Automated Visual Surveillance in Realistic Scenarios

Mubarak Shah
University of Central Florida

Omar Javed and Khurram Shafique
Object Video

In this article, we present Knight, an automated surveillance system deployed in a variety of real-world scenarios ranging from railway security to law enforcement. We also discuss the challenges of developing surveillance systems, present some solutions implemented in Knight that overcome these challenges, and evaluate Knight's performance in unconstrained environments.

Using video cameras for monitoring and surveillance is common in both federal agencies and private firms. Most current video surveillance systems share one feature: they need a human operator to constantly monitor them. Their effectiveness and response is largely determined not by the technological capabilities but by the vigilance of the person monitoring the camera system.

Furthermore, the number of cameras and the area under surveillance are limited by the personnel available. To overcome these limitations of traditional surveillance methods, a major effort is under way in the computer vision and artificial intelligence community to develop automated systems for the real-time monitoring of people, vehicles, and other objects.[1-3] For a breakdown of the tasks and problems involved, see the sidebar "Surveillance System Tasks and Related Technical Challenges." These systems can create a description of the events happening within their area and generate warnings if they detect a suspicious person or unusual activity.

In this article, we introduce the key logical components of a general automated surveillance system, noting important technical challenges in the real-world deployment of such a system. We specifically discuss Knight, a monitoring system that we've developed and used in a number of surveillance-related scenarios. For an overview of other work in the field, see the "Brief Literature Review" sidebar (page 34).

Knight

Knight is a fully automated, multiple camera surveillance and monitoring system that we developed at the University of Central Florida's Computer Vision Laboratory, which is being used for projects funded by the Florida Department of Transportation, Orlando Police Department, DARPA Small Business Technology Transfer (STTR) program, and Lockheed Martin Corporation. Knight is a commercial, off-the-shelf surveillance system that detects, categorizes, and tracks moving objects in the scene using state-of-the-art computer vision techniques. It also flags significant events and presents a summary in terms of keyframes and a textual description of observed activities to a human operator for final analysis and response decision. Figure 1 shows (page 35) the block diagram of the information flow in Knight. In the following sections, we detail each of Knight's modules.

Object detection

An image sequence's color properties change greatly when illumination varies in the scene, while the gradients are relatively less sensitive to illumination changes. These color and gradient features can be combined effectively and efficiently to perform a quasi-illumination invariant background subtraction.

The object detection algorithm in Knight[4] performs subtraction at multiple levels. At the pixel level, Knight separately uses statistical models of gradients and color to classify each pixel as belonging to the background or foreground.

In the second level, it groups foreground pixels obtained from the color-based subtraction[5] into regions. Each region is tested for the presence of foreground gradients at its boundaries. If the region boundary doesn't overlap with detected foreground gradients, such regions are removed. The pixel-based models are updated based on decisions made at the region level.

The intuition behind this approach is that interesting objects have well-defined boundaries that cause high gradient changes at the object's perimeter with respect to the background model.

1070-986X/07/$25.00 © 2007 IEEE

Surveillance System Tasks and Related Technical Challenges

We can break down the general problem of an automated surveillance system into a series of subproblems. In general, a surveillance system must be able to detect the presence of objects moving in its field of view, track these objects over time, classify them into various categories, and detect some of their activities. It should also be capable of generating a description of the events happening within its field of view (FOV). Each of these tasks poses its own challenges and hurdles for the system designers.

Object detection

The first step toward automated activity monitoring is detecting interesting objects in the camera's FOV. While the definition of an interesting object is context dependent, for a general automated system any independently moving object—such as a person, vehicle, or animal—is deemed interesting. We can achieve object detection by building a representation of the scene called the *background model* and then finding deviations from the model for each incoming frame. Any significant change in an image region from the background model signifies a moving object. The pixels undergoing change are marked for further processing. We use the term *background subtraction* to denote this process. Background subtraction is used as a focus-of-attention method—for example, further processing for tracking and activity recognition is limited to the regions of the image consisting of foreground pixels only. Figure A shows an example of the background subtraction output.

The detection methods based on background subtraction face several problems in accurately detecting objects in realistic environments:

- *Illumination changes*. In outdoor systems, the change in illumination with the time of day alters the appearance of the scene and causes deviation from the background model. This results in a drastic increase in the number of falsely detected foreground regions. This shortcoming makes automated surveillance unreliable under changing illumination conditions.

- *Camouflage*. If an object is similar to the background then it might not be possible to distinguish between the two.

- *Uninteresting moving objects*. Every moving object might not be of interest for monitoring—for example, waving flags, flowing water, or moving leaves of a tree. Classification of such regions as interesting objects can result in false alarms.

- *Shadows*. Objects cast shadows that might also be classified as foreground because of the illumination change in the shadow region.

For a survey of recent developments in the area of background subtraction, see Radke et al.[1]

