A System for Video Surveillance and Monitoring *

Robert T. Collins, Alan J. Lipton and Takeo Kanade
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA
E-MAIL: {rcollins,ajl,kanade}@cs.cmu.edu PHONE: 412-268-1450
HOMEPAGE: http://www.cs.cmu.edu/~vsam

Abstract

The Robotics Institute at Carnegie Mellon University (CMU) and the Sarnoff Corporation are developing a system for autonomous Video Surveillance and Monitoring. The technical objective is to use multiple, cooperative video sensors to provide continuous coverage of people and vehicles in cluttered environments. This paper presents an overview of the system and significant results achieved to date.

1 Introduction

The DARPA Image Understanding (IU) program is funding basic research in the area of Video Surveillance and Monitoring (VSAM) to provide battlefield awareness. The thrust of CMU's VSAM research is to develop automated video understanding algorithms that allow a network of active video sensors to automatically monitor objects and events within a complex, urban environment. We have developed video understanding technology that can automatically detect and track multiple people and vehicles within cluttered scenes, and to monitor their activities over long periods of time. Human and vehicle targets are seamlessly tracked through the environment using a network of active sensors to cooperatively track targets over areas that cannot be viewed continuously by a single sensor alone. Each sensor transmits symbolic events and representative imagery back to a central operator control station, which provides a visual summary of activities detected over a broad area. The user interacts with the system using an intuitive map-based interface. For example, the user can specify that objects entering a region of interest should trigger an alert, relieving the burden of continually watching that area. The system automatically allocates sensors to optimize system performance while fulfilling user commands.

Although developed within a context of providing battlefield awareness, we believe this technology has great potential for applications in remote monitoring of nuclear facilities. Sample tasks that could be automated are verification that routine maintainance activities are being performed according to schedule, logging and tracking visitors and personnel as they enter and move through the site, and providing security against unauthorized intrusion. Other applications in military and law enforcement scenarios include providing perimeter security for troops, monitoring peace treaties or refugee movements using unmanned air vehicles, providing security for embassies or airports, and staking out suspected drug or terrorist hide-outs by collecting time-stamped pictures of everyone entering and exiting the building.

The following sections present an overview of video surveillance algorithms developed at CMU over the last two years (Section 2) and their incorporation into a prototype system for remote surveillance and monitoring (Section 3).

*This work is funded by the DARPA IU program under VSAM contract number DAAB07-97-C-J031.

2 Video Understanding Technologies

Keeping track of people, vehicles, and their interactions in a complex environment is a difficult task. The role of VSAM video understanding technology in achieving this goal is to automatically "parse" people and vehicles from raw video, determine their geolocations, and automatically insert them into a dynamic scene visualization. We have developed robust routines for detecting moving objects (Section 2.1) and tracking them through a video sequence (Section 2.2) using a combination of temporal differencing and template tracking. Detected objects are classified into semantic categories such as human, human group, car, and truck using shape and color analysis, and these labels are used to improve tracking using temporal consistency constraints (Section 2.3). Further classification of human activity, such as walking and running, has also been achieved (Section 2.4). Geolocations of labeled entities are determined from their image coordinates using either wide-baseline stereo from two or more overlapping camera views, or intersection of viewing rays with a terrain model from monocular views (Section 2.5). The computed geolocations are used to provide higher-level tracking capabilities, such as tasking multiple sensors with variable pan, tilt and zoom to cooperatively track an object through the scene (Section 2.6).

2.1 Moving Target Detection

The initial stage of the surveillance problem is the extraction of moving targets from a video stream. There are three conventional approaches to automated moving target detection: temporal differencing (two-frame or three-frame) [Anderson *et al.*, 1985]; background subtraction [Haritaoglu *et al.*, 1998, Wren *et al.*, 1997]; and optical flow (see [Barron *et al.*, 1994] for an excellent discussion). Temporal differencing is very adaptive to dynamic environments, but generally does a poor job of extracting all relevant feature pixels. Background subtraction provides the most complete feature data, but is extremely sensitive to dynamic scene changes due to lighting and extraneous events. Optical flow can be used to detect independently moving targets in the presence of camera motion; however, most optical flow computation methods are very complex and are inapplicable to real-time algorithms without specialized hardware.

The approach presented here is similar to that taken in [Grimson and Viola, 1997], and is an attempt to make background subtraction more robust to environmental dynamism. The key idea is to maintain an evolving statistical model of the background to provide a mechanism that adapts to slow changes in the environment. For each pixel value p_n in the n^{th} frame, a running average $\overline{p}_n$ and a form of standard deviation σ_{p_n} are maintained by temporal filtering, implemented as:

$$\begin{aligned} \overline{p}_{n+1} &= \alpha p_{n+1} + (1-\alpha)\overline{p}_n \\ \overline{\sigma}_{n+1} &= \alpha|p_{n+1} - \overline{p}_{n+1}| + (1-\alpha)\overline{\sigma}_n \end{aligned} \tag{1}$$

where $\alpha = \tau \times f$, f is the frame rate and τ is a time constant specifying how fast (responsive) the background model should be to intensity changes. the influence of old observations decays exponentially over time, and thus the background gradually adapts to reflect current environmental conditions.

If a pixel has a value which is more than 2σ from $\overline{p}_n$, then it is considered a foreground pixel. At this point a multiple hypothesis approach is used for determining its behavior. A new set of statistics $(\overline{p}', \sigma')$ is initialized for this pixel and the original set is remembered. If, after time $t = 3\tau$, the pixel value has not returned to its original statistical value, the new statistics are chosen as replacements for the old.

Foreground (moving) pixels are aggregated using a connected component approach so that individual target "blobs" can be extracted. Transient moving objects cause short term changes to the image stream that are not included in the background model, but are continually tracked, whereas more permanent changes are (after a time increment of 3τ has elapsed) absorbed into the background (see Figure 1).

The moving target detection algorithm described above is prone to three types of error: incomplete extraction of

(A) (B)

Figure 1: Example of moving target detection by dynamic background subtraction.

