# Advances in Cooperative Multi-Sensor Video Surveillance*

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

**Peter Burt and Lambert Wixson**
The Sarnoff Corporation, Princeton, NJ
E-MAIL: {pburt,lwixson}@sarnoff.com

## Abstract

Carnegie Mellon University (CMU) and the Sarnoff Corporation (Sarnoff) are performing an integrated feasibility demonstration of Video Surveillance and Monitoring (VSAM). The objective is to develop a cooperative, multi-sensor video surveillance system that provides continuous coverage over battlefield areas. Significant achievements have been demonstrated during VSAM Demo I in November 1997, and in the intervening year leading up to Demo II in October 1998.

## 1 Introduction

The thrust of the VSAM Integrated Feasibility Demonstration (IFD) research program is cooperative multi-sensor surveillance to support enhanced battlefield awareness [Kanade *et al.*, 1997]. We are developing automated video understanding technology that will enable a single human operator to monitor activities over a complex area using a distributed network of active video sensors. Our advanced Video Understanding technology can automatically detect and track multiple people and vehicles within cluttered scenes, and 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. The idea is to allow a commander or military analyst to tap into a network of sensors deployed on and over the battlefield to get a broad overview of the current situation. Other military and law enforcement applications include providing perimeter security for troops, monitoring peace treaties or refugee movements from 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.

Keeping track of people, vehicles, and their interactions, over a chaotic area such as the battlefield, is a difficult task. Populating the battlefield with digital sensing units can provide the commander with up-to-date sensory feedback leading to improve situational awareness and better decision making. 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 using a combination of temporal differencing and template tracking [Lipton *et al.*, 1998] (in this proceedings). 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. Further classification of human activity, such as walking and running, has also been achieved [Fujiyoshi and Lipton, 1998] (in

*Funded by DARPA contract DAAB07-97-C-J031.

this proceedings). 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 [Collins *et al.*, 1998] (in this proceedings.) An airborne surveillance platform has been incorporated into the system, and novel real-time algorithms for camera fixation, sensor multi-tasking, and moving target detection have been developed for this moving platform [Wixson *et al.*, 1998]. Resulting target hypothesis information from all sensor processing units (SPUs), including target type and trajectory, are transmitted as symbolic data packets back to a central operator control unit (OCU), where they are displayed on a graphical user interface to give a broad overview of scene activities.

This paper provides an overview of VSAM research at Carnegie Mellon University and the Sarnoff Corporation, drawing upon results achieved during VSAM Demo I held on November 12 1997 at Bushy Run, and preliminary results from the new VSAM IFD testbed system that will be unveiled during Demo II at CMU on October 8, 1998. Section 2 describes the components and infrastructure underlying the current VSAM testbed system, while Section 3 details the video understanding technologies that provide core system functionality. Section 4 outlines a key idea of our research, namely the use of geospatial site models to enhance system performance. The role of the human operator in achieving battlefield awareness also can not be overstated – our intention is not to design a fully automated, stand-alone system, but rather to provide a human commander with timely information regarding events unfolding on the battlefield. To this aim, a graphical user interface for visualizing large scale, multi-agent events is a vital system component, and the relevant human-computer interface issues are discussed in Section 5. Finally, Section 6 provides a roadmap of where we have been and where we are going by outlining the significant technological accomplishments that we have achieved during VSAM Demo I, and that are planned for Demos II and III.

# 2 VSAM Testbed System

The current VSAM testbed system is evolving along the path outlined in the 1997 VSAM PI report [Kanade *et al.*, 1997]. The system consists of a central operator control unit (OCU) which receives video and ethernet data from remote sensor processing units (SPUs) (Figure 1). The OCU is responsible for integrating symbolic and video information accumulated by each of the SPUs and presenting it in a concise, meaningful form to users operating VSAM visualization tools. A central graphical user interface (GUI) is used to task the system by specifying which areas, targets and events are worthy of special attention.

## 2.1 Sensor Processing Units (SPUs)

Sensor processing units (SPUs) are the front end nodes of the VSAM network. Their function is to analyze video imagery for the presence of significant entities or events and transmit that information to the OCU. The notion is that SPUs act as intelligent filters between sensing devices and the VSAM network. This arrangement allows for many different sensor modalities to be seamlessly integrated into the system. Furthermore, performing as much video processing as possible on-board the SPU reduces the bandwidth requirements of the VSAM network. Full video signals do not need to be transmitted; only symbolic data extracted from video signals.

The VSAM testbed can handle a wide variety of sensors and sensor platforms. SPUs can vary from simple sensors with rudimentary vision processing capabilities to very sophisticated sensing systems capable of making intelligent inferences about activities in their field of regard. SPUs can be connected to the system by radio links, serial lines, wireless ethernet, cables, or any other medium. The 1997 demo saw the integration of monochrome CCD sensors with an airborne color CCD sensor. In 1998, the list of SPUs includes:

- Five fixed-mount color CCD sensors with variable pan, tilt and zoom control, affixed to buildings around the CMU campus.

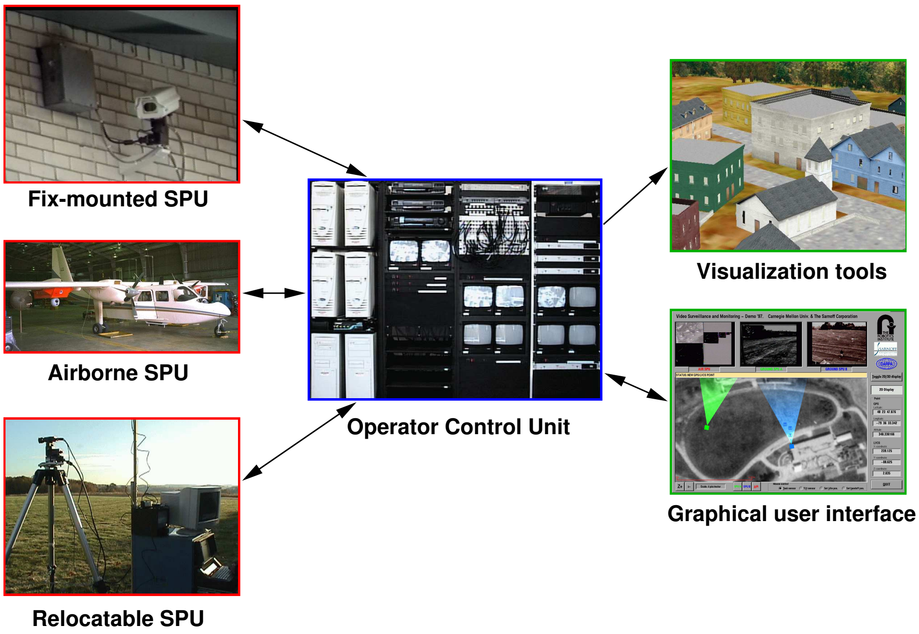



**Figure 1:** Overview of the VSAM testbed system.

- 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. They differ only in the type of physical connection required to the OCU. In future years, it is hoped that other VSAM sensor modalities will be added, including thermal infra-red sensors, multi-camera omnidirectional sensors, and stereo sensors.

### 2.2 Airborne SPU

The airborne SPU warrants further discussion. The sensor and computation packages are mounted on a Britten-Norman Islander twin-engine aircraft operated by the U.S. Army Night Vision and Electronic Sensors Directorate. The Islander, shown in Figure 2 is equipped with a

**Figure 2:** The Night Vision Lab's Islander aircraft. Camera turret is below wing at left.

**Figure 3:** The Sarnoff airborne camera simulator.

FLIR Systems Ultra-3000 turret that has two degrees of freedom (pan/tilt), a GPS system for measuring position, and an AHRS (Attitude Heading Reference System) device for measuring orientation. Video processing is performed using on-board Sarnoff VFE-100 and PVT-200 video processing engines [Hansen *et al.*, 1994] in Years 1 and 2, respectively.

Because the cost of testing algorithms on the airplane is high, we have constructed a simulated airborne platform using a gantry, shown in Figure 3. The gantry travels in the XY plane with a camera suspended beneath it on a pan/tilt mount. This enables us to obtain quantitative results and simulate airborne performance while debugging the airborne VSAM algorithms.

## 2.3 Operator Control Unit (OCU)

Figure 4 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. Data about relevant entities and events is packaged up and transmitted to the GUI and other visualization tools.