continued on p. 32

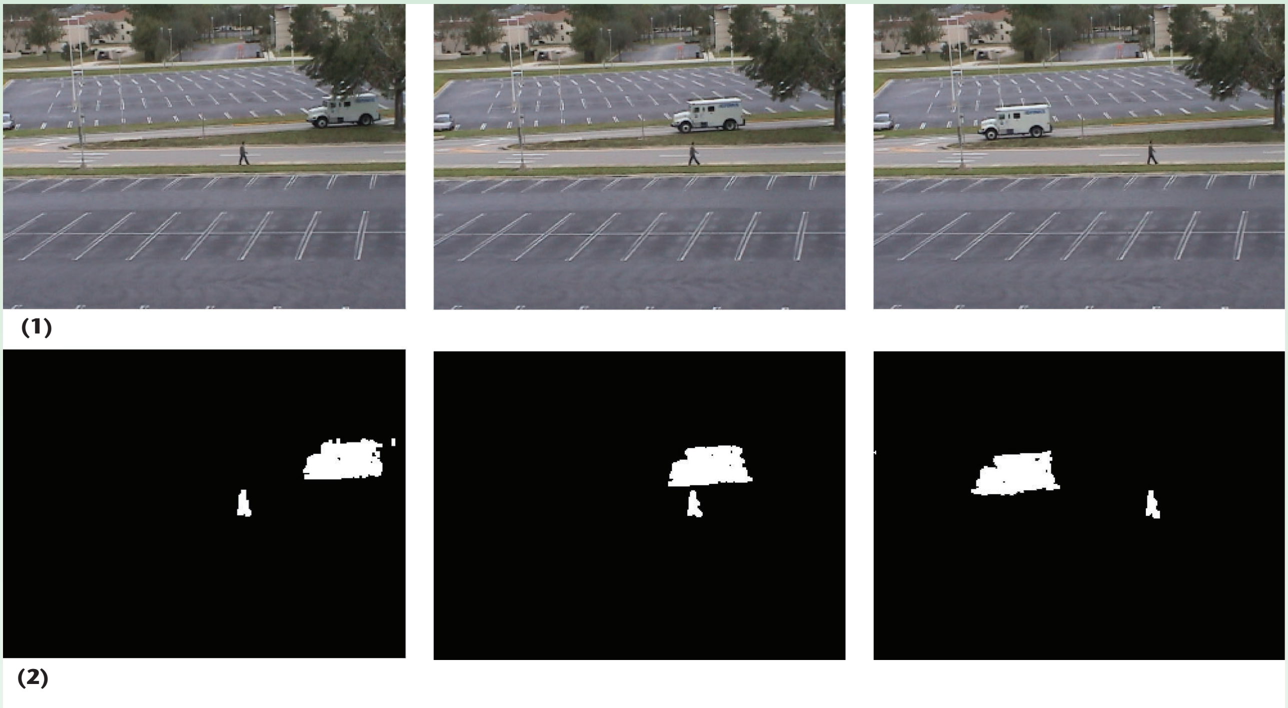

Figure A. (1) Images from a parking lot camera. (2) Output of the background subtraction module.

Surveillance System Tasks and Related Technical Challenges

continued from p. 31

Tracking

Once the system detects the interesting objects, it's useful to have a record of their movement over time. We can define tracking as the problem of estimating the trajectory of an object as the object moves around a scene. Simply stated, we want to know where the object is in the image at each instant in time. If the object is continuously observable and its shape, size, or motion doesn't vary over time, then tracking isn't difficult. However, in realistic environments, like a busy street of a shopping mall, none of these assumptions hold true. Objects—especially people—undergo a change in shape while moving. In addition, their motion isn't constant.

Objects also undergo occlusion—that is, the view of one object is blocked by another object or structure. Occlusion leads to discontinuity in the observation of objects, and it's one of the major issues that a tracking algorithm must solve. Tracking objects under occlusion is difficult because we can't determine the accurate position and velocity of an occluded object. Two major cases of occlusion can be described as follows:

- Interobject occlusion occurs when one object blocks the view of other objects in the camera's FOV. The background subtraction method outputs a single region for occluding objects. If two initially nonoccluding objects cause occlusion then this condition can be easily detected. However, if objects enter the scene while occluding each other then it's difficult to determine if interobject occlusion is occurring. The problem is identifying that the foreground region contains multiple objects and to determine the location of each object in the region. Because people usually move in groups, resulting in frequent interobject occlusion, detecting and resolving this is important for surveillance applications.

- Occlusion of objects due to scene structures causes an object to disappear for a certain amount of time, leaving no foreground region to represent the object during the occlusion. For example, a person walks behind a building, or a person enters a car. A decision must be made to either wait for the object's reappearance, or to conclude that the object has exited the scene.

In addition to the imperfection of input data, tracking methods also have to deal with imperfections in the output of detection methods. Detection methods aren't perfect and are susceptible to miss the detection of interesting objects or to detect uninteresting objects. All these difficulties make tracking in realistic scenarios a difficult problem for automated surveillance systems. See elsewhere[2] for a review on algorithms for human motion tracking.

Object categorization

To make an intelligent analysis of the scene and to recognize various activities, surveillance systems need to perform a variety of object categorization tasks. In urban monitoring systems, this task can involve labeling the detected object as a person, a group of persons, a vehicle, an animal, a bicycle, and so on.

On the other hand, foreground regions generated due to local illumination changes or shadows have diffused boundaries resulting in minor variation in the boundary gradients relative to the background. Thus, errors in the color-based subtraction can be removed by using the gradient information at the region boundaries.

This method has the ability to deal with some of the common problems not addressed by most background subtraction algorithms such as quick illumination changes because of adverse weather conditions, the repositioning of static background objects, and initializating the background model with moving objects present in the scene.

Tracking

The output of the background subtraction method (object detection module) for each frame is a binary image composed of foreground regions. Each region is a set of connected pixels in the binary image. Note that there's not necessarily a one-to-one correspondence between foreground regions and actual objects in the scene. In case of occlusion, multiple objects can merge into the same region. Also, similarity in color between an object and the background can result in splitting that object's silhouette into multiple regions. Therefore, this requires an object model that can tolerate these split-and-merge cases.

Knight models an object using a combination of its color, shape, and motion models.[6] A Gaussian distribution represents the spatial model. The color model is a probability density function (PDF) that a normalized histogram approximates.