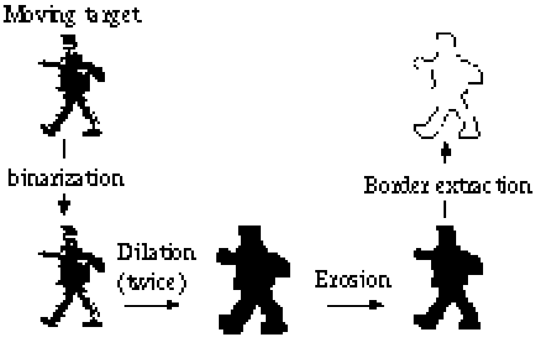


Figure 2: Target pre-processing. A moving target region is morphologically dilated (twice), eroded and then its border is extracted.

a moving object; erroneous extraction of non-moving pixels; and legitimate extraction of illegitimate targets (such as trees blowing in the wind). Incomplete targets are partially reconstructed by blob clustering and morphological dilation (Figure 2). Erroneously extracted "noise" is removed using a size filter whereby blobs below a certain critical size are ignored. Illegitimate targets must be removed by other means such as temporal consistency and domain knowledge. This is the purview of the target tracking algorithm.

2.2 Target Tracking

To begin to build a temporal model of activity, individual objects must be tracked over time. The first step in this process is to take the blobs generated by motion detection and match them between frames of a video sequence.

Many systems for target tracking are based on Kalman filters. However, as Isard and Blake point out, these are of limited use because they are based on unimodal Gaussian densities that cannot support simultaneous alternative motion hypotheses [Isard and Blake, 1996]. Isard and Blake present a new stochastic algorithm called CONDENSATION that does handle alternative hypotheses. Work on the problem of multiple data association in radar tracking contexts is also relevant [Bar-Shalom and Fortmann, 1988].

We employ a much simpler approach based on a frame-to-frame matching cost function. A record of each blob is kept with the following information:

- image trajectory (position $p(t)$ and velocity $v(t)$ as functions of time) of the object centroid,
- blob "appearance" in the form of an image template,

- blob size s in pixels,
- color histogram h of the blob.

The position and velocity of each blob T_i is determined from the last time step t_{last} and used to predict a new image position at the current time t_{now}:

$$p_i(t_{now}) \approx p_i(t_{last}) + v_i(t_{last}) \times (t_{now} - t_{last}) \quad (2)$$

Using this information a matching cost is determined between a known target T_i and a candidate moving blob R_j

$$C(T_i, R_j) = f(|p_i - p_j|, |s_i - s_j|, |h_i - h_j|). \quad (3)$$

Targets that are "close enough" in cost space are considered to be potential matches. To lend more robustness to changes in appearance and occlusions, the full tracking algorithm uses a combination of cost and adaptive template matching, as described in detail in [Lipton *et al.*, 1998]. Recent results from the system are shown in Figure 3.

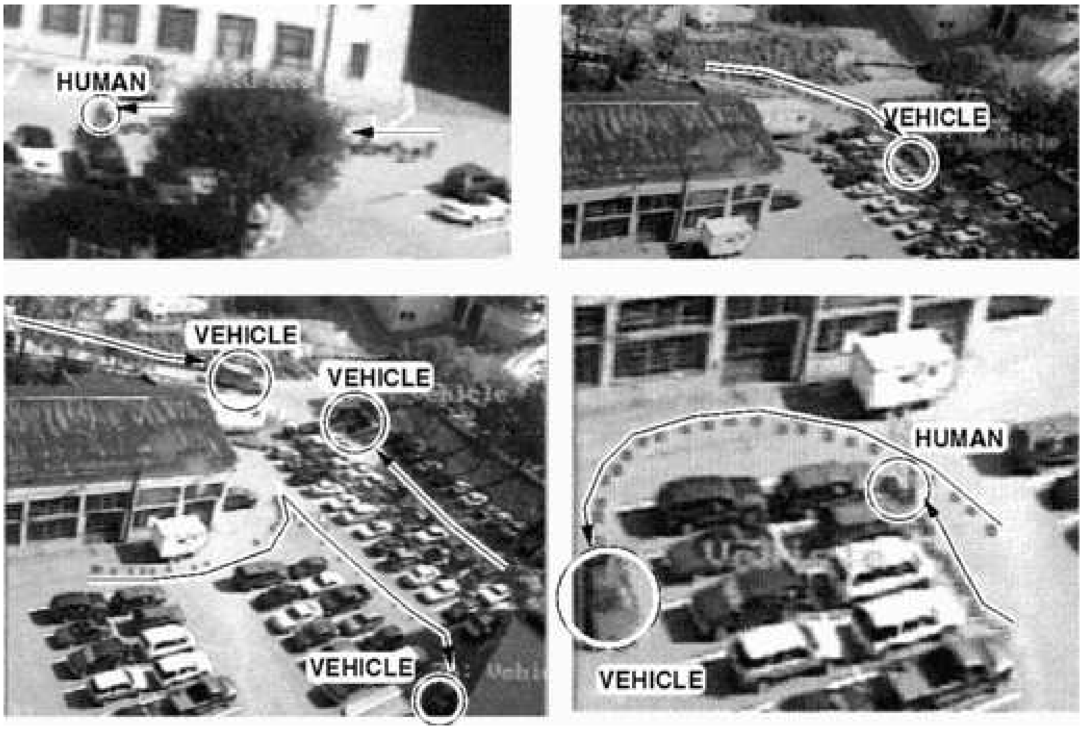


Figure 3: Recent results of moving entity detection and tracking showing detected objects and trajectories overlaid on original video imagery. Note that tracking persists even when targets are temporarily occluded or motionless.

2.3 Target Classification

The ultimate goal of the VSAM effort is to be able to identify individual entities (such as the "FedEx truck", the "4:15pm bus to Oakland" and "Fred Smith") and determine what they are doing. As a first step, entities are classified into specific class groupings such as "humans" and "vehicles".