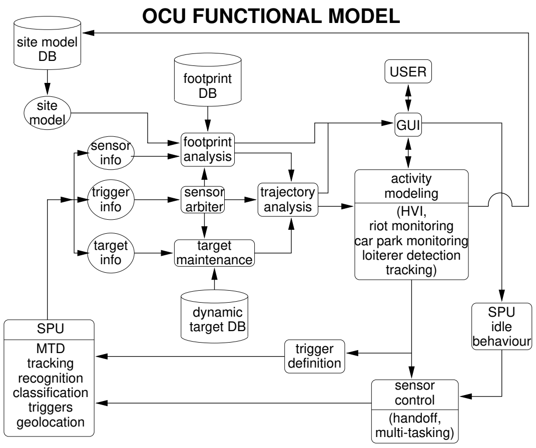



**Figure 4:** Functional architecture of the VSAM OCU.

One key piece of system functionality provided by the OCU is sensor arbitration. At any given time, the system has a number 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 tasks, 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.

## 2.4 Graphical User Interface (GUI)

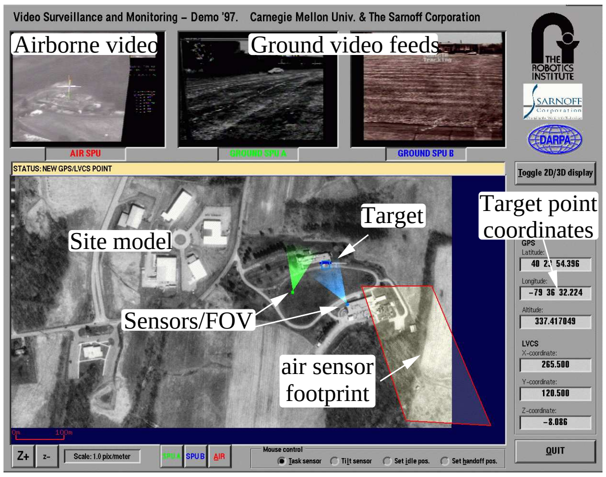



**Figure 5:** VSAM Demo I Graphical User Interface.

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. Towards this end, the testbed employs a graphical user interface for scene visualization and 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. The operator may choose to see a map of the area, with all target and sensor platform locations overlaid on it (a sample of the Demo I GUI can be seen in Figure 5).

## 2.5 Communication

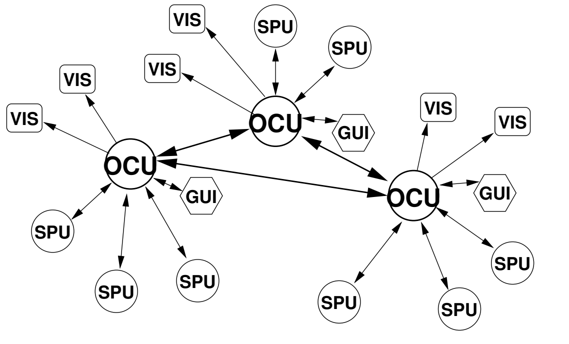



**Figure 6:** A nominal architecture for expandable VSAM networks.

Figure 6 depicts a nominal architecture for the VSAM network. It allows multiple OCUs to be linked together, each controlling multiple SPUs. Each OCU supports exactly one GUI through which all user related command and control information is passed. However, data dissemination is not limited to a single user interface, but is also accessible through a series of visualisation nodes (VIS).

There are two independent communication protocols and packet structures supported in this architecture: the Carnegie Mellon University Packet Architecture (CMUPA) and the Distributed Interactive Simulation (DIS) protocols. The CMUPA is designed to be a low bandwidth, highly flexible architecture in which relevant VSAM information can be compactly packaged without redundant overhead. The concept of the CMUPA packet architecture is a hierarchical decomposition. There are six data sections that can be encoded into a packet: command; sensor; image; target; event; and region of interest (ROI). A short packet header section describes which of these six sections are present in the packet. Within each section it is possible to represent multiple instances of that type of data, with each instance potentially containing a different layout of information. At each level, short bitmasks are used to describe the contents of the various blocks within the packets, keeping wasted space to a minimum. All communication between SPUs, OCUs and GUIs is CMUPA compatible. The CMUPA protocol specification document is accessible from `http://www.cs.cmu.edu/~vsam`.

VIS nodes are designed to distribute the output of the VSAM network to where it is needed. They provide symbolic representations of detected activities overlaid on maps or imagery. Information flow to VIS nodes is unidirectional, originating from an OCU. All of this communication uses the DIS protocol, which is described in detail in [IST, 1994]. An important benefit to keeping VIS nodes DIS compatible is that it allows us to easily interface with synthetic environment visualization tools such as ModSAF and ModStealth (Section 5).

# 3 Video Understanding Technologies

## 3.1 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 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 [Haritaoglu *et al.*, 1998] and is an attempt to make background subtraction more robust to environmental dynamism. The notion is to use an adaptive background model to accommodate changes to the background while maintaining the ability to detect independently moving targets.

The first of these issues is dealt with by using a statistical model of the background to provide a mechanism to adapt 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 $\sigma_{p_n}$ are maintained by temporal filtering. Due to the filtering process, these statistics change over time reflecting dynamism in the environment.

The filter is of the form

$$F(t) = e^{\frac{t}{\tau}} \tag{1}$$

where $\tau$ is a time constant which can be configured to refine the behavior of the system. The filter is implemented:

$$\begin{aligned} \overline{p}_{n+1} &= \alpha p_{n+1} + (1-\alpha)\overline{p}_n \\ \overline{\sigma}_{n+1} &= \alpha |p_{n+1} - \overline{p}_{n+1}| + (1-\alpha)\overline{\sigma}_n \end{aligned} \tag{2}$$

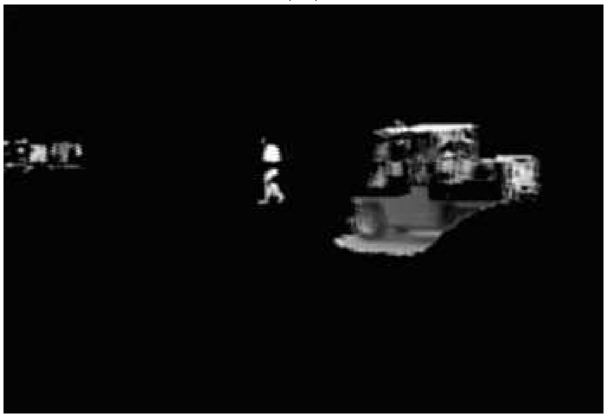
where $\alpha = \tau \times f$, and $f$ is the frame rate. Unlike the models of both [Haritaoglu *et al.*, 1998] and [Wren *et al.*, 1997], this statistical model incorporates noise measurements to determine foreground pixels, rather than a simple threshold. This idea is inspired by [Grimson and Viola, 1997].

If a pixel has a value which is more than $2\sigma$ 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.

"Moving" pixels are aggregated using a connected component approach so that individual target regions 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 $3\tau$) absorbed into the background (see Figure 7).

While this class of detection methods is inapplicable to video streams from moving cameras, it can be employed in a "step and stare" mode in which the system can predict the position of a target and point the camera in such a way as to reacquire it. Most background subtraction schemes require a period of time in which

(A)



(B)

**Figure 7:** Example of moving target detection by dynamic background subtraction.

the scene remains static so that a background model can be built. This is unacceptable in a video surveillance application where motion detection must be available to the system immediately after a "step and stare" move. In this implementation, temporal differencing (3-frame differencing) is used as a stop-gap measure until the background model is stabilized. While the quality of the moving target detection is diminished using this method, it is assumed that since the system is in a tracking mode, it has some notion of which target it is looking for, and where it may be located, so this inferior motion detection scheme will be adequate for the system to reacquire the target.

The MTD algorithm is prone to three types of error: incomplete extraction of a moving object; erroneous extraction of non-moving pixels; and legitimate extraction of illegitimate targets (such as trees blowing in the wind). Incomplete

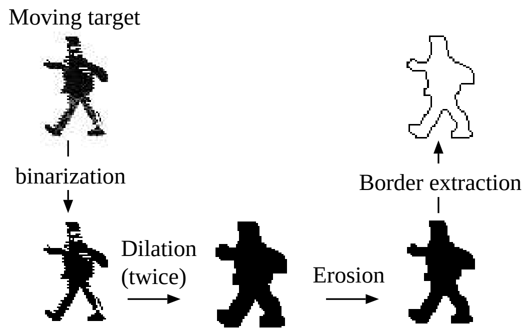



**Figure 8:** Target pre-processing. A moving target region is morphologically dilated (twice), eroded and then its border is extracted.

targets are partially reconstructed by blob clustering and morphological dilation (Figure 8). 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.