Each pixel in the foreground region votes for an object's label, 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

In military scenarios this task may be more refined, such as identifying which type of vehicle is detected (for example, whether it's a tank or a Humvee). Ideally, we can attempt object classification by using shape information from a single image. However, the work on object recognition in the past 30 years has demonstrated that object recognition or categorization from a single image is a highly complex task. The requirement to classify objects in real time makes the categorization task even more difficult.

Tracking across cameras

In general, surveillance systems are required to observe large areas like airport perimeters, naval ports, or shopping malls. In these scenarios, it isn't possible for a single camera to observe the complete area of interest because sensor resolution is finite and structures in the scene limit the visible areas. Therefore, surveillance of wide areas requires a system with the ability to track objects while observing them through multiple cameras (see Figure B).

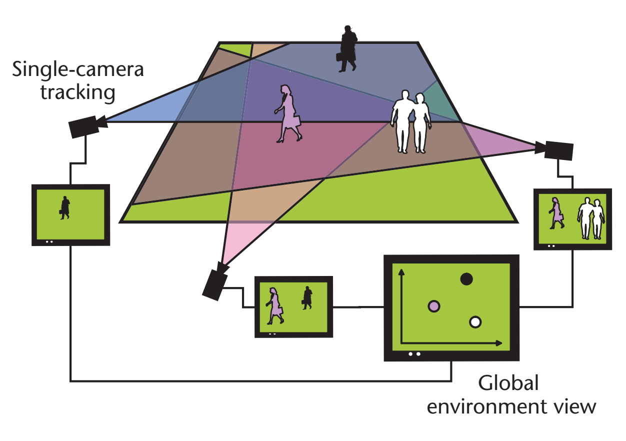

Figure B. A distributed surveillance system. Inputs from each camera are integrated at the server level to determine a central view of the environment.

It's better if the tracking approach doesn't require camera calibration or complete site modeling, since the luxury of calibrated cameras or site models isn't available in most situations. Also, maintaining calibration between a large network of sensors is a daunting task, since a slight change in the position of a sensor will require the calibration process to be repeated. Thus, it's better if the system can learn the intercamera geometry and the scene model directly from the environment. In addition, the system should be adaptive to small changes in the geometry or scene so that they don't have a detrimental effect on its performance. For the system to be effective in realistic scenarios, it should also be easy to deploy without extensive manual work.

References

1. R.J. Radke, O. Al-Kofahi, and B. Roysam, "Image Change Detection Algorithms: A Systematic Survey," *IEEE Trans. Image Processing*, vol. 14, no. 3, 2005, pp. 294-307.
2. J.K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 3, 1999, pp. 428-440.

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, we can assume 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 occluded object's position. 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 aren't undergoing occlusion. Figure 2 (page 35) shows the result of tracking under occlusion.

Object categorization

Knight classifies objects into three classes: people, groups of people, and vehicles. Instead of relying on the objects' spatial primitives, it uses a motion-based approach and exploits the fact that people undergo a repeated change in shape while walking, whereas vehicles are rigid bodies and don't exhibit repeating change in shape while moving. The solution is based on temporal templates that are called recurrent motion images (RMIs)[7] and are used to represent the repeated motion of objects.

An RMI is a template that has high values at the pixels where motion occurs repeatedly and low values at pixels where there's little or no recurring motion. The RMI is computed by aligning and accumulating the foreground regions obtained by the object detection module. Therefore, it's not affected by small changes in lighting or background clutter. One major advantage of using RMIs is that the system doesn't have to explicitly store the history of previous observations, making the computation both fast and memory efficient.

Brief Literature Review

Here, we present a brief review of some of the automated surveillance systems proposed in recent years. Interested readers are referred elsewhere[1-4] for detailed surveys.

Among the earlier automated monitoring systems, Pfinder[5] is perhaps the most well known. It tracks the full body of a person in the scene that contains only one unoccluded person in the upright posture. It uses a unimodal background model to locate the moving person.

In Rehg et al.,[6] a smart kiosk is proposed that can detect and track moving people in front of a kiosk by using face detection, color, and stereo. Stauffer and Grimson[7] used an adaptive multimodal background subtraction method for object detection that can deal with slow changes in illumination, repeated motion from background clutter, and long-term scene changes. They also proposed detection of unusual activities by statistically learning the common patterns of activities over time. They tracked detected objects using a multiple hypothesis tracker.

Ricquebourg and Bouthemy[8] proposed tracking people by exploiting spatiotemporal slices. Their detection scheme involves the combined use of intensity, temporal differences between three successive images, and comparing the current image to a background reference image, which is reconstructed and updated online. Boult et al. presented a system for monitoring uncooperative and camouflaged targets.[9]

The W4[10] uses dynamic appearance models to track people. Single persons and groups are distinguished using projection histograms, and each person in a group is tracked by tracking the head of that person. Lipton et al.[11] developed a system to detect and track multiple people and vehicles in a cluttered scene and monitor activities over a large area and extended periods of time. Their system could also classify objects as a person, group of persons, vehicles, and so on, using shape and color information.

References

1. J.K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 3, 1999, pp. 428-440.
2. R. Collins, A. Lipton, and T. Kanade, "Introduction to the Special Section on Video Surveillance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000, p. 745.
3. W. Hu et al., "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, 2004, pp. 334-352.
4. C. Regazzoni, V. Ramesh, and G. Foresti, "Scanning the Issue/Technology," *Proc. IEEE*, IEEE Press, vol. 89, no. 10, p. 1355.
5. C. Wren et al., "Pfinder, Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785.
6. J. Rehg, M. Loughlin, and K. Waters, "Vision for a Smart Kiosk," *Computer Vision and Pattern Recognition*, IEEE Press, 1997, pp. 690-696.
7. C. Stauffer and W.E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000, pp. 747-757.
8. Y. Ricquebourg and P. Bouthemy, "Real-Time Tracking of Moving Persons by Exploiting Spatiotemporal Image Slices," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000, pp. 797-808.
9. T. Boult et al., "Into the Woods: Visual Surveillance of Noncooperative and Camouflaged Targets in Complex Outdoor Settings," *Proc. IEEE*, IEEE Press, vol. 89, no. 10, 2001, pp. 1382-1402.
10. I. Haritaoglu, D. Harwood, and L.S. Davis, "W4: Real-Time Surveillance of People and Their Activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000, pp. 809-830.
11. H. Fujiyoshi et al., "Algorithms for Cooperative Multisensor Surveillance," *Proc. IEEE*, IEEE Press, vol. 89, no. 10, 2001, pp. 1456-1477.

