KNIGHT^M: A Multi-Camera Surveillance System

Omar Javed and Mubarak Shah
Computer Vision Lab,
University of Central Florida, Orlando, Fl 32816
{ojaved,shah}@cs.ucf.edu

Abstract

In this paper, we present an automated wide area surveillance system that detects, tracks, classifies moving objects across multiple cameras. In addition, it detects unusual activities carried out by objects in the area under observation. At the single camera level, objects are detected using a robust background subtraction approach, then tracking is performed using a voting scheme that utilizes color and shape cues to establish correspondence. The system uses the single camera tracking results along with the relationship between camera field of view (FOV) boundaries to establish correspondence between views of the same object in multiple cameras. The proposed approach combines tracking in cameras with overlapping and/or non-overlapping FOVs in a unified framework, without requiring explicit calibration. The proposed algorithm has been implemented in a system with real time warning and report generation capability. The system uses client-server architecture and runs at 10 Hz with three cameras.

1 Introduction

In recent years, there has been a growing trend in both federal agencies and private firms to employ video cameras for monitoring and surveillance purposes. All video surveillance systems currently in use share one feature; a human operator must constantly monitor them. Their effectiveness and response is largely determined, not by the technological capabilities or placement of the cameras but by the vigilance of the person monitoring the camera system. The number of cameras and the area under surveillance is limited by the number of personnel available. Also even well trained people can't maintain their attention span for extended periods of time. Thus, there is an urgent need of automated surveillance and security systems with 24-hour warning capability.

Urban surveillance of wide areas requires a network of cameras. One of the major tasks of an automated surveillance system is to maintain the identity of a person moving across cameras in the environment. Most of the automated surveillance approaches require overlapping field of views (FOVs) for tracking targets across multiple cameras. However, it is not always possible to have cameras with overlapping FOVs while covering large areas in urban environments. In addition, site models or calibrated cameras are not available in many situations. Furthermore, maintaining complete calibration of a large network of sensors is a significant maintenance task, since cameras can accidentally be moved.

Here, we propose a framework to reliably locate and track people and vehicles using *un-calibrated* cameras, which can have overlapping and/or non-overlapping fields of view. Client-server architecture is used to implement the proposed algorithm. Workstations attached with each camera perform the single camera tracking and send the current trajectories to a central server. Initially, the relationship between the camera

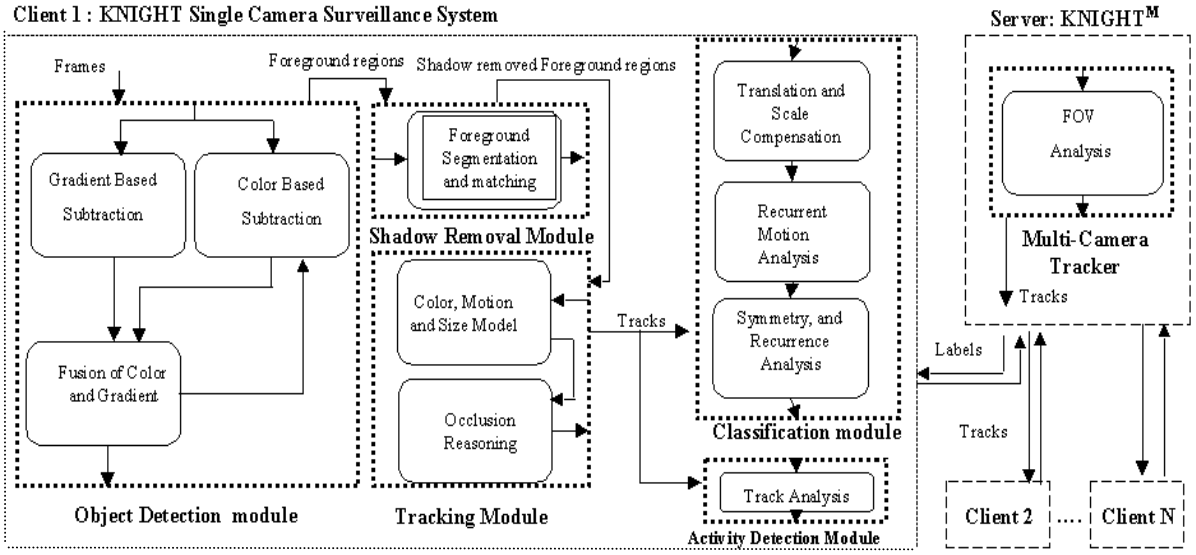

Figure 1: Components of the KNIGHT^M surveillance system.