### 3.2 Target Tracking

One main purpose of the system is to build a temporal model of activity. To do this, 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 frame-to-frame.

Many systems for target tracking are based on Kalman filters, but as pointed out by [Isard and Blake, 1996], they are of limited use because they are based on unimodal Gaussian densities and cannot support simultaneous alternative motion hypotheses. A few other approaches have been devised; Isard and Blake [Isard and Blake, 1996] present a new stochastic algorithm for robust tracking which is superior to previous Kalman filter based approaches; and Bregler [Bregler, 1997] presents a probabilistic decomposition of human dynamics to learn and *recognise* human beings (or their gaits) in video sequences.

The IFD testbed system uses a much simpler approach based on a frame-to-frame matching cost function. A record of each blob is kept with the following information:

- trajectory (position $p(t)$ and velocity $v(t)$ as functions of time) in image coordinates,
- associated camera calibration parameters so the target's trajectory can be normalized to an absolute coordinate system ($\hat{p}(t)$ and $\hat{v}(t)$),
- the "blob" data as an image chip,
- "blob" size $s$ and centroid $c$,
- Color histogram $h$ of "blob".

First, the position and velocity of $T_i$ from the last time step $t_{last}$ is used to determine a predicted position for $T_i$ at the current time $t_{now}$.

$$\hat{p}_i(t_{now}) \approx \hat{p}_i(t_{last}) + \hat{v}_i(t_{last}) \times (t_{now} - t_{last}) \tag{3}$$

Using this information a matching cost can be determined between a known target $T_i$ and current moving "blob" $R_j$

$$C(T_i, R_j) = f(|\hat{p}_i - \hat{p}_j|, |s_i - s_j|, |c_i - c_j|, |h_i - h_j|) \tag{4}$$

Matched targets are then maintained over time. This method will fail if there are occlusions. For this reason, significant targets (chosen by either the user or the system) are tracked using a combination of the cost function and adaptive template matching [Lipton *et al.*, 1998]. Recent results from the system are shown in Figure 9.

### 3.3 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". As a first step, entities are classified into specific class groupings such as "humans" and "vehicles". An initial effort in this work [Lipton *et al.*, 1998] used view independent visual properties to classify entities into three classes: humans; vehicles; and clutter. In the rural environment of the Bushy Run demo, this is an adequate first step; however, in the less constrained urban environment of Demo II,

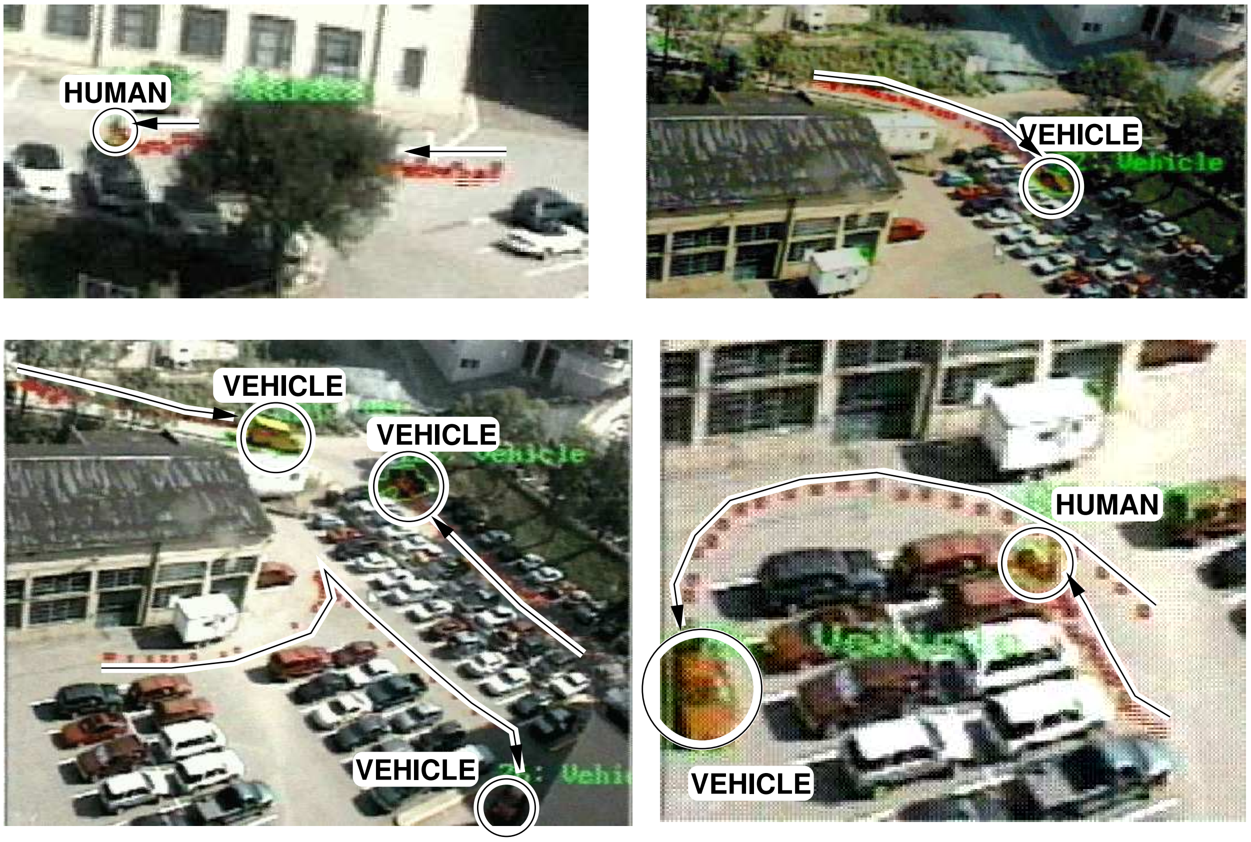



**Figure 9:** Recent results of moving entity detection and tracking showing detected objects and trajectories overlaid on original video imagery. Note that tracking persists even when targets are temporarily occluded or motionless.

a more robust method is employed with a larger number of entity classes.

As mentioned in section 3.1 two different MTD algorithms might be employed depending on the stability of the background scene. The quality of the motion detection extracted by three frame differencing is inferior to that of dynamic background subtraction, so a different classification metric must be employed.

In the usual case when entities are accurately extracted by background subtraction, a neural network approach is used (Figure 10). 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)); area (pixels); apparent aspect ratio; and camera zoom. The network will output three classes: human; vehicle; or human group.

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.

Results from the neural network are interpreted as follows:

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

In the case when the background model is unstable, and three frame differencing is used to detect moving targets, a different classification criterion is used. Given the geolocation of the entity, its actual width $w$ and height $h$ in meters

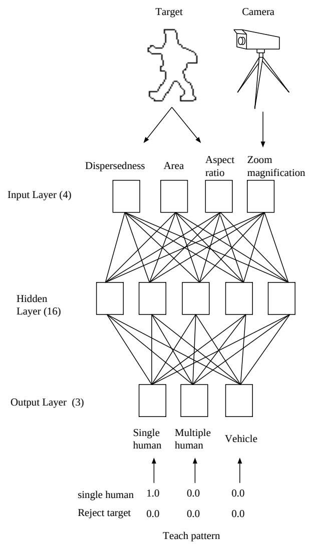



**Figure 10:** Neural network approach to target classification.

| Class | Samples | % 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 classification algorithms on VSAM IFD data

can be determined. Then a heuristic is applied to these values:

$$\begin{array}{llll} w < 1.1 & h \in [0.5, 2.5] & \Rightarrow & \text{human} \\ w \in [1.1, 2.2] & h \in [0.5, 2.5] & \Rightarrow & \text{group} \\ w \in [2.2, 20] & h \in [0.7, 4.5] & \Rightarrow & \text{vehicle} \\ \textbf{ELSE} & & \Rightarrow & \text{reject} \end{array} \quad (5)$$

The results for this classification scheme are summarized in table 1.

These classification metrics are effective for single images. 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. At each time step, the most likely class label is then chosen as the entity classification, as described in [Lipton *et al.*, 1998].