Tracking across cameras

Knight is capable of seamlessly tracking objects across multiple cameras.[8] It doesn't require the cameras to be calibrated nor does it require scene geometry as input. Rather, it uses the observations of people through the system of cameras to discover relationships between the cameras. We observe that people or vehicles tend to follow the same paths in most cases, such as roads, walkways, and corridors. Our tracking algorithm uses this conformity in traversed paths to establish correspondence.

For example, consider the scenario of Figure 3b (page 36) and suppose people moving along one direction of the walkway initially observed in camera 2 are also observed entering camera 3's field of view (FOV) after a certain time interval. The people can take many paths across cameras 2 and 3. However, because of physical and practical constraints, some of the paths will be more likely to be taken by people than others.

For example, it's more likely for a person exiting camera 2's FOV from point A to enter camera 3's FOV at point B rather than entering camera 3's FOV at point D. Thus, we can use the usual locations of exits and entrances between cameras, the direction of movement, and average time taken to reach one camera from another, to constrain correspondences. Knight exploits these space and time cues to learn the intercamera relationships. These relationships are learned in the form of PDFs. We use the Parzen window[9] tech-

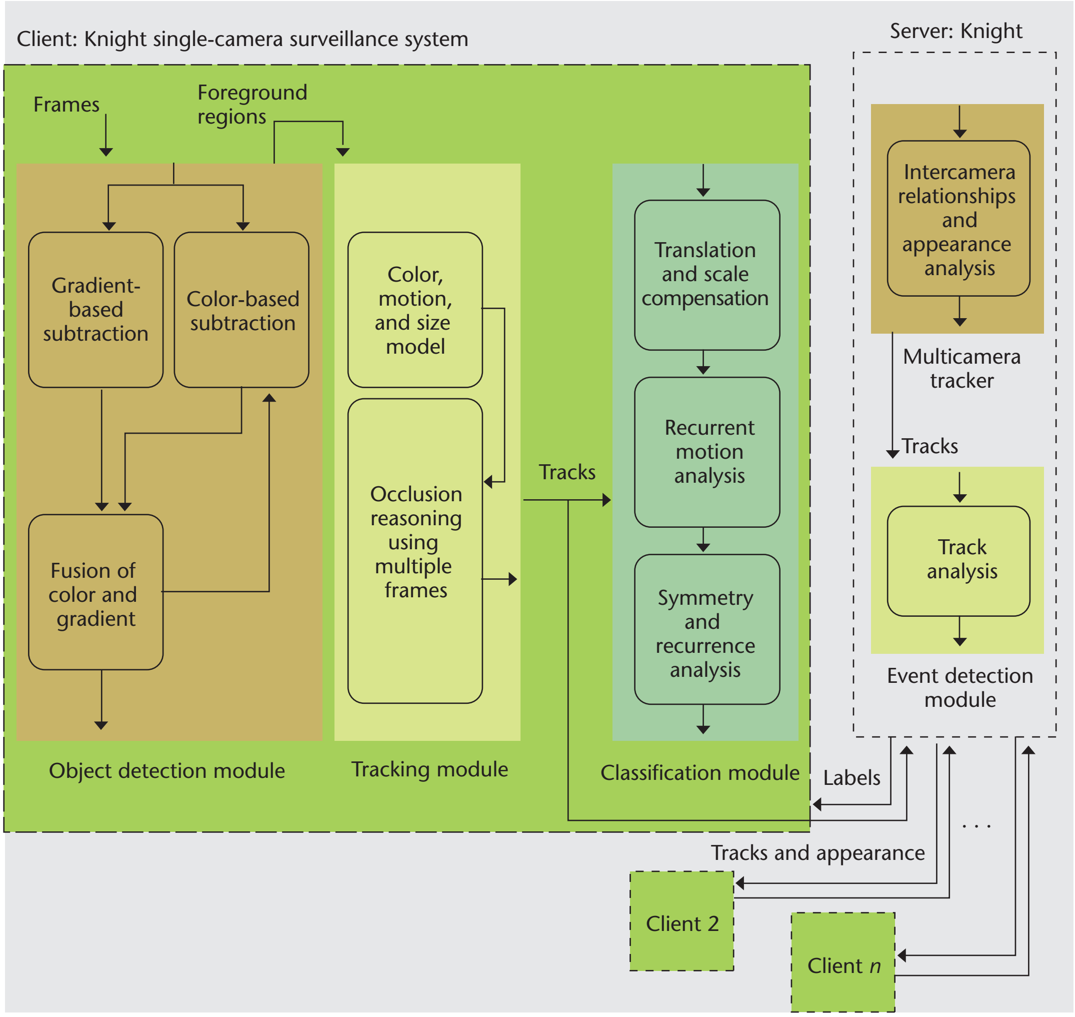
Client: Knight single-camera surveillance system
Server: Knight
Frames
Foreground regions
Gradient-based subtraction
Color-based subtraction
Fusion of color and gradient
Color, motion, and size model
Occlusion reasoning using multiple frames
Tracks
Translation and scale compensation
Recurrent motion analysis
Symmetry and recurrence analysis
Intercamera relationships and appearance analysis
Multicamera tracker
Tracks
Track analysis
Event detection module
Object detection module
Tracking module
Classification module
Labels
. . .
Tracks and appearance
Client 2
Client *n*

