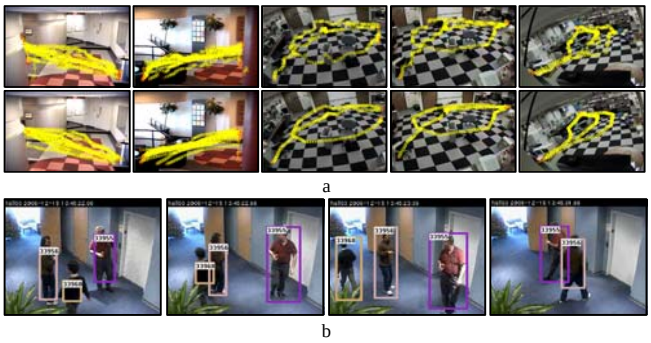

Fig. 4. Tracking results through occlusion. a) Our tracking results(first row) and ground truth (second row) of IBM datasets. b) Tracking examples of our surveillance system.

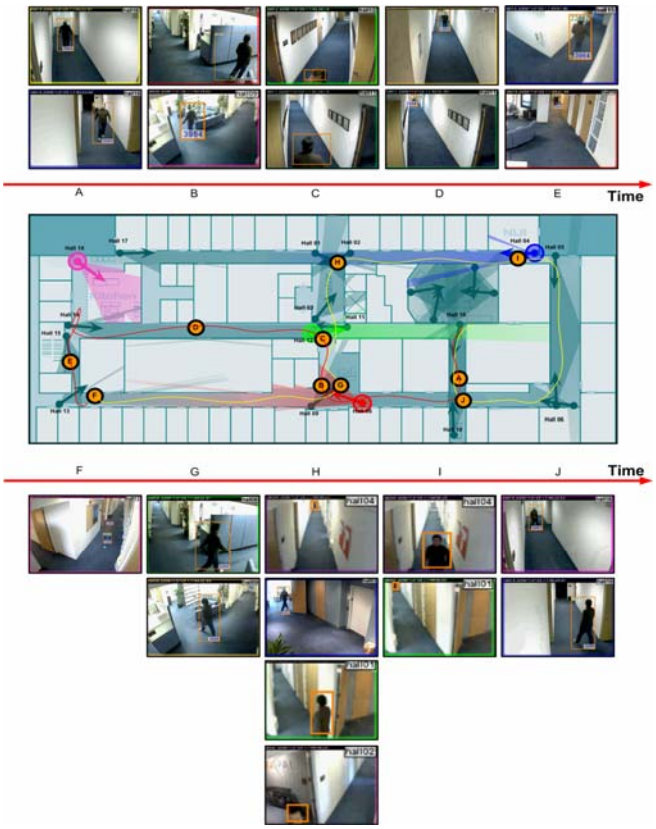

Fig. 5. People tracking across multiple cameras