### 3.4 Target Recognition

One of the key features of the VSAM IFD testbed system is the ability to re-acquire a specific target in a video image. It may be necessary to do this from a single camera when the target has temporarily been lost, or between two different cameras when performing handoff. 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 this system, two such criteria are used: absolute trajectory; and color histogram. The first is computed by geolocating targets using viewing ray intersection with a scene model (Section 4.3), and the second is determined in a normalized RGB color space.

The first step in recognition is to predict the position in which the target is likely to appear. This is done using equation 3. After this, candidate motion regions are tested by applying a matching cost function. The form of the cost function is similar to equation 4, but with fewer parameters.

$$C_{reacquire} = f(|\hat{p}_i - \hat{p}_j|, |h_i - h_j|) \quad (6)$$

### 3.5 Activity Analysis

Using video in machine understanding has recently become a significant research topic. One of the more active areas is activity understanding from video imagery [Kanade *et al.*, 1997]. Understanding activities involves being able to detect and classify targets of interest and analyze what they are doing. Human motion analysis is one such research area. There have been several good human detection schemes, such as [Oren *et al.*, 1997] which use static imagery. But 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].

The VSAM IFD proposes the use of the "star" skeletonization procedure for analyzing the motion of targets [Fujiyoshi and Lipton, 1998] - particularly, human targets. The notion is that a simple form of skeletonization which 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 11).

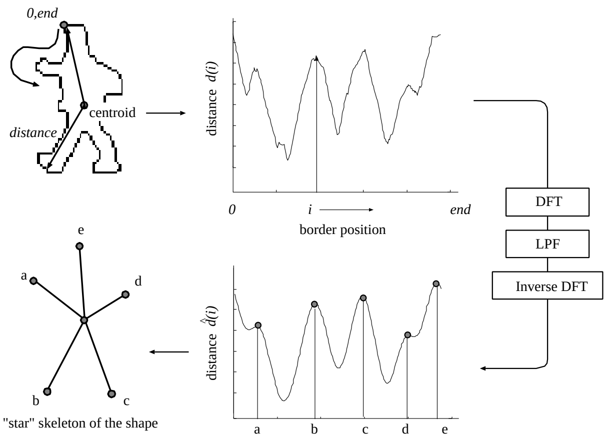



**Figure 11:** The boundary is "unwrapped" as a distance function from the centroid. This function is then smoothed and extremal points are extracted.

Figure 12 shows the skeletons of various objects. It is clear that while this form of skeletonization provides a sparse set of points, it can nevertheless be used to classify and analyze the motion of various different types of entity.

### 3.6 Human motion analysis

One technique often used to analyze the motion or gait of an individual target is the cyclic motion of skeletal components [Tsai *et al.*, 1994]. However, in this implementation, the knowledge of individual joint positions cannot be de-

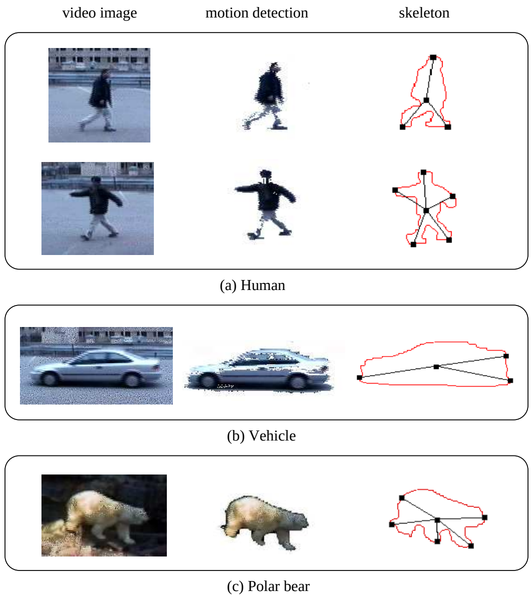



**Figure 12:** Skeletonization of different moving targets. It is clear the structure and rigidity of the skeleton is significant in analyzing target motion.

termined in real-time, so a more fundamental cyclic analysis must be performed. Another cue to the gait of the target is its posture. Using only a metric based on the "star" skeleton, it is possible to determine the posture of a moving human. Figure 13 shows how these two properties are simply extracted from the skeleton. The uppermost skeleton segment is assumed to represent the torso, and the lower left segment is assumed to represent a leg, which can be analyzed for cyclic motion.

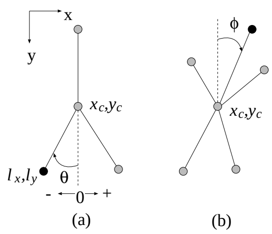



**Figure 13:** Determination of skeleton features. (a) $\theta$ is the angle the left cyclic point (leg) makes with the vertical, and (b) $\phi$ is the angle the torso makes with the vertical.

Figure 14 shows human target skeleton motion sequences for walking and running and the values of $\theta_n$ for the cyclic point. These data were acquired in real-time from a video stream with frame rate 8Hz.

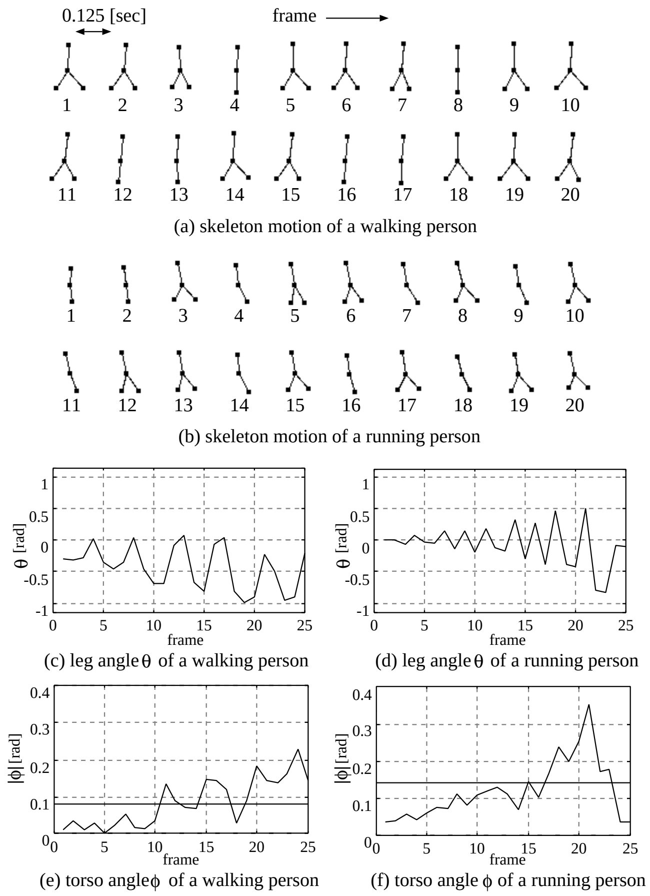



**Figure 14:** Skeleton motion sequences. Clearly, the periodic motion of $\theta_n$ provides cues to the target's motion as does the mean value of $\overline{\phi}_n$.

Comparing the average values $\overline{\phi}_n$ in figures 14(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 can also be used to determine human activity.

Another approach to classifying a moving object is to determine whether it is rigid or non-rigid by examining changes in its appearance over multiple frames [Wixson and Selinger, 1998]. This is most useful for distinguishing vehicles from humans and animals.

## 3.7 Airborne Surveillance

Fixed ground-sensor placement is fine for defensive monitoring of static facilities such as depots, warehouses or parking lots. In those cases, sensor placement can be planned in advance to get maximum usage of limited VSAM resources. However, the battlefield is a large and constantly shifting piece of real-estate, and it may be necessary to move sensors around in order to maximize their utility as the battle unfolds. While the airborne sensor platform directly addresses this concern, the self-motion of the aircraft itself introduces challenging video understanding issues.

### 3.7.1 Airborne Target Tracking

Target detection and tracking is a difficult problem from a moving sensor platform. The difficulty arises from trying to detect small blocks of moving pixels representing independently moving target objects when the whole image is shifting due to self-motion (also known as host motion). The key to our success with the airborne sensor is characterization and removal of host motion from the video sequence using the Pyramid Vision Technologies PVT-200 real-time video processor system. As new video frames stream in, the PVT processor registers and warps each new frame to a chosen reference image, resulting in a cancelation of pixel movement caused by host motion, and leading to a "stabilized" display that appears motionless for several seconds. Airborne target detection and tracking is then performed using three-frame differencing after using image alignment to register frame $I_{t-2}$ to $I_t$ and frame $I_{t-1}$ to $I_t$. This registration is performed at 30 frames/sec.