Figure 1. Local components of the Knight multicamera surveillance system.

nique to estimate the space and time between each pair of cameras. Formally, suppose we have a sample *S* consisting of *n*, *d* dimensional, data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ from a multivariate distribution $p(\mathbf{x})$, then we can calculate an estimate $\hat{p}(\mathbf{x})$ of the density at **x** using

$$\hat{p}(\mathbf{x}) = \frac{1}{\mathbf{n}} |\mathbf{H}|^{-\frac{1}{2}} \sum_{\mathbf{i}=1}^{\mathbf{n}} \kappa\left(\mathbf{H}^{-\frac{1}{2}}(\mathbf{x}-\mathbf{x}_{\mathbf{i}})\right) \qquad (1)$$

where the *d* variate kernel $\kappa(\mathbf{x})$ is a bounded function satisfying integral $\int \kappa(\mathbf{x})d\mathbf{x} = 1$, and **H** is the symmetric $d \times d$ bandwidth matrix. The position–time feature vector **x**—used for learning the space–time PDFs from camera C_i to C_j— consists of the exit location and entry locations in cameras, indices of entry and exit cameras, exit velocities, and the time interval between exit and entry events. The system uses the space–time PDFs to obtain an object's probabil-

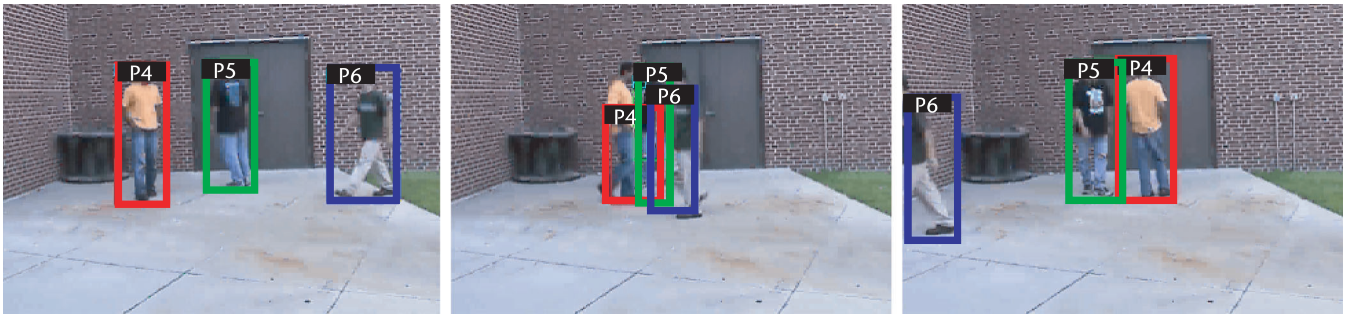

Figure 2. Tracking results in the presence of occlusion.

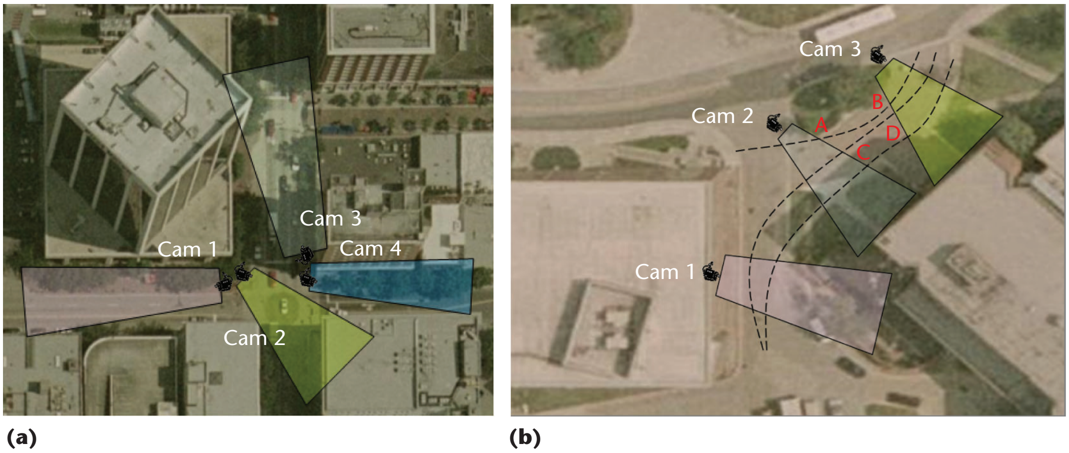

Figure 3. (a) An overhead view of the fields of view (FOV) of the cameras installed in downtown Orlando for a real-world evaluation of our proposed algorithms. (b) A multiple-camera setup at the University of Central Florida (top view) showing expected paths of people through the multicamera system. We can use these paths to find relationships between the cameras' FOV.

ity of entering a certain camera at a certain time given the location, time, and velocity of its exit from other cameras.

With this in mind, we used some of the following key observations when we modeled the Knight system:

- dependence of the intercamera travel time on the magnitude and direction of the object's motion;
- dependence of the intercamera travel time interval on the location of the exit from one camera and location of entrance in the other; and
- correlation among the locations of exits and entrances in cameras.

The reason for using the Parzen window approach for estimation is that, rather than imposing assumptions, the nonparametric technique lets us directly approximate the *d* dimensional density describing the intercamera relationships. We can use these space–time probabilities together with object appearance histograms to track objects across cameras.