Currently, we are experimenting with a neural network approach (Figure 4). The neural network is a standard three-layer network which uses a back propagation algorithm for hierarchical learning. Inputs to the network are a mixture of image-based and scene-based entity parameters: dispersedness (perimeter^2/area (pixels)); image area (pixels); aspect ratio (height/width); and camera zoom factor. Using a set of motion regions automatically extracted but labeled by hand, the network is trained to output one of three classes: human; vehicle; or human group (two or more humans walking close together). When teaching the network that an input entity is a human, all outputs are set to 0.0 except for "human", which is set to 1.0. Other classes are trained similarly. If the input does not fit any of the classes, such as a tree blowing in the wind, all outputs are set to 0.0.

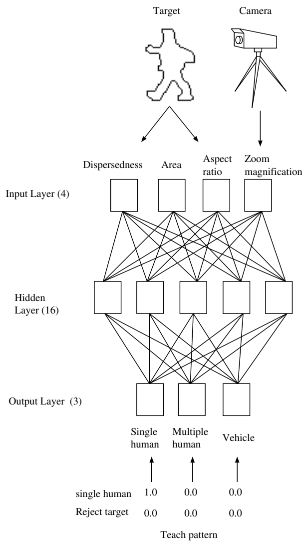


Figure 4: Neural network approach to target classification.

Results from the neural network are interpreted as follows:

```
if (output > THRESHOLD)
  classification = maximum NN output
else
  classification = REJECT
```

The results for this classification scheme are summarized in Table 1. This classification approach is effective for single images. However, one of the advantages of video is its temporal component. To exploit this, classification is performed on every entity at every frame and the results of classification are kept in a histogram with the *i*th bucket containing the number of times the object was classified as class *i*. At each time step, the class label that has been output most often for each object is chosen its most likely classification.

2.4 Activity Analysis

After classifying an object, we want to determine what it is doing. Understanding human activity is one of the most difficult open problems in the area of automated video surveillance. Detecting and analyzing human motion in real time from video imagery has only recently become viable with algorithms like *Pfinder* [Wren *et al.*, 1997] and W^4 [Haritaoglu *et al.*, 1998]. These algorithms represent a good first step to the problem of recognizing and analyzing humans, but they still have some drawbacks. In general, they work by detecting features (such as hands, feet and head), tracking them, and fitting them to some *a priori* human model such as the *cardboard model* of Ju *et al* [Ju *et al.*, 1996]. Therefore the human subject must dominate the image frame so that the individual body components can be reliably detected.

Class	Samples	% Correctly Classified
Human	430	99.5
Human group	96	88.5
Vehicle	508	99.4
False alarms	48	64.5
Total	1082	96.9

Table 1: Results for neural network classification algorithm.

We use a "star" skeletonization procedure for analyzing the motion of humans that are relatively small in the image. Details can be found in [Fujiyoshi and Lipton, 1998]. The key idea is that a simple form of skeletonization that only extracts the broad internal motion features of a target can be employed to analyze its motion. This method provides a simple, real-time, robust way of detecting extremal points on the boundary of the target to produce a "star" skeleton. The "star" skeleton consists of the centroid of an entity and all of the local extremal points which can be recovered when traversing the boundary of the entity's image (Figure 5a).

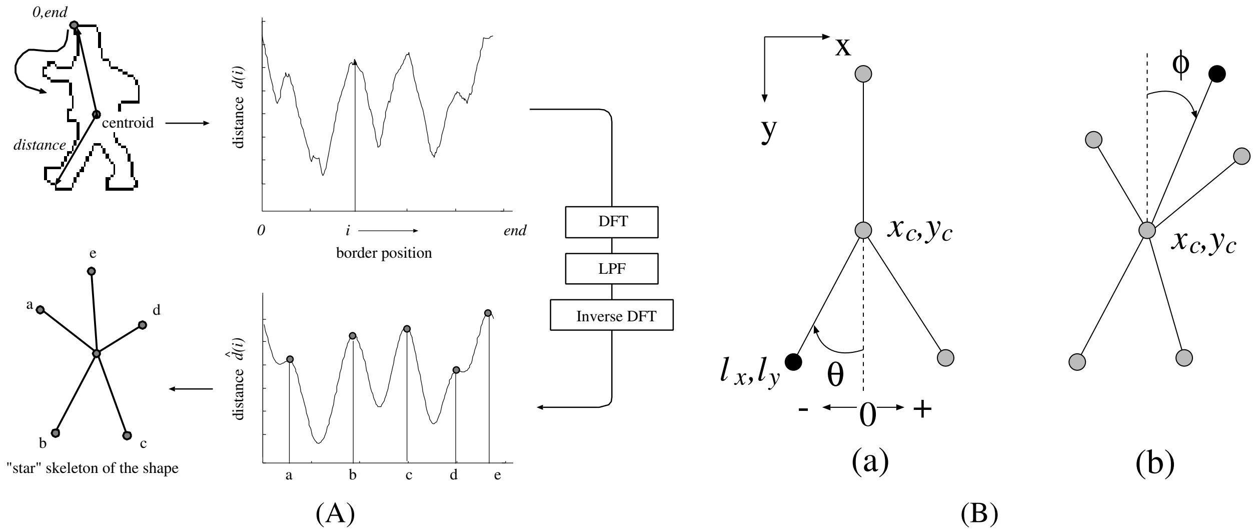


Figure 5: (A) The star skeleton is formed by "unwrapping" a region boundary as a distance function from the centroid. This function is then smoothed and extremal points are extracted. (B) Determination of skeleton features measuring gait and posture. θ is the angle the leftmost leg makes with the vertical, and ϕ is the angle the torso makes with the vertical.

Using only measurements based on the "star" skeleton, it is possible to determine the gait and posture of a moving human being. Figure 5b shows how two angles ϕ_n and θ_n are extracted from the skeleton. The value ϕ_n represents the angle of the torso with respect to vertical, while θ_n represents the angle of the leftmost leg in the figure. Figure 6 shows skeleton motion for typical sequences of walking and running humans, along with the values of ϕ_n and θ_n. These data were acquired in real-time from a video stream with frame rate 8Hz. Comparing the average values $\overline{\phi}_n$ in figures 6(e)-(f) show that the posture of a running target can easily be distinguished from that of a walking one using the angle of the torso segment as a guide. Also, the frequency of cyclic motion of the leg segments provides cues to distinguishing running from walking.