During stabilization, the problem of moving target detection from a moving platform is ideally reduced to performing VSAM from a stationary camera, in the sense that moving objects are readily apparent as moving pixels in the image. However, under real circumstances some of the remaining residual pixel motion is due to parallax caused by significant 3D scene structure. This is a subject of on-going research.

### 3.7.2 Camera Fixation and Aiming

It is well known that human operators fatigue rapidly when controlling cameras on moving airborne and ground platforms. This is because they must continually adjust the turret to keep it locked on a stationary or moving target. Additionally, the video is continuously moving, reflecting the ego-motion of the camera. The combination of these factors often leads to operator confusion and nausea. We have built upon image alignment techniques [Bergen *et al.*, 1992, Hansen *et al.*, 1994] to stabilize the view from the camera turret and used the same techniques to automate the camera control, thereby significantly reducing the strain on the operator. In particular, we use real-time image alignment to keep the camera locked on a stationary or moving point in the scene, and to aim the camera at a known geodetic coordinate for which reference imagery is available. More details can be found in [Wixson *et al.*, 1998], this proceedings.

Figure 15 shows the performance of the stabilization/fixation algorithm on two ground points as the aircraft traverses an approximate ellipse over them. The field of view in these examples is 3°, and the aircraft took approximately 3 minutes to complete each orbit.

(A) (B)

**Figure 15:** Fixation on target point A and on target point B. The images shown are taken 0, 45, 90 and 135 seconds after fixation was started. The large center cross-hairs indicate the center of the stabilized image, i.e. the point of fixation

### 3.7.3 Air Sensor Multi-Tasking

Occasionally, a single camera resource must be used to track multiple moving objects, not all of which fit within a single field of view. This problem is particularly relevant for high-altitude air platforms that must have a narrow field of view in order to see ground targets at a reasonable resolution. Sensor multi-tasking is employed to switch the field of view periodically between two (or more) target areas that are being monitored. This process is illustrated in Figure 16 and described in detail in [Wixson *et al.*, 1998].

## 4 Site Modeling

We have used site models in the VSAM program both to improve the human-computer interface and to enable various tracking capabilities such as inferring target positions and camera visibility. These models have included automatically-generated image mosaics, USGS orthophotos, Digital Elevation Models (DEMs), and CAD models in VRML.

The OCU site model contains 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 some specific VSAM capabilities:

• The site model must primarily support OCU workstation graphics that allow the operator to visualize the whole site and quickly comprehend geometric relationships between sensors,

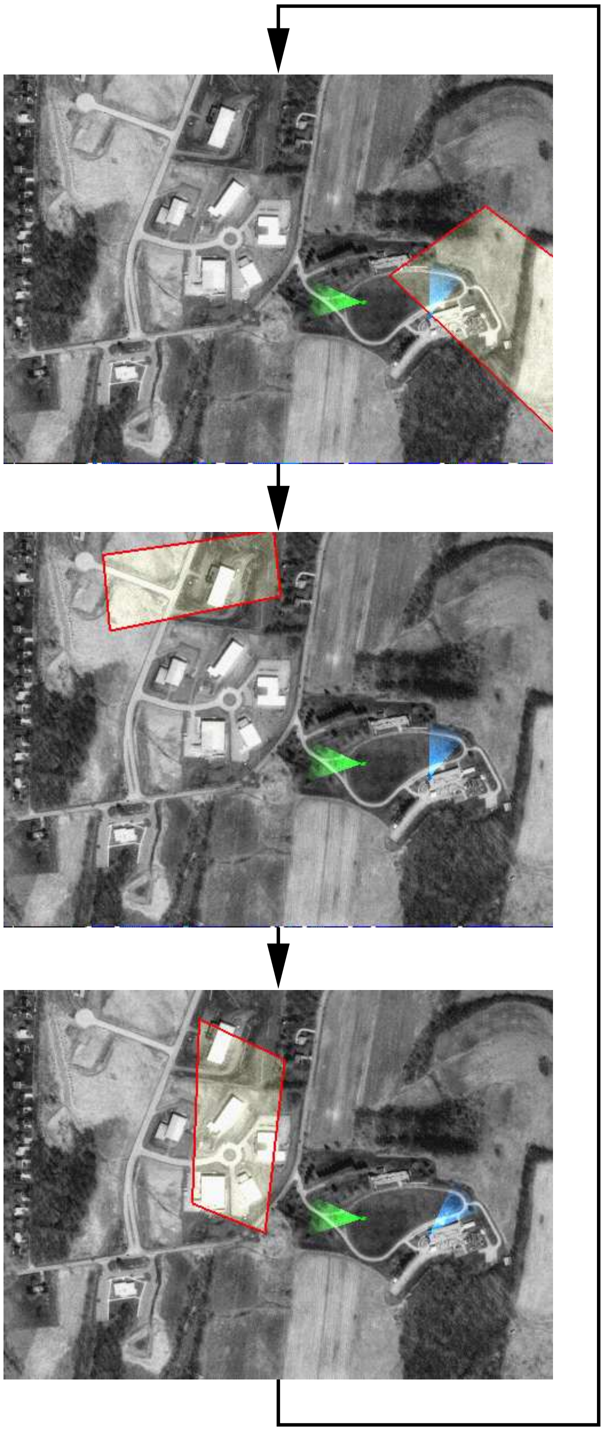


**Figure 16:** Footprints of airborne sensor being autonomously multi-tasked between three disparate geodetic scene coordinates.

targets, and scene features.

• The site model provides a geometric context for VSAM. For example, we might directly task a sensor to monitor the door of a building for people coming in and out, or specify that vehicles should appear on roads.

• A 3D site model supports visibility analysis (predicting what what portions of the scene are visible from what sensors( so that sensors can be efficiently tasked.

• An accurate 3D site model supports target geopositioning via intersection of viewing rays with the terrain.

### 4.1 Coordinate Systems

Two geospatial site coordinate systems are used interchangeably within the VSAM testbed. The WGS84 geodetic coordinate system provides a reference frame that is standard, unambiguous and global (in the true sense of the word). Unfortunately, even simple computations such as the distance between two points become complicated as a function of latitude, longitude and elevation. For this reason, geometric processing is performed within a site-specific Local Vertical Coordinate System (LVCS) [ASP, 1980]. An LVCS is a Cartesian system oriented so that the positive X axis points east, positive Y points true north, and positive Z points up. All that is needed to completely specify an LVCS is the 3D geodetic coordinate of its origin point. Conversion between geodetic and LVCS coordinates is straightforward, so that each can be used as appropriate to a task.

### 4.2 Model Representations

Figure 17 illustrates the wide variety of site model representations that have been used in either the 1997 IFD VSAM demo system or the 1998 testbed system.

**A) USGS orthophoto.** The United States Geological Survey (USGS) produces several digital mapping products that can be used to create an initial site model. These include

• Digital Orthophoto Quarter Quad (DOQQ) -

a nadir (down-looking) image of the site as it would look under orthographic projection (Figure 17A). The result is an image where scene features appear in their correct horizontal positions.

• Digital Elevation Model (DEM) - an image whose pixel values denote scene elevations at the corresponding horizontal positions. Each grid cell of the USGS DEM shown encompasses a 30-meter square area.

• Digital Topographic Map (DRG) - a digital version of the popular USGS topo maps.

• Digital Line Graph (DLG) - vector representations of public roadways and other cartographic features. Many of these can be ordered directly from the USGS EROS Data Center web site, located at URL `http://edcwww.cr.usgs.gov/`. The ability to use existing mapping products from USGS or National Imagery and Mapping Agency (NIMA) to bootstrap a VSAM site model demonstrates that rapid deployment of VSAM systems to monitor trouble spots around the globe is a feasible goal.

**B) Custom DEM**

The Robotics Institute autonomous helicopter group has mounted a high precision laser range finder onto a remote-control Yamaha helicopter. This was used to create a high-resolution (half-meter grid spacing) DEM of the Bushy Run site for VSAM DEMO I (Figure 17B). Raw radar returns were collected with respect to known helicopter position and orientation (using on-board altimetry data) to form a cloud of points representing returns from surfaces in the scene. These points were converted into a DEM by projecting into LVCS horizontal-coordinate bins, and computing the mean and standard deviation of height values in each bin.