Knight in action

Knight has been actively used in several surveillance-related projects funded by different government and private agencies. We implemented Knight in Visual C++ and it's capable of running on both PCs and laptops. The single-camera system operates at 15 Hz on a Pentium 2.0-GHz machine with 512 Mbytes of RAM and takes video input from any video camera capable of transmitting video through an IEEE 1394 FireWire cable. In the case of multiple camera networks, we can also use a server machine to maintain the consistent identities of objects over the network and to handle the hand-off between cameras.

Knight is currently being deployed for a project funded by the Florida Department of Transportation (FDOT) to monitor the railroad grade crossings, prevent accidents involving trains, and automatically inform the authorities of any potential hazard[10]—for example, the presence of a person or a vehicle on tracks while a train is approaching. Approximately 261,000 highway-rail and pedestrian crossings exist in the United States according to the studies by the National Highway Traffic Safety Administration and Federal Railroad Administration (FRA). According to the FRA's Railroad Safety report (see http://safetydata.fra.dot.gov/officeofsafety), from 1998 to 2004 there were 21,952 highway-rail crossing incidents involving motor vehicles—averaging 3,136 incidents a year. In Florida alone, there were 650 highway-rail grade crossing incidents, resulting in 98 fatalities during this period. Thus, there is a significant need for innovative technologies that can monitor railroad grade crossings.

In addition to the standard functionality of a surveillance system, the system deployed for FDOT lets users crop a zone (shown by the yellow bounding boxes in Figure 3) in the image corresponding to a specific location in the scene. This zone (called the danger zone) is usually the area in the scene of interest, where the presence of a person or vehicle can be hazardous in case a train is approaching.

The system receives two inputs—one from the traffic signal (triggered when a train approaches) and the other from the detection module giving the position of pedestrians and vehicles with respect to the danger zone. A simple rule-based algorithm recognizes activities based on the object detection and track patterns. At the onset of an undesirable event, such as the presence of a person or vehicle on or near the tracks while a train is approaching, the system generates an audio alert and an email is sent to an authorized individual through a wireless network. The system also has the ability to detect the presence of a train in the video using the motion information in a designated area.

We evaluated the performance of different system modules (detection, tracking, and classification) by manually determining the ground truth from 6 hours of videos and comparing the ground truth to the results of the automated system. We set up the Knight system at two different highway railroad crossings in central Florida and collected a total of five videos from different views and in different conditions—for example, time of day, lighting, wind, camera focus, and traffic density. The collection of videos under different weather conditions—such as sunny, overcast, and partly cloudy—ensured that the system was tested under different as well as constantly changing illumination conditions.

Note that the system only works during the day and turns off automatically when the illumination is below a certain predefined level; hence, we only performed testing during the day. The system also doesn't perform during rain and storms, so these weather conditions weren't considered during testing.

We measured the accuracy of Knight's object detection as the ratio of the number of correct detections and the total number of objects. The system correctly detected 706 objects out of 725 and it generated 23 false positives during this period. This amounts to 97.4 percent recall and 96.8 percent detection precision.

We defined the accuracy of tracking as the ratio of the number of completely correct tracks and the number of correct detections. We found that it tracked 96.7 percent of the objects accurately over the complete period of their presence in the FOV.

Similarly, we measured the classification accuracy as the ratio of the number of correct classifications and the number of correct detections, which we found to be 88 percent. Figure 4 graphically depicts the performance of each module. The system's performance under different scenarios justifies our claims that the system is robust to changes in environmental conditions.