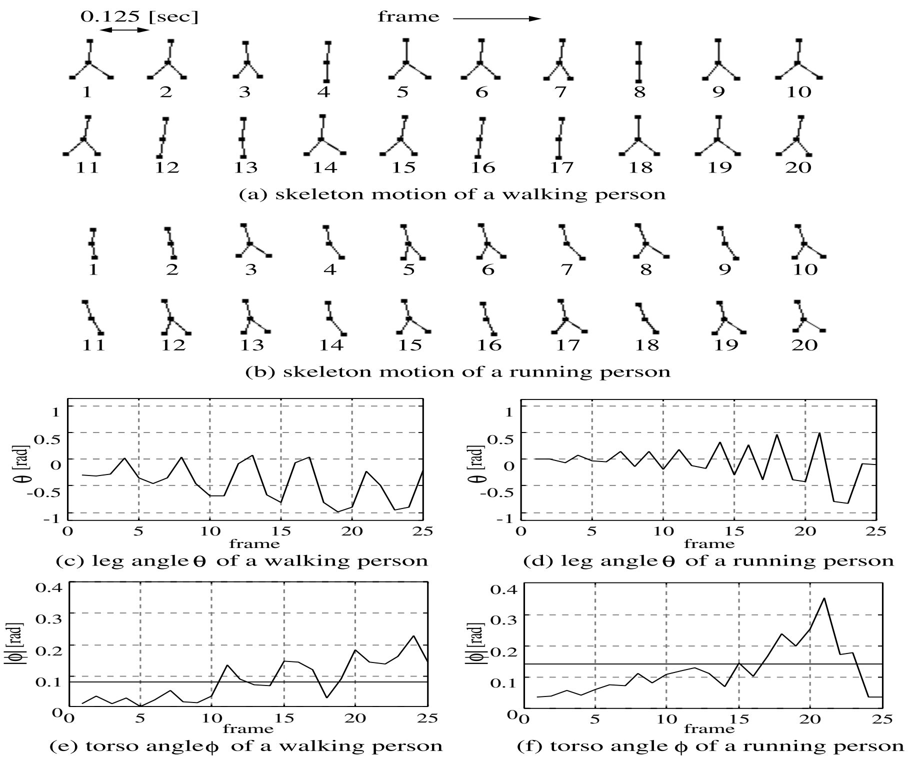


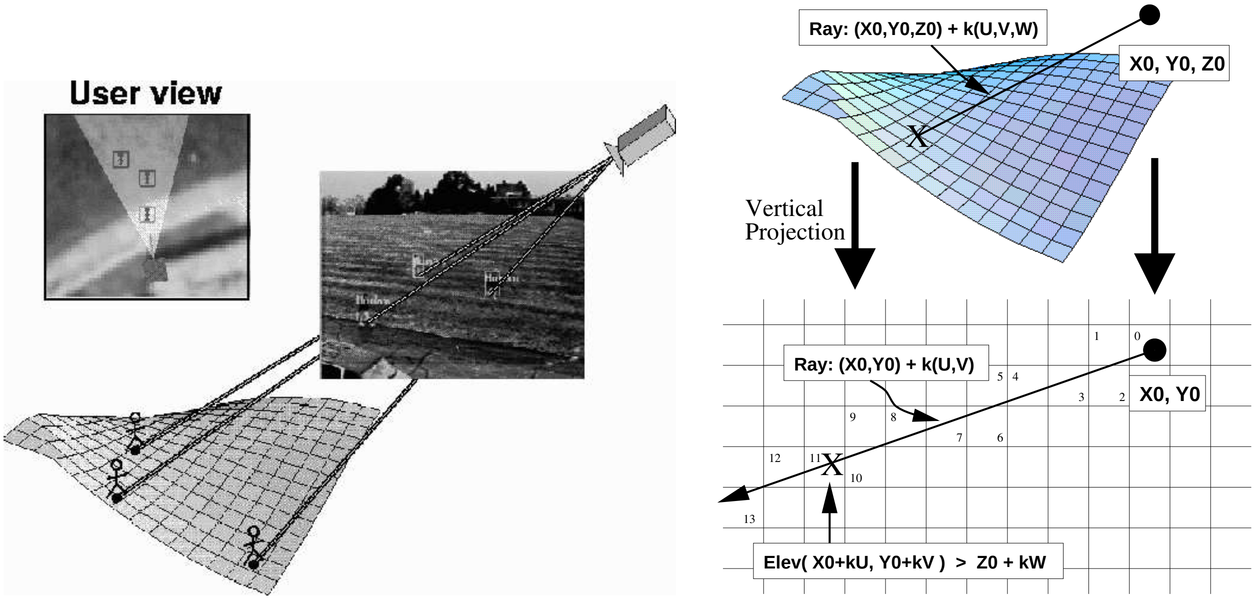
Figure 6: Skeleton motion sequences. Clearly, the periodic motion of θ_n provides cues to the target's motion as does the mean value of $\overline{\phi}_n$.

2.5 Model-based Geolocation

The video understanding techniques described so far have operated purely in image space. A large leap in terms of descriptive power can be made by transforming image blobs and measurements into 3D scene-based objects and descriptors. In particular, determination of object location in the scene allows us to infer the proper spatial relationships between sets of objects, and between objects and scene features such as roads and buildings. Furthermore, we believe the key to coherently integrating a large number of target hypotheses from multiple widely-spaced sensors is computation of target spatial geolocation.

In regions where multiple sensor viewpoints overlap, object locations can be determined very accurately by wide-baseline stereo triangulation. However, regions of the scene that can be simultaneously viewed by multiple sensors are likely to be a small percentage of the total area of regard in real outdoor surveillance applications, where it is desirable to maximize coverage of a large area given finite sensor resources. Determining target locations from a single sensor requires domain constraints, in this case the assumption that the object is in contact with the terrain. This contact location is estimated by passing a viewing ray through the bottom of the object in the image and intersecting it with a model representing the terrain (see Figure 7a). Sequences of location estimates over time are then assembled into consistent object trajectories.