**C) Mosaics.** A central challenge in surveillance is how to present sensor information to a human operator [Miller and Amidi, 1998]. The relatively narrow field of view presented by each sensor makes it very difficult for the operator to maintain a sense of context that enables him or her to know just what lies outside the camera's immediate image. Image mosaics from moving cameras overcome this problem by providing extended views of regions swept over by the camera. Figure 17C displays an aerial mosaic of the Demo I Bushy Run site. The video sequence was obtained by flying over the demo site while panning the camera turret back and forth and keeping the camera tilt constant[Hansen *et al.*, 1994, Sawhney and Kumar, 1997, Sawhney *et al.*, 1998]. The VSAM IFD team also demonstrated coarse registration of this mosaic with a USGS orthophoto using a projective warp to determine an approximate mapping from mosaic pixels to geographic coordinates. It is feasible that this technology could lead to automated methods for updating existing orthophoto information using fresh imagery from a recent fly-through. For example, seasonal variations such as fresh snowfall (as in the case of VSAM Demo I) can be integrated into the orthophoto.

**D) VRML models.** Figure 17D shows a VRML model of one of the Bushy Run buildings and its surrounding terrain. This model was created by the $K^2T$ company using the factorization method [Tomasi and Kanade, 1992] applied to aerial and ground-based video sequences. Another use of VRML models in the VSAM system is to display spherical mosaics with the user immersed at the location of the focal point (more on this below).

**E) Compact Terrain Data Base (CTDB).** Recently, we have begun to use the Compact Terrain Data Base (CTDB) to represent full site models. The CTDB was originally designed to represent large expanses of terrain within the context of advanced distributed simulation, and has been optimized to efficiently answer geometric queries such as finding the elevation at a point in real-time. Terrain can be represented as either a grid of elevations, or as a Triangulated Irregular Network (TIN), and hybrid data bases containing both representations are allowed. The CTDB also represents relevant cartographic features on top of the terrain skin, including buildings, roads, bodies of water, and tree canopies. Figure 17E shows a small portion of the Schenley Park / CMU campus CTDB currently being generated for the 1998 VSAM demo. In addition to determining object geolocation by intersecting viewing rays with the ground, we are using the CTDB to perform oc-

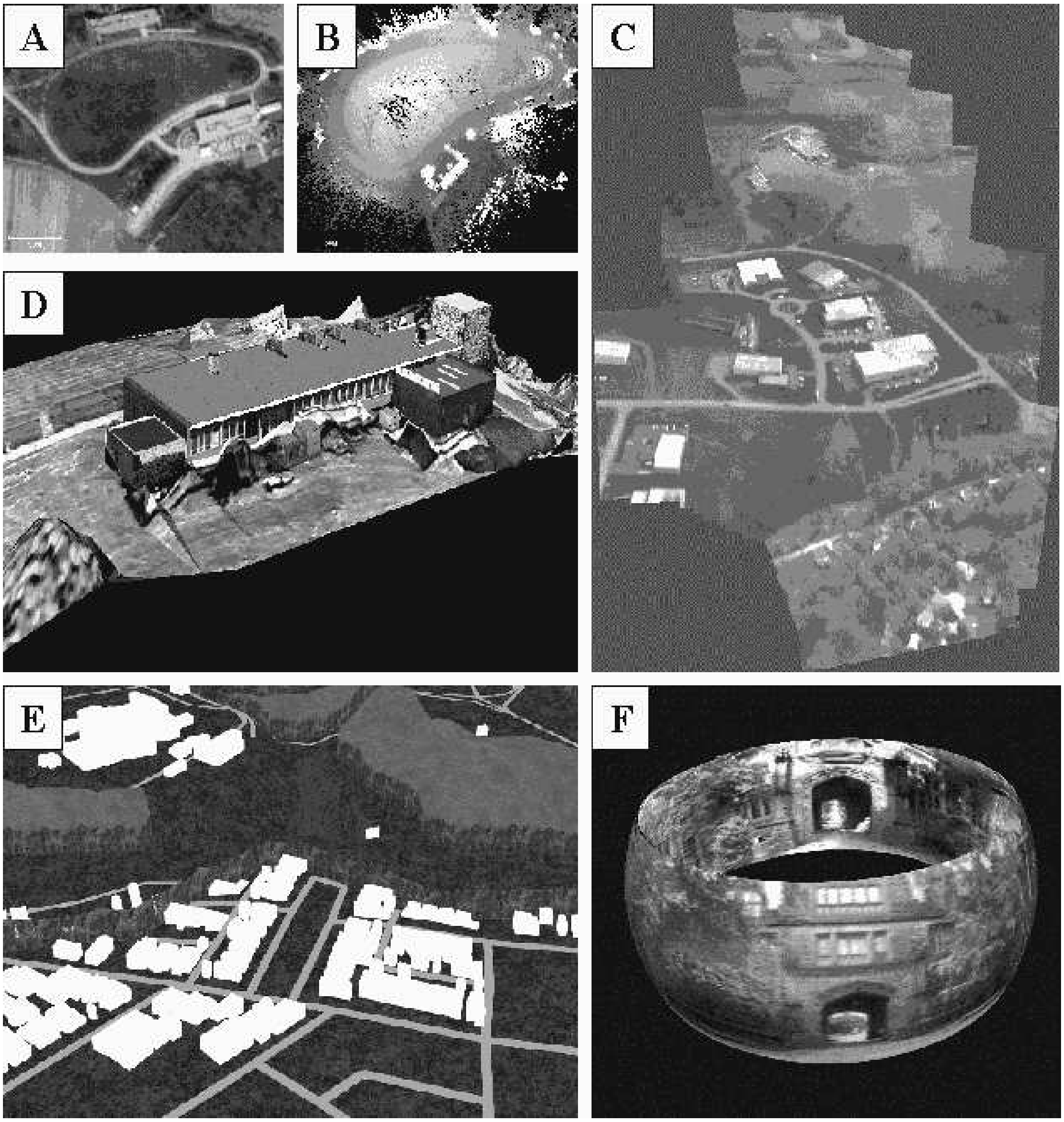



**Figure 17:** A variety of site model representations have been used in the VSAM IFD testbed system: A) USGS orthophoto; B) custom DEM; C) aerial mosaic; D) VRML model; E) CTDB site model; and F) spherical representations.

clusion analysis by determining inter-visibility of one point from the viewpoint of another. Another important benefit to using CTDB as a site model representation for VSAM processing is that it allows us to easily interface with synthetic environment tools ModSAF and ModStealth.

**F) Spherical Representations.**

Everything that can be seen from a stationary camera can be represented on the surface of a viewing sphere [Adelson and Bergen, 1991]. This is true even if the camera is allowed to pan and tilt about the focal point, and to zoom in and out – the image at any given (pan,tilt,zoom) setting is essentially a discrete sample of the bundle of light rays impinging on the camera's focal point. This year, the VSAM IFD team is making use of this fact to design camera-specific representations of the geometry and photometry of the scene (Figure 17F). Spherical lookup tables are being built for each fixed-mount SPU to precompile and store the 3D locations and surface material types of the points of intersection of that camera's viewing rays with the CTDB site model. Spherical mosaics are being produced in real-time to provide an extended view of what can potentially be seen from the camera, again to provide a better sense of context to the human observer. Figure 18 displays a spherical mosaic constructed by panning and tilting a stationary camera mounted on a rooftop at the VSAM Demo II site. This representation can be displayed using a VRML or RealSpace viewer with the observer situated at the center of the sphere for a realistic surround-video effect. The ultimate goal is to display extracted moving entities in real-time superimposed on this extended spherical background.

## 4.3 Model-based Geolocation

One example of model-based VSAM is computation of target geolocation from a monocular view. 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 tra-

**Figure 18:** Spherical mosaic from a camera at the VSAM Demo II site.

jectories 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.

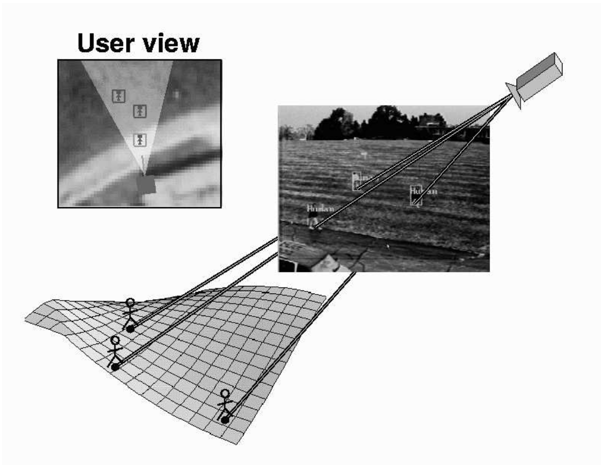



**Figure 19:** Estimating object geolocations by intersecting target viewing rays with a terrain model.

Determining target trajectories 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 19). 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, the VSAM testbed is designed for much larger scene areas that may contain significantly varied terrain. We perform geolocation using ray intersection with full terrain models provided by either digital elevation maps, or the compact terrain database. See [Collins *et al.*, 1998] in this proceedings for more details.