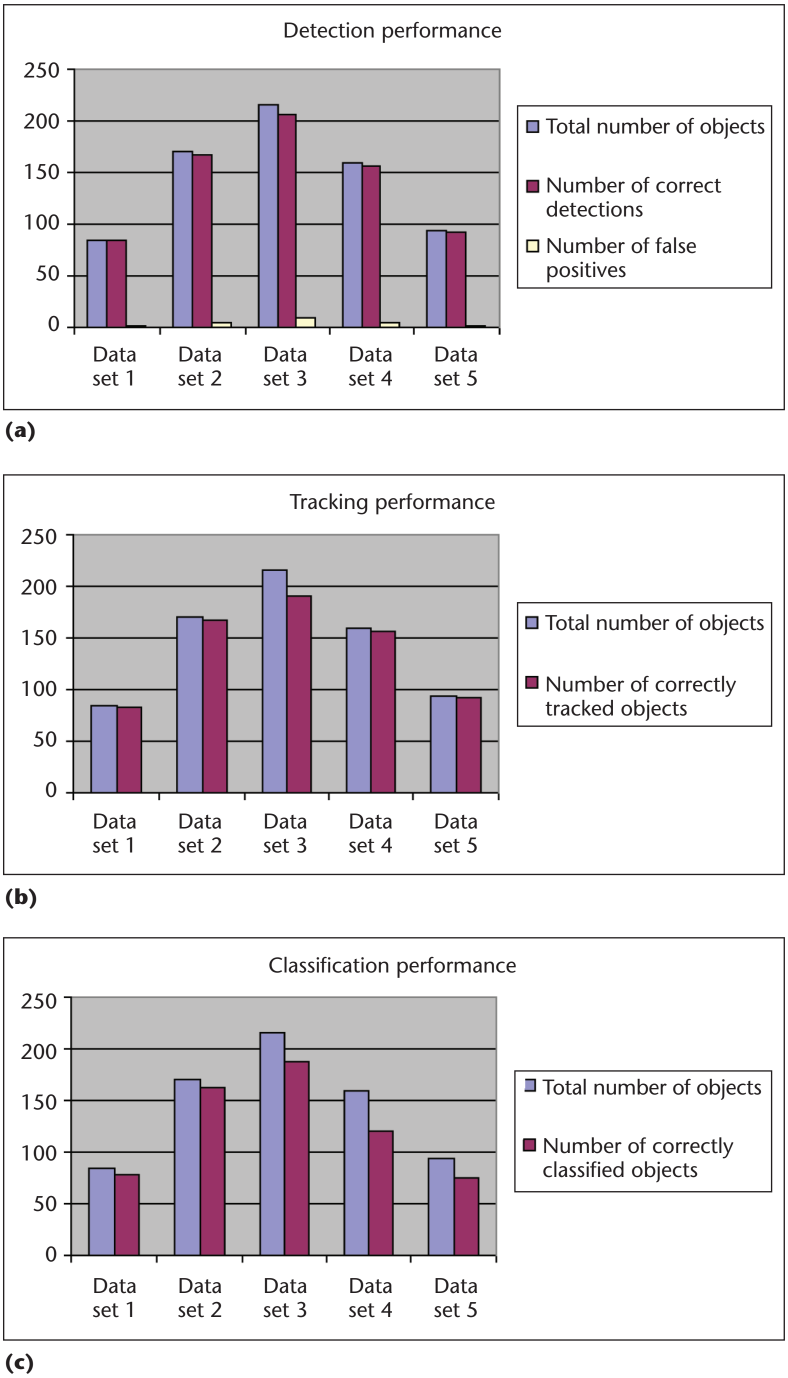

Figure 4. Performance plots. (a) Detection performance. (b) Tracking performance. (c) Classification performance.

Most of the object detection errors were caused by interobject occlusion or objects having a similar color to the background. The tracking errors were caused by multiple people with similarly colored clothes walking close to each other. In such cases, our statistical models of object appearance and location weren't able to distinguish between the different objects. Note that even if the objects were assigned incorrect labels because of a tracking error, the trespass warning was still correctly generated if these objects were

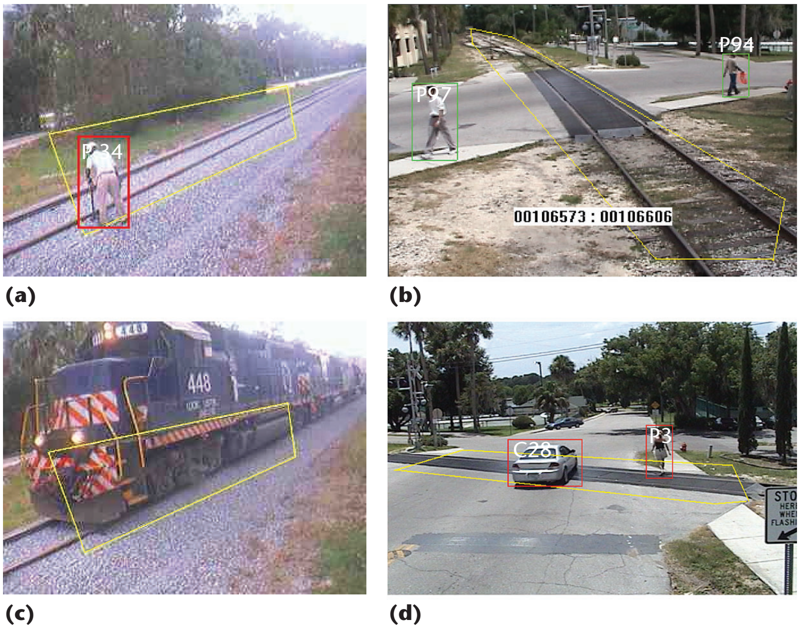

Figure 5. Knight at railroads. (a) A danger zone is defined manually through a GUI. (b) An alarm is generated if people or vehicles—but not (c) trains—enter the danger zone. Moreover, the bounding box of individuals (d) turns red as they enter the polygon defining the danger zone.

detected successfully in the danger zone by the background subtraction module.

We also tested the performance of the proposed intrusion detection algorithm. We achieved this by first defining a danger zone in the image (shown as a yellow bounding box in Figure 5) and by letting the system run over a period of seven days (only during the day). Figure 5 shows two different testing sites in central Florida, along with persons, vehicles, and a train detected by the system. A red bounding box around an object signifies that the object is in the danger zone. The danger zone is marked by a yellow polygon. Overall, the system detected a number of trespassing violations over its running period. When compared to the ground truth (obtained manually by browsing through the archived videos), the system produced no errors during this extended period of time.

We're also using the Knight system in a number of other surveillance-related projects. Recently, we augmented Knight to help the Orlando police department with automated surveillance and installed it at four locations in the downtown Orlando area.

We designed the system to provide automatic notification to a monitoring officer in case of unusual activities, such as a person falling, one or more people running, and unattended objects. Figure 6 shows the cameras at downtown Orlando and the FOVs of all four cameras. The capabilities of Knight were enhanced with a correlation-based automatic target recognition algorithm to classify the objects into finer categories for a project funded by Lockheed Martin Corporation.[11] A modified version of Knight is also the workhorse for a joint project with Perceptek on nighttime surveillance (using infrared cameras), funded by DARPA.

Conclusions and future work

We've shown that Knight can detect and classify targets and seamlessly track them across multiple cameras. It also generates a summary in terms of keyframes and the textual description of trajectories to a monitoring officer for final analysis and response decision. This level of interpretation was the goal of our research effort, and we believe that it's a significant step forward in the development of intelligent systems that can deal with the complexities of real-world scenarios.