Previous uses of the ray intersection technique for object localization in surveillance research have been restricted to small areas of planar terrain, where the relation between image pixels and terrain locations is a simple 2D homography [Bradshaw *et al.*, 1997, Flinchbaugh and Bannon, 1994, Koller *et al.*, 1993]. This has the benefit that no camera calibration is required to determine the back-projection of an image point onto the scene plane, provided the mappings of at least four coplanar scene points are known beforehand. However, large outdoor scene areas may



(A) (B)

Figure 7: (A) Estimating object geolocations by intersecting target viewing rays with a terrain model. (B) A Bresenham-like traversal algorithm determines which DEM cell contains the first intersection of a viewing ray and the terrain.

contain significantly varied terrain. To handle this situation, we perform geolocation using ray intersection with a full terrain model provided, for example, by a digital elevation map (DEM).

Given a calibrated sensor, and an image pixel corresponding to the assumed contact point between an object and the terrain, a viewing ray (x_0+ku, y_0+kv, z_0+kw) is constructed, where (x_0, y_0, z_0) is the 3D sensor location, (u, v, w) is a unit vector designating the direction of the viewing ray emanating from the sensor, and $k \geq 0$ is an arbitrary distance. General methods for determining where a viewing ray first intersects a 3D scene (for example, ray tracing) can be quite involved. However, when scene structure is stored as a DEM, a simple geometric traversal algorithm suggests itself, based on the well-known Bresenham algorithm for drawing digital line segments. Consider the vertical projection of the viewing ray onto the DEM grid (see Figure 7b). Starting at the grid cell (x_0, y_0) containing the sensor, each cell (x, y) that the ray passes through is examined in turn, progressing outward, until the elevation stored in that DEM cell exceeds the z-component of the 3D viewing ray at that location. The z-component of the view ray at location (x, y) is computed as either

$$z_0 + \frac{(x-x_0)}{u}w \quad \text{or} \quad z_0 + \frac{(y-y_0)}{v}w \tag{4}$$

depending on which direction cosine, u or v, is larger. This approach to viewing ray intersection localizes objects to lie within the boundaries of a single DEM grid cell. A more precise sub-cell location estimate can then be obtained by interpolation. If multiple intersections with the terrain beyond the first are required, this algorithm can be used to generate them in order of increasing distance from the sensor, out to some cut-off distance. See [Collins *et al.*, 1998] for more details.

2.6 Multi-Sensor Cooperation

In most complex outdoor scenes, it is impossible for a single sensor to maintain its view of an object for long periods of time. Objects become occluded by environmental features such as trees and buildings, and sensors have limited effective fields of regard. A promising solution to this problem is to use a network of video sensors to cooperatively track an object through the scene. Tracked objects are then *handed-off* between cameras to greatly extend the total effective area of surveillance coverage.

There has been little work done on autonomously coordinating multiple active video sensors to cooperatively track a moving target. One approach is presented by Matsuyama for a controlled indoor environment where four cameras lock onto onto a particular object moving across the floor [Matsuyama, 1998]. We approach the problem more generally by using the object's 3D geolocation as computed in the last section to determine where each sensor should look. The pan, tilt and zoom of the closest sensors are then controlled to bring the object within their fields of view, while a viewpoint independent cost function is used to determine which of the moving objects they find are the specific target of interest. These steps are described below.

Assume that at time t_0 a sensor with pan, tilt value (θ_0, ϕ_0) has been tasked to track a particular object with 3D ground location X_0 and velocity $\dot{X}$. Given a function $G(X)$ that converts a ground coordinate to a pan, tilt point (determined by camera calibration), the object's location X_0 is converted to a desired sensor pan, tilt value $(\theta_d, \phi_d) = G(X_0)$. The behavior of the pan, tilt unit is approximated by a linear system with infinite acceleration and maximum velocity $(\pm\dot{\theta}, \pm\dot{\phi})$ as

$$\begin{aligned} \theta(t) &= \theta_0 \pm \dot{\theta}(t - t_0) \\ \phi(t) &= \phi_0 \pm \dot{\phi}(t - t_0) \end{aligned} \tag{5}$$

Substituting the desired sensor pan, tilt (θ_d, ϕ_d) into the lefthand side of this equation and solving for $(t - t_0)$ yields a prediction of the acquisition time, that is, how long it would take for the pan, tilt device to point at the object's current location. However, the object will have moved further along its trajectory by that time. This new object position is estimated as

$$X(t) = X_0 + \dot{X}(t - t_0) \tag{6}$$

This predicted object position is then converted into a new desired sensor pan, tilt, and the whole procedure iterates until the time increments $(t - t_0)$ become small (convergence) or start to increase (divergence). This algorithm guarantees that if it converges, the sensor will be able to reacquire the object.

An appropriate camera zoom setting can be determined directly given a desired size of the object's projection in the image. Knowing the classification of the object C (as determined from Section 2.3), we employ the heuristic that humans are approximately 6 (2m) feet tall and vehicles are approximately 15 feet (5m) long to set the zoom. Given the position of the object and the sensor and, therefore the range r to the object, the angle ρ subtended by the image of the object is approximately

$$\rho = \begin{cases} \tan^{-1}\frac{2}{r}, & \text{human} \\ \tan^{-1}\frac{5}{r}, & \text{vehicle} \end{cases}$$

Knowing the focal length of the sensor as a function of zoom, as determined from camera calibration, the appropriate zoom setting is easily chosen.

Once the sensor is pointing in the right direction at the right zoom factor, all moving targets extracted are compared to the specific target of interest to see if they match. This need to re-acquire a specific target is a key feature necessary for multi-camera cooperative surveillance. Obviously viewpoint-specific appearance criteria are not useful, since the new view of the target may be significantly different from the previous view. Therefore, recognition features are needed that are independent of viewpoint. In our work we use two such criteria: the object's 3D scene trajectory as determined from geolocation, and a normalized color histogram of the object's image region. Candidate motion regions are tested by applying a matching cost function in a manner similar to that described in Section 2.2.