## 5 Human-Computer Interface

Keeping track of people, vehicles, and their interactions, over a chaotic area such as the battlefield, is a difficult task. The commander 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 visualization of the battlefield by using 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 the target is no longer tied to the original resolution and viewpoint of the video sensor, since a synthetic replay of the dynamic events can be constructed using high-resolution, texture-mapped graphics, from any perspective. Particularly striking is the amount of data compression that can be achieved by transmitting only symbolic georegistered target information back to the operator control unit instead of raw video data. Currently, we can 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 sensor 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.

### 5.1 Benning MOUT Site Experiment

Ultimately, the key to comprehending large-scale, multi-agent events is a full, 3D immersive visualization that allows the human operator to fly at will through the environment to view dynamic events unfolding in real-time from any viewpoint. A geographically accurate 3D model-based visualization can give the commander a comprehensive overview of the battlefield by displaying people and vehicles dynamically interacting in their proper spatial relationships to each other and to 3D cartographic features such as buildings and roads.

We envision a graphical user interface based on cartographic modeling and visualization tools developed within the Synthetic Environments (SE) community. The site model used for model-based VSAM processing and visualization is represented using the Compact Terrain Database (CTDB). Targets are inserted as dynamic agents within the site model and viewed by Distributed Interactive Simulation clients such as the Modular Semi-Automated Forces (ModSAF) program and the associated 3D immersive ModStealth viewer. We have already demonstrated proof-of-concept of this idea at the Dismounted Battle Space Battle Lab (DBBL) Simulation Center at Fort Benning Georgia as part of the April 1998 VSAM workshop. On April 13, researchers from CMU set up a portable VSAM system at the Benning Mobile Operations in Urban Terrain (MOUT) training site. The camera was set up at the corner of a building roof whose geodetic coordinates had been measured by a previous survey [GGB, 1996], and the height of the camera above that known location was measured. The camera was mounted on a pan-tilt head, which in turn was mounted on a leveled tripod, thus fixing the roll and tilt angles of the pan-tilt-sensor assembly to be zero. The yaw angle (horizontal orientation) of the sensor assembly was measured by sighting through a digital compass. After processing several troop exercises, log files containing camera calibration information and target hypothesis data packets were sent by FTP back to CMU and processed using the CTDB to determine a time-stamped list of moving targets and their geolocations. Later in the week, this information was brought back to the DBBL Simulation Center at Benning where, with the assistance of colleagues from BDM, it was played back for VSAM workshop attendees using custom software that broadcast time-sequenced simulated entity packets to the network for display by both ModSAF and ModStealth. Some processed VSAM video data and screen dumps of the resulting synthetic environment playbacks were shown previously in Figure 20.

### 5.2 3D Immersive VSAM

The Fort Benning experiment demonstrated that it is possible to automatically detect and track multiple people and vehicles from a VSAM system and insert them as dynamic actors in a synthetic environment for after-action review. We are proposing that, with some modifications, this process can also form the basis for a real-time immersive visualization tool. First, we are currently porting object geolocation computation using the CTDB onto the VSAM sensor processing unit platforms. Estimates of target geolocation will be computed within the frame-to-frame tracking process and will be transmitted in data packets back to the operator control workstation. Secondly, at the operator workstation, incoming object identity and geolocation data packets will be repackaged on the fly into Distributed Interactive Simulation (DIS) packets that are understood by ModSAF and ModStealth clients. At that point, targets will be viewable within the context of the site model by the 3D ModStealth viewer.

A real-time immersive visualization system would allow us to explore several experimental human-machine interface questions, the most important of which is: can the user get a feel for what activity is taking place purely by viewing events within the synthetic environment? If not, what additional information, such as infrequently updated images or very low-resolution video, is necessary to complete the picture? Can the operator effectively task troops to intercept a moving target on the basis of what he sees in the synthetic environment? Can the 3D synthetic environment be used to interactively control a multi-sensor VSAM system? (For example, the operator could be placed at the same location of the sensor and allowed to rotate their viewpoint within the simulated environment – thereby teleoperating the camera pan-tilt unit to point in the same direction.) What might be a good approach to specifying 3D regions of interest while immersed within a simulated environment? We expect to answer some of these questions within our third year research program.

## 6 VSAM Demonstrations

### 6.1 Demo I : Bushy Run

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 consisted of an OCU with two ground-based and one airborne SPU. The demo successfully highlighted the following technical achievements:

• **Modeling.** A site model was generated using a USGS orthophoto in combination with a high resolution DEM generated using an airborne laser range-finder. At the time of the demo, an aerial scene mosaic was generated from the airborne sensor and geo-registered with the orthophoto using a projective warp to

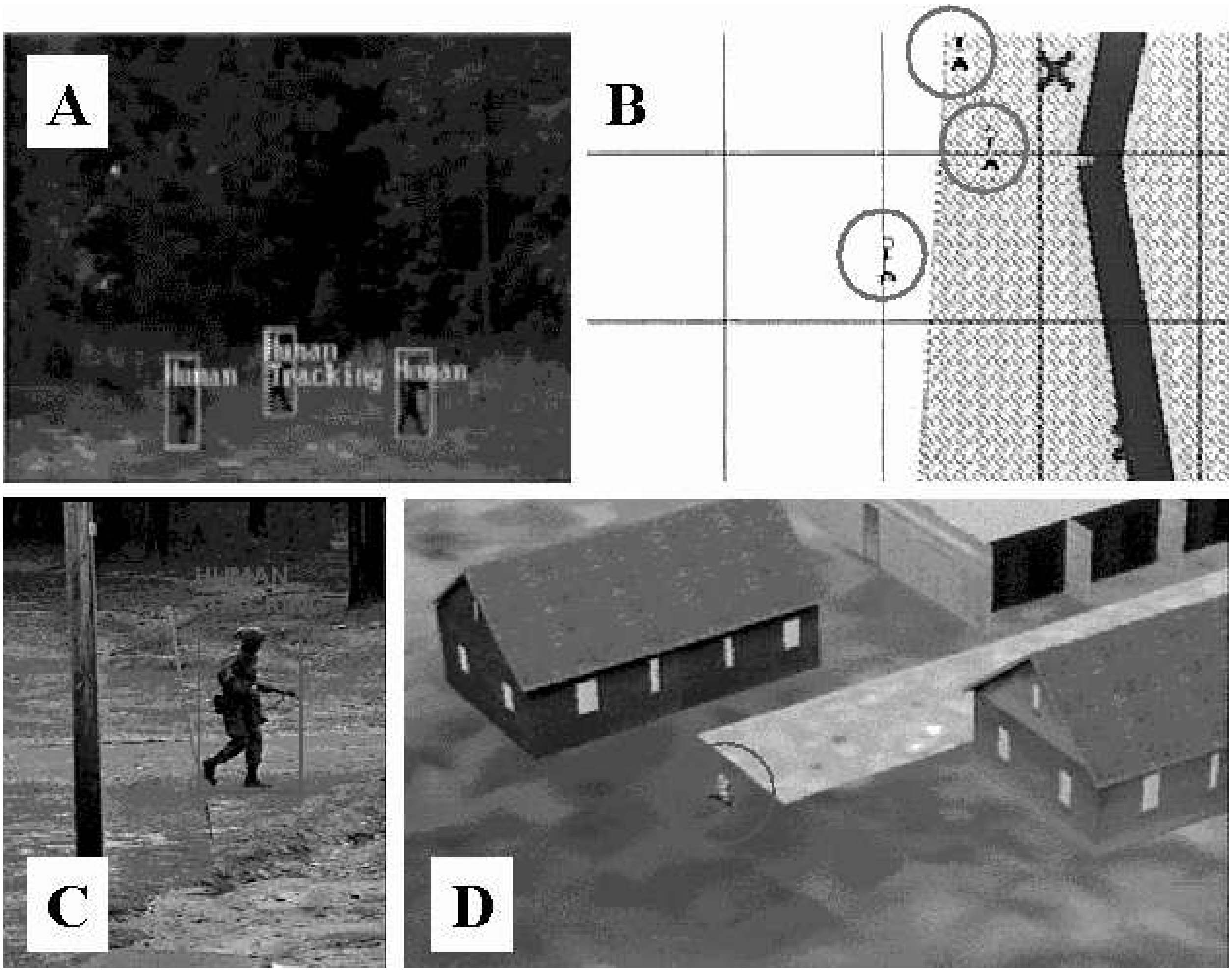



**Figure 20:** Sample synthetic environment visualizations of data collected at the Benning MOUT site. A) Automated tracking of three human targets. B) ModSAF 2D orthographic map display of estimated geolocations. C) Tracking of a soldier walking out of town. D) Immersive, texture-mapped 3D visualization of the same event, seen from a user-specified viewpoint.

hand-selected tie points.