FOVs is learnt during a training phase, which assumes that the multi-camera correspondences are known. In the case of non-overlapping cameras the FOVs are virtually expanded so that they have an overlap in an extended image coordinate space. In the testing phase, initial detection and tracking is performed at the single camera level. As soon as an object enters the FOV of a camera it's associated client queries the server for the label. The server uses the inter-camera relationships to determine if the object is a new entry into a system or it is already being tracked by another camera. If the object is already being tracked, the server hands over the object label to the client that generated the particular query.

In the next section we discuss the related work. The single camera surveillance system is described in Section 3. The use of FOVs of different cameras to solve the multi-camera tracking problem is discussed in Section 4. Results are given in Section 5. The conclusion is given in Section 6.

2 Related Work

In related work, Collins et. al. [1] have developed a system consisting of multiple calibrated cameras and a site model. The objects were tracked using correlation and 3D location on the site model. Lee et. al. [2] proposed an approach for tracking in cameras with overlapping FOVs that did not require calibration. The camera calibration information was recovered by matching motion trajectories obtained from different views and plane homographies were computed from the most frequent matches. Cai and Aggarwal [3] used calibrated cameras with overlapping FOVs for tracking. The correspondence between objects was established by matching geometric and appearance features.

Huang and Russell [7] used a combination of appearance matching and transition times (across cameras) of objects, in non-overlapping cameras with known topology, to establish correspondence. Kettnaker and Zabih [8] also used the transition times of objects across cameras for tracking. A Bayesian formulation of the problem was used to reconstruct the paths of objects across multiple cameras. The topology of allowable paths of movement was manually given to the system.

In contrast to the above-mentioned work, our proposed method combines tracking across overlapping and non-overlapping cameras in a single framework. Also it does not require manual input of camera topology or calibration information.

3 Single Camera Surveillance

The single camera system consists of the object detection, tracking, object classification and activity recognition components. Each of these modules is explained in detail below. The tracking and classification results obtained by the single camera system are sent to the server for global correspondence. Figure 1 shows the data flow through the various modules of the system.

3.1 Object Detection

Color based background subtraction methods are susceptible to sudden changes in illumination. Gradients of image are relatively less sensitive to changes in illumination and can be combined with color information effectively and efficiently to perform quasi illumination invariant background subtraction. The background differencing [4] performs subtraction at multiple levels. At the pixel level, statistical models of gradients and color are separately used to classify each pixel as belonging to background or foreground. In the second level, foreground pixels obtained from the color-based subtraction are grouped into regions. Each region is tested for the presence of gradient-based foreground pixels at its boundaries. If the region boundary does not have gradient-based foreground then such regions are removed. The pixel level models are updated based on decisions made at the region level. This approach provides the solution to some of the common problems that are not addressed by most background subtraction algorithms such as quick illumination changes due to adverse weather conditions, repositioning of static background objects, and initialization of background model with moving objects present in the scene.

Figure 2: Detection and tracking of two people in down town Orlando.

3.2 Tracking

Each object is modeled by color and spatial pdfs. A Gaussian distribution represents the spatial pdf, with variance equal to the sample variance of the object silhouette. The color pdf is approximated by a normalized histogram. Each pixel in the foreground region votes for the label of an object, for which the product of color and spatial probability is the highest. Each region in the current frame is assigned an object's label if the number of votes from the region's pixels for the object is a significant percentage, say T_p, of all the pixels belonging to that object in the last frame. If two or more objects receive votes, greater than T_p, from a region, it is assumed that multiple objects are undergoing occlusion. The position of a partially occluded object is computed by the mean and variance of pixels that voted for

that particular object. In case of complete occlusion, a linear velocity predictor is used to update the position of the occluded object. This method takes care of both a single object splitting into multiple regions and multiple objects merging into a single region. The spatial and color models are updated for objects that are not undergoing occlusion. Fig. 2 shows detection and tracking results with two people in the scene.

3.3 Object Classification

People undergo a repeated change in shape while walking. Vehicles, on the other hand, are rigid bodies and do not exhibit repeating change in shape while moving. We have developed a specific feature vector called a 'Recurrent Motion Image' (RMI) [5] to calculate repeated motion of objects. The system is able to distinguish between single persons, groups of persons and vehicles using this method.