3 A VSAM Testbed System

We have built a prototype VSAM testbed system to demonstrate how the automated video understanding technology described in the last section can be combined into a coherent surveillance system that enables a single human operator to monitor a wide area. The testbed system consists of a central operator control unit (OCU) which receives video and ethernet data from multiple remote sensor processing units (SPUs) (see Figure 8). The OCU is responsible for integrating symbolic object trajectory information accumulated by each of the SPUs together with a 3D geometric site model, and presenting the results to the user on a map-based graphical user interface. Each component of the testbed system architecture is described briefly below.

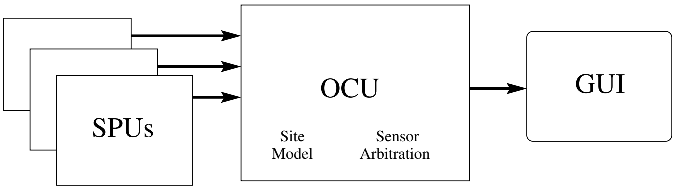


Figure 8: Overview of the VSAM testbed system.

3.1 Sensor Processing Units (SPUs)

Sensor processing units (SPUs) are the front-end sensing nodes of the testbed system. Their function is to automatically extract significant entities and events from video imagery, using the algorithms described in the last section. All video processing is performed on board the SPU, and the resulting object hypotheses are transmitted in symbolic form back to the OCU, significantly reducing the bandwidth requirements of the surveillance network. For example, we currently process NTSC color imagery with a frame size of 320x240 pixels at 10 frames per second on a Pentium II PC, so that data is streaming into the system through each SPU at a rate of roughly 2.3Mb per second per sensor. After VSAM processing, detected targets hypotheses contain information about object type, target location and velocity, as well as measurement statistics such as a time stamp and a description of the sensor (current pan, tilt, and zoom for example). Each target data packet takes up roughly 50 bytes. If a sensor tracks 3 targets for one second at 10 frames per second, it ends up transmitting 1500 bytes back to the OCU, well over a thousandfold reduction in data bandwidth.

The VSAM testbed can handle a wide variety of sensors and modalities. The current list of SPUs that we have successfully integrated into the system are:

- Five fixed-mount color CCD sensors with variable pan, tilt and zoom control, affixed to buildings around the CMU campus.
- One van-mounted relocatable SPU that can be moved from one point to another during a surveillance mission.
- A FLIR Systems camera turret mounted on an aircraft.
- A Columbia-Lehigh CycloVision ParaCamera with a hemispherical field of view.
- A Texas Instruments (TI) indoor surveillance system, which after some modifications is capable of directly interfacing with the VSAM network.

Logically, all of these SPUs are treated identically. In future years, it is hoped that other VSAM sensor modalities will be added, including thermal infrared sensors, multi-camera omnidirectional sensors, and stereo sensors.

3.2 Operator Control Unit (OCU)

Figure 9 shows the functional architecture of the VSAM OCU. It accepts video processing results from each of the SPUs and integrates the information with a site model and a database of known targets to infer activities that are of interest to the user. This data is sent to the GUI and other visualization tools as output from the system.

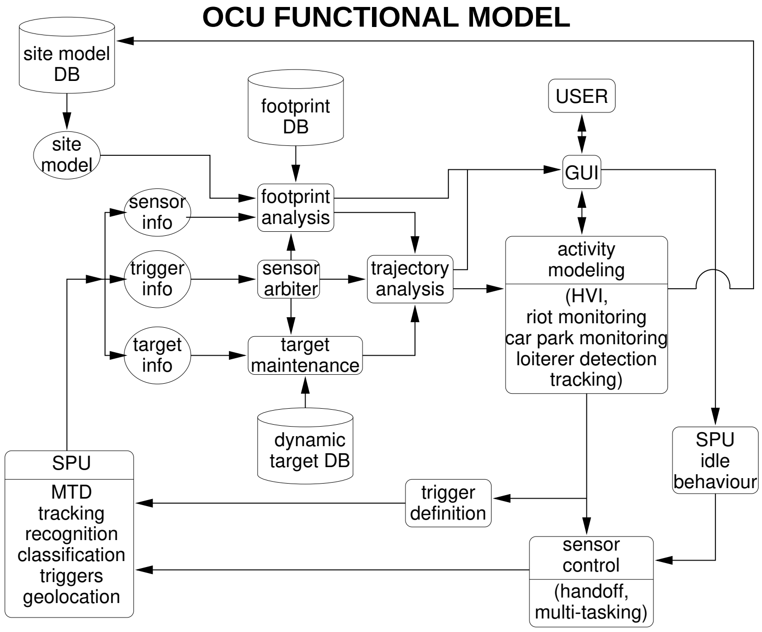


Figure 9: Functional architecture of the VSAM OCU.

One key piece of system functionality provided by the OCU is sensor arbitration. It is important in a surveillance system that sensor assets are not underutilised. Sensors must be allocated to surveillance tasks in such a way that all possible user-specified tasks can be performed, and if enough sensors are present, multiple sensors can be assigned to track important objects. At any given time, the OCU maintains a list of known objects and sensor parameters, as well as a set of "tasks" that may need attention. These tasks are explicitly indicated by the user through the GUI, and may include such things as specific targets to be tracked, specific regions to be watched, or specific events to be detected (such as a person loitering near a particular doorway). Sensor arbitration is performed by an arbitration cost function. The arbitration function determines the cost of assigning each of the SPUs to each of the tasks. These costs are based on the priority of the tasks, the load on the SPU, the visibility of the objects from a particular sensor, and so on. The system performs a greedy optimization of the cost to determine the best combination of SPU tasking to maximize overall system performance requirements.

The OCU also contains a site model representing VSAM-relevant information about the area being monitored. This includes both geometric and photometric information about the scene, represented using a combination of image and symbolic data. Site model representations have intentionally been kept as as simple as possible, with an eye towards efficiently supporting specific VSAM capabilities:

- target geolocation via intersection of viewing rays with the terrain.
- visibility analysis (predicting what what portions of the scene are visible from what sensors) so that sensors can be efficiently tasked.
- specification of of the geometric location and extent of relevant scene features. For example, we might directly

task a sensor to monitor the door of a building, or to look for vehicles passing through a particular intersection.