Current system limitations include the inability to detect camouflaged objects, handling large crowds, and operating in rain and extreme weather conditions. For the latest results and information on Knight, visit http://www.cs.ucf.edu/~vision/projects/Knight/Knight.html. **MM**

References

1. R. Collins, A. Lipton, and T. Kanade, "Introduction to the Special Section on Video Surveillance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000, p. 745.
2. W. Hu et al., "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, 2004, pp. 334-352.
3. C. Regazzoni, V. Ramesh, and G. Foresti, "Scanning the Issue/Technology," *Proc. IEEE*, IEEE Press, vol. 89, no. 10, p. 1355.
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 and Video Computing*, IEEE CS Press, 2002, pp. 22-27.
5. C. Stauffer and W.E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000, p. 747.
6. O. Javed et al., "Knight-m: A Real-Time Surveillance System for Multiple Overlapping and Non-Overlapping Cameras," *Proc. IEEE Conf. Multimedia and Expo*, IEEE CS Press, 2003, pp. 649-652.
7. O. Javed and M. Shah, "Tracking and Object Classification for Automated Surveillance," *Proc. 7th European Conf. Computer Vision*, Springer, 2002, p. 343.
8. O. Javed et al., "Tracking across Multiple Cameras with Disjoint Views," *Proc. 9th IEEE Int'l Conf.*

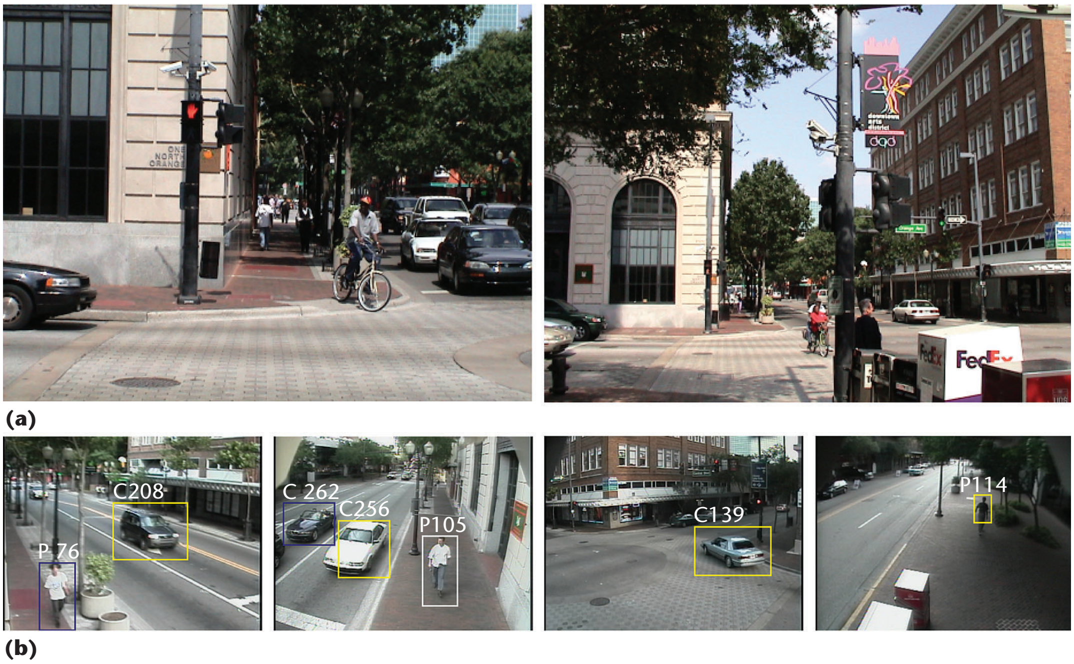

Figure 6. (a) The first row shows the cameras installed in downtown Orlando. (b) The second row shows the fields of view of all four cameras. The objects that are being detected, tracked, and classified are shown in bounding boxes.

Computer Vision, Springer-Verlag, 2003, pp. 343-357.

9. R.O. Duda, et al., *Pattern Classification*, Wiley, 2000.
10. Y. Sheikh et al., "Visual Monitoring of Railroad Grade Crossing," *SPIE Defense and Security Symp.*, SPIE Press, 2004, pp. 654-660.
11. A. Mahalanobis et al., "Network Video Image Processing for Security, Surveillance, and Situational Awareness," *Proc. SPIE Defense and Security Symp.*, SPIE Press, 2004, pp. 1-8.

Mubarak Shah, the Agere Chair Professor of Computer Science and the founding director of the Computer Vision Laboratory at University of Central Florida, is a researcher in computer vision. He has worked in several areas including activity and gesture recognition, violence detection, event ontology, object tracking, video segmentation, story and scene segmentation, view morphing, automatic target recognition, wide-baseline matching, and video registration. Shah received his BE in electronics from the Dawood College of Engineering and Technology in Karachi, Pakistan. He completed his EDE diploma at the Philips International Institute of Technology in Eindhoven, the Netherlands. He then received his MS and PhD degrees in computer engineering from Wayne State University. He's a fellow of IEEE and a recipient of the following awards: the Harris Corporation Engineering Achievement Award, the IEEE Outstanding Engineering Educator Award, and three Tokten Awards from the United Nations Development Program in 1995, 1997, and 2000.

Omar Javed is a research scientist in the Center for Video Understanding Excellence at ObjectVideo. His research interests include wide area surveillance, tracking using a forest of sensors, video compression, multimedia content extraction, and semisupervised classification. Javed earned his PhD in computer science from the University of Central Florida.

Khurram Shafique is a research scientist in the Center for Video Understanding Excellence at ObjectVideo. Shafique received a PhD in computer science from the University of Central Florida. His research interests include real-time surveillance systems, tracking in single and multiple cameras, point correspondence, graph partitioning, and data clustering.

Readers may contact Omar Javed at ojaved@cs.ucf.edu.

For further information on this or any other computing topic, please visit our Digital Library at http://www.computer.org/publications/dlib.