3.4 Activity Recognition

The system uses a rule-based algorithm to detect both single object activities and multi-object interactions. The recognized single person activities include falling and running. Interactions, for example meetings between persons and placement of a carried object, are also detected. Speed, direction, and orientation of silhouettes and, their inter object distances are used as features to detect the activities. These features are computed from the tracks of each object. The speed information is utilized to detect running. Silhouette orientation is used to detect falling. Inter object distances and the directions are the features exploited for detecting a meeting. Placement of a carried object is recognized in two steps. If a connected region splits into two, and one of the split regions remains motionless for a period of time then this behavior is deemed to be indicative of dropping off a carried object.

Figure 3: Generation of FOV lines. Two correct correspondences can be used to find a line. In the top pair of images, a person is entering or leaving the right camera. The position of this person in the left camera can be used to find the left FOV line of right camera as seen in the left camera.

4 Multi-Camera Surveillance

We assume for multi-camera tracking that all cameras are viewing the same ground plane. In order to track people across cameras we first need to discover the relationship between the FOVs of the cameras. When the tracking is initiated there is no information about the FOV lines of the cameras. The system can, however find this information by observing motion in an environment. This 'training' phase is described in the next subsection.

4.1 Establishing Relationship between FOVs of Cameras

In order to perform consistent labeling across multiple cameras, we need to determine the FOV lines [6] of each camera as viewed in other cameras. These lines are determined

during a training phase in which a single person walks in the environment.

Suppose, without loss of generality, that L_l^{ij} and L_r^{ij} are the projections of the left and right FOV lines of camera C^i on camera C^j such that i,j $\in$ {1…n} , where *n* is the total number of cameras. Suppose, the object being tracked in camera C^j enters or exits camera C^i from its left side. The point in C^j, at which the bottom of object touches the ground plane, actually lies on the projection of FOV of C^i on camera C^j. A least squares method is used to obtain L_l^{ij} from multiple such observations. In case of non-overlapping cameras, suppose an object exits from C^j. We keep on predicting the position of the object for a certain time interval *T*. The prediction is made by using a linear velocity model. If the object enters C^i from the left side, within interval *T*, the predicted position in C^j provides a constraint to determine L_l^{ij}. Basically, we are extending the coordinate space of C^j to obtain a virtual overlap between FOVs of C^i and C^j. Note that all correspondences are known during the training phase since there is a single object in the environment. Thus, by using the above-mentioned method, we can find the relationships between the FOVs of all pairs of cameras in which transition of objects is possible within time *T*. An example of the line generation process is shown in Figure 2.

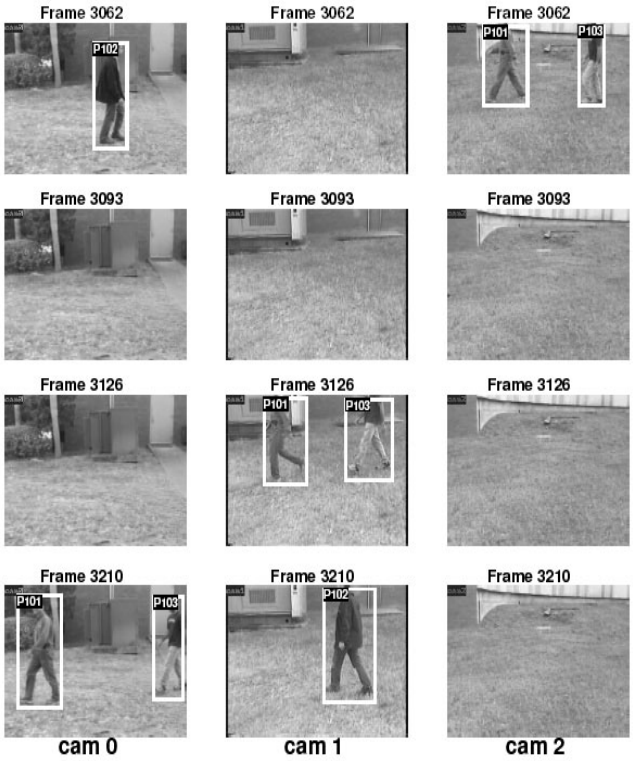

Figure 4: Tracking in multiple non-overlapping cameras. Frame 3062: Three people are seen in the cameras. Frame 3093: No person is seen in any camera as they walk through the non-overlapping area. Frame 3125: Both P101 and P103 are correctly re-assigned to people as they become visible in Cam 1. Frame 3210: Tracking continues with correct correspondences.