3.3 Graphical User Interface

One of the technical goals of the VSAM project is to demonstrate that a single human operator can effectively monitor a significant area of interest. Keeping track of multiple people, vehicles, and their interactions, within a complex urban environment is a difficult task. The user obviously shouldn't be looking at two dozen screens showing raw video output – that amount of sensory overload virtually guarantees that information will be ignored and would require a prohibitive amount of transmission bandwidth. Our approach is to provide an interactive, graphical user interface (GUI) that uses VSAM technology to automatically place dynamic agents representing people and vehicles into a synthetic view of the environment. This approach has the benefit that visualization of scene events is no longer tied to the original resolution and viewpoint of a single video sensor. The GUI currently consists of a map of the area, with all target and sensor platform locations overlaid on it.

In addition to scene visualization, the GUI is also used for sensor suite tasking. Through this interface, the operator can task individual sensor units, as well as the entire testbed sensor suite, to perform surveillance operations such as generating a quick summary of all target activities in the area.

4 VSAM Demonstrations

We have held two significant demonstrations of the VSAM system in the past two years. VSAM Demo I was held at CMU's Bushy Run research facility on November 12, 1997, roughly nine months into the program. The VSAM testbed system at that time consisted of an OCU with two ground-based and one airborne SPU. The two ground sensors cooperatively tracked a car as it entered the Bushy Run site, parked and let out two occupants. The two pedestrians were detected and tracked as they walked around and then returned to their car. The system continued tracking the car as it commenced its journey around the site, handing off control between cameras as the car left the field of view of each sensor. All entities were detected and tracking using temporal differencing motion detection and correlation-based tracking. Targets were classified into "vehicle" or "human" using a simple image-based property (aspect ratio) in conjunction with a temporal consistency constraint. Target geolocation was accomplished by intersection of back-projected viewing rays with the DEM terrain model. A synopsis of the vehicle trajectory computed automatically by the system is shown in Figure 10.

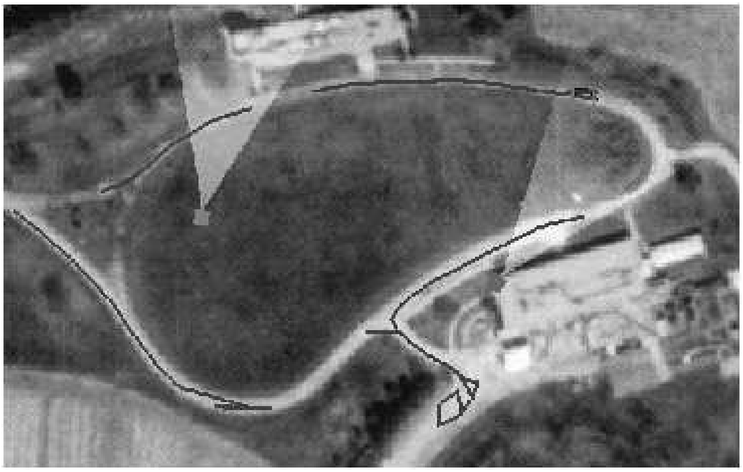


Figure 10: Synopsis of vehicle trajectory during the Bushy Run demo.

VSAM Demo II was help on October 8 on the urban campus of CMU. Five fixed-mount pan-tilt-zoom cameras were mounted on buildings around campus, a relocatable SPU mounted in a van, an airborne SPU operated by the Army's Night Vision and Electronic Sensor Directorate, a hemispherical field of view sensor operated by Lehigh and Columbia Universities, and an indoor video alarm system run by Texas Intruments. These sensors cooperated to track a vehicle from nearby Schenley Park onto campus, followed its path as it wound through campus and stopped at the OCU building, alerted the operator when one of the vehicle's occupants entered the building, and followed the ensuing car and foot chases as the vehicle and its occupants attempted to flee from the police. An example of multi-sensor tracking of the vehicle as it attempted to leave campus is shown in Figure 11. This diagram shows the continuous, autonomous tracking of a single object for a distance of approximately 400m and a time of approximately 3 minutes. In Figure 11(a) two sensors cooperatively track the object. At the time shown in Figure 11(b) the object is occluded from sensor 2, but is still visible from sensor 1, which continues to track it. When the object moves out of the occlusion area, sensor 2 is automatically retasked to track it, as shown in Figure 11(c).

Finally, when the object moves out of the field of regard of both sensors, a third sensor is automatically tasked to continue surveillance, as shown in Figure 11(d). By automatically managing multiple, redundant camera resources, the vehicle is continously tracked through a complex urban environment.

5 Conclusion

CMU and the Sarnoff Corporation have developed a testbed system for automated vision surveillance and monitoring. Multiple sensors cooperate to track moving objects through a complex, urban environment. More information on this project, particularly on the airborne processing component, can be found in [Kanade *et al.*, 1998].

As the VSAM effort enters its third year, we are focusing on making the video understanding algorithms more robust to lighting and environmental conditions, adding thermal cameras to the permanent test sensor suite located on the CMU campus, adding more complex ground sensor control strategies such as sensor multi-tasking, and the ability to perform unsupervised monitoring of limited domains such as parking lots, and expanding the testbed system's network architecture to handle Web-VSAM – remote site monitoring and SPU control over the internet using a JAVA DIS client.

Acknowledgments

The authors would like to thank the CMU VSAM team members: Hironobu Fujiyoshi, Dave Duggins, David Tolliver, Raju Patil, Yanghai Tsin, and Alan Lee, for their tireless efforts and good humor over the last two years.