• **Ground SPUs.** 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 is shown in Figure 21.

• **Air SPU.** The airborne platform showcased real-time, frame-to-frame affine motion estimation and image warping using the Sensar VFE-100 video processing system (precursor to PVT-200). Specific capabilities demonstrated were fixating on a designated ground point while compensating for airplane movement and vibration using electronic stabilization and active camera control, multi-tasking the airborne sensor to servo between multiple designated areas

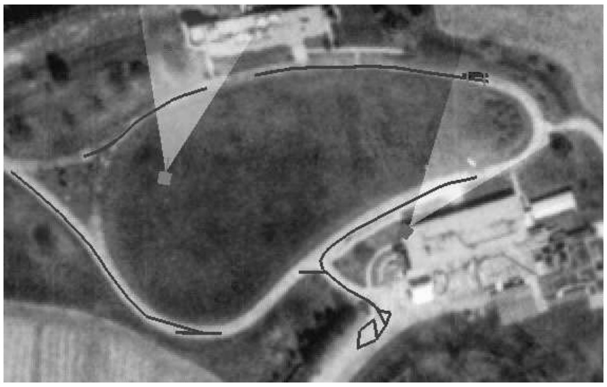


**Figure 21:** Synopsis of vehicle trajectory during the Bushy Run demo.

so that the human operator could monitor several regions simultaneously, and limited moving target detection in open areas using temporal differencing on electronically stabilized video frames.

• **OCU.** The operator control unit featured a 2D graphical user interface that displayed all sensor positions, fields of view, and target locations overlayed on the orthophoto, along with live microwave video feeds from all sensors (shown previously in Figure 5). The GUI also allowed a human operator to task the ground sensors by selecting regions of interest and locations for sensor hand-off, and to directly control sensor pan/tilt and various control algorithm parameters.

## 6.2 Demo II : CMU

Demo II will be held on the campus of CMU, a much more urban environment than either Bushy Run or the Benning MOUT site. Figure 22 shows the placement of sensors and the OCU. IFD sensor assets include five fixed-mount pan-tilt-zoom cameras and the Islander airborne sensor. These will be augmented with two FRE sensor packages provided by LeHigh-Columbia (ParaCamera) and TI (video alarm system). These sensors will all cooperate to track a vehicle from nearby Schenley Park onto campus, follow its path as it winds through campus and stops at the OCU building, alert the operator when the vehicle's occupants attempt to break into the building, and follow the ensuing

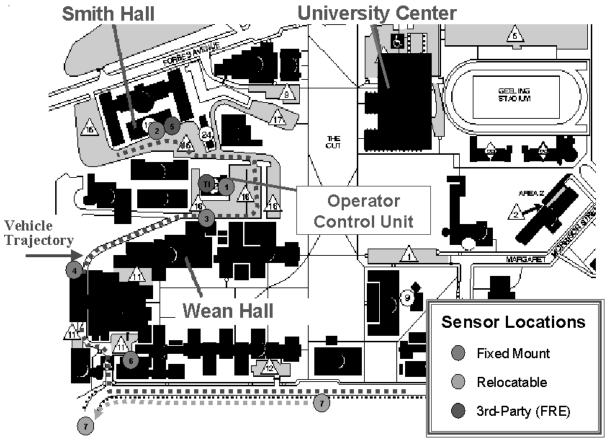



**Figure 22:** Overview of sensor placement and OCU location for VSAM Demo II, to be held on the campus of CMU on October 8, 1998.

car and foot chases as the vehicle and its occupants attempt to flee from the police. Some preliminary moving object detection, tracking and geolocation results from the VSAM Demo II testbed system were shown in Figure 9. The testbed system and current video understanding technologies were presented in Sections 2 and 3.

## 6.3 Demo III : CMU

The VSAM IFD effort has been funded for a third year. Demo III will also be held on the CMU campus, to take advantage of the VSAM infrastructure now in place. This section outlines planned goals for the third year effort.

**System Architecture:** A second OCU with additional cameras will be added to the far side of campus to begin exploring issues in distributed VSAM. One or two thermal sensors will be installed to provide continuous day/night surveillance and to gain experience in target classification and recognition from thermal imagery.

**Sensor Control:** We will add 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.

**Video Understanding:** We will develop improved capabilities for doing VSAM while in motion, including tracking targets while the sensor is panning and zooming, and performing real-time spherical mosaic background subtraction. Improved representation and classification of targets based on edge gradients, surface markings and internal motion will be explored, as well as analysis of target behaviors using gait analysis, crowd flow analysis, and detection of human-vehicle interactions. We will also use domain knowledge of road networks to predict trajectories of vehicles.

**User Interaction:** Year 3 will see the CTDB campus site model fully integrated into the VSAM testbed system. We will also pursue the goal of 3D immersive graphical user interfaces for operator visualization and sensor tasking to the operator control workstation. Finally, year 3 will see the advent of Web-VSAM – remote site monitoring and SPU control over the internet using a JAVA DIS client. Potential sites being evaluated for a VSAM web-cam are the CMU campus and the Benning MOUT site.

## Acknowledgments

The authors would like to thank the U.S. Army Night Vision Lab team at Davison Airfield, Ft. Belvoir, Virginia for their help with the airborne operations. We would also like to thank Chris Kearns and Andrew Fowles for their assistance at the Fort Benning MOUT site, and Steve Haes and Joe Findley at BDM/TEC for their help with the the CTDB site model and distributed simulation visualization software.

## References

[Adelson and Bergen, 1991] E.H. Adelson and J.R. Bergen. The plenoptic function and the elements of early vision. In Michael Landy and J.Anthony Movshon, editors, *Computational Models of Visual Processing, Chapter 1.* The MIT Press, Cambridge MA., 1991.

[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.

[ASP, 1980] American Society of Photogrammetry ASP. *Manual of Photogrammetry.* Fourth Edition, American Society of Photogrammetry, Falls Church, 1980.

[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.

[Bergen *et al.*, 1992] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, 1992.

[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.

[Bregler, 1997] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proceedings of IEEE CVPR 97*, pages 568–574, 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.

[GGB, 1996] Geometric Geodesy Branch GGB. *Geodetic Survey.* Publication SMWD3-96-022, Phase II, Interim Terrain Data, Fort Benning, Georgia, May 1996.

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

[Hansen *et al.*, 1994] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt. Real-time scene stabilization and mosaic construction. In *Proc. Workshop on Applications of Computer Vision*, 1994.

[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.

[IST, 1994] Institute for Simulation & Training IST. *Standard for Distributed Interactive Simulation – Application Protocols, Version 2.0.* University of Central Florida, Division of Sponsored Research, March 1994.

[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.*, 1997] Takeo Kanade, Robert Collins, Alan Lipton, P. Anandan, and Peter Burt. Cooperative multisensor video surveillance. In *Proceedings of DARPA Image Understanding Workshop*, volume 1, pages 3–10, May 1997.

[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.

[Miller and Amidi, 1998] Ryan Miller and Omead Amidi. 3-d site mapping with the cmu autonomous helicopter. In *Proc. 5th International Conference on Intelligent Autonomous Systems*. Sapporo, Japan, June 1998.

[Oren *et al.*, 1997] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proceedings of IEEE CVPR 97*, pages 193–199, 1997.

[Sawhney and Kumar, 1997] H. S. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[Sawhney *et al.*, 1998] H. S. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Proc. European Conference on Computer Vision*, 1998.

[Tomasi and Kanade, 1992] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams: factorization method. *International Journal of Computer Vision*, 9(2