4.2 *Establishing Correspondence Across Multiple Cameras*

The correspondence problem occurs when an object enters the FOV of a camera. We need to determine if the object is already being tracked by another camera or it is a new object in the environment. Suppose an object *O* enters camera C^i from the left side. Let ***S*** be the set of the cameras, which contains the projection of left FOV line of C^i. Let L_l^{ij} denotes the projected line in camera $C^j \in \boldsymbol{S}$. Let ***P*** be the set consisting of objects that are currently visible in C^j or have exited the camera within time *T*. For each object $P_k \in \boldsymbol{P}$, where *k* being the object's label, a Euclidean distance, $D(P_k^j, L_l^{ij})$, is computed from line L_l^{ij}. Note that the positions of exited objects are continuously updated by linear velocity prediction. If the object *O* is present in any C^j, its distance from L_l^{ij} should be small. Therefore the object *O* is assigned a label based on following criteria:

$$Label(O) = \arg\min_k D(P_k^j, L_l^{i,j})$$

Finally the object *O* is given a label as described above. Note that the distance from the line only puts a spatial and temporal constraint for label assignment. If an object exits the environment while in the non-overlap area and a new person enters in the same time frame then it will be assigned the

wrong label. To cater for this situation an appearance based distance measure can be combined with the distance function.

5 Results

The single camera tracking and classification approach described in Section 3 was applied to a variety of video sequences. The algorithm was also applied on video sequences provided for the purpose of performance evaluation of tracking in the ``Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2001" [9]. Specifically the video sequences (dataset 1 test sequences 1 and 2) were used. The sequences were jpeg compressed and therefore noisy. The sequences contain persons, groups or people walking together and vehicles. The single camera tracking algorithm successfully handled occlusions between people, between vehicles and between people and vehicles. Another set of sequences was used to test activity recognition. The system was able to detect running, falling and object drop-off if the people performing those actions were not occluded.

To evaluate the performance of proposed multi-camera algorithm, seven sequences were tested with three different camera setups. For the detection of FOV lines in each camera setup, we had a person walked around for short periods. The system was able to compute FOV lines for both overlapping and non-overlapping cases. An example of the computed FOV line is shown in Figure 3. The multiple camera sequences contained both overlapping and non-overlapping views. The algorithm was able to track people correctly in both cases. Results for a non-overlapping camera sequence are shown in Figure 4.

6 Conclusion

We have presented novel algorithms for solving elementary problems of automated surveillance i.e. detection, tracking, activity recognition and classification of objects in single and multiple camera systems. We have combined these algorithms in a real time system and have used it in realistic scenarios for surveillance. The system has been implemented in VC++ using a client server architecture and runs at around 10 frames a second with 3 cameras. More information about the system can be obtained from http://www.cs.ucf.edu/~vision/projects/Knight/KnightM.html.

7 References

[1] R. Collins, A. Lipton, H. Fujiyoshi and T. Kanade, “Algorithms for cooperative multi-sensor surveillance”, *Proceedings of the IEEE*, vol. 89, no. 10, October 2001.

[2] L. Lee, R. Romano, and G. Stein, “Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame”, *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 22, no. 8, August 2000.

[3] Q. Cai and J. K. Aggarwal, “Tracking Human Motion in Structured Environments Using a Distributed-Camera System” *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 2, no.11, November 1999.

[4] O. Javed, K. Shafique and M. Shah, “A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information”, *Proc. IEEE Workshop on Motion andComputing*, December 2002.

[5] O. Javed and M. Shah, “Tracking And Object Classification For Automated Surveillance”, *Proc. European Conference on Computer Vision*, May 2002.

[6] S. Khan, O. Javed, Z. Rasheed and M. Shah, "Human Tracking in Multiple Camera", *Proc. IEEE Computer Vision*, 2001.

[7] T. Huang and S. Russell., "Object Identification in a Bayesian Context", *Proc. IJCAI*, 1997.

[8] V. Kettnaker and R. Zabih, "Bayesian Multi-camera surveillance", *Proc. IEEE Computer Vision and Pattern Recognition*, 1999.

[9] S. Khan, O. Javed, and M. Shah, "Tracking in Un-calibrated Cameras with Overlapping Field of View," *PETS (with IEEE CVPR)*, 2001.