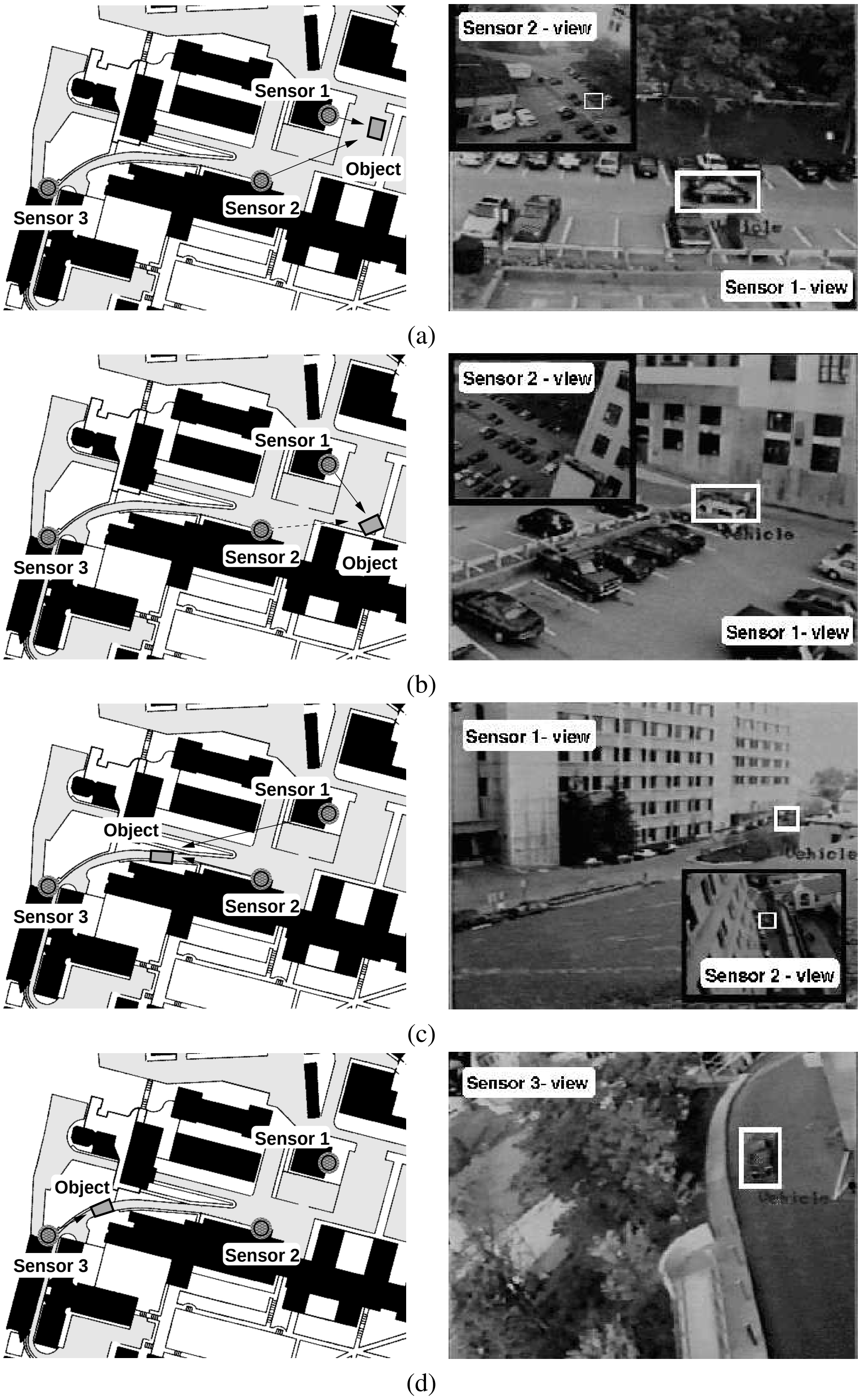


Figure 11: Cooperative, multi-sensor tracking (see text for description).

References

[Anderson *et al.*, 1985] C. Anderson, Peter Burt, and G. van der Wal. Change detection and tracking using pyramid transformation techniques. In *Proceedings of SPIE - Intelligent Robots and Computer Vision*, volume 579, pages 72–78, 1985.

[Bar-Shalom and Fortmann, 1988] Yaakov Bar-Shalom and Thomas Fortmann. *Tacking and data association*. Academic Press, Boston, 1988.

[Barron *et al.*, 1994] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):42–77, 1994.

[Bradshaw *et al.*, 1997] K. Bradshaw, I. Reid, and D. Murray. The active recovery of 3d motion trajectories and their use in prediction. *PAMI*, 19(3):219–234, March 1997.

[Collins *et al.*, 1998] Robert Collins, Yanghai Tsin, J.Ryan Miller, and Alan Lipton. *Using a DEM to Determine Geospatial Object Trajectories*. CMU technical report CMU-RI-TR-98-19, 1998.

[Flinchbaugh and Bannon, 1994] B. Flinchbaugh and T. Bannon. Autonomous scene monitoring system. In *Proc. 10th Annual Joint Government-Industry Security Technology Symposium*. American Defense Preparedness Association, June 1994.

[Fujiyoshi and Lipton, 1998] Hironobu Fujiyoshi and Alan Lipton. Real-time human motion analysis by image skeletonization. In *Proceedings of IEEE WACV98*, 1998.

[Grimson and Viola, 1997] Eric Grimson and Paul Viola. A forest of sensors. In *Proceedings of DARP - VSAM workshop II*, November 1997.

[Haritaoglu *et al.*, 1998] I. Haritaoglu, Larry S. Davis, and D. Harwood. w^4 who? when? where? what? a real time system for detecing and tracking people. In *FGR98 (submitted)*, 1998.

[Isard and Blake, 1996] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proceedings of European Conference on Computer Vision 96*, pages 343–356, 1996.

[Ju *et al.*, 1996] S. Ju, M. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proceedings of International Conference on Face and Gesture Analysis*, 1996.

[Kanade *et al.*, 1998] Takeo Kanade, Robert Collins, Alan Lipton, Peter Burt, and Lambert Wixson. Advances in cooperative multi-sensor video surveillance. In *Proceedings of DARPA Image Understanding Workshop*, volume 1, pages 3–24, November 1998.

[Koller *et al.*, 1993] D. Koller, K. Daniilidis, and H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *IJCV*, 10(3):257–281, June 1993.

[Lipton *et al.*, 1998] Alan Lipton, Hironobu Fujiyoshi, and Raju S. Patil. Moving target detection and classification from real-time video. In *Proceedings of IEEE WACV98*, 1998.

[Matsuyama, 1998] Takashi Matsuyama. Cooperative distributed vision. In *Proceedings of DARPA Image Understanding Workshop*, volume 1, pages 365–384, November 1998.

[Wren *et al.*, 1997] C. Wren, A. Azarbayejani, T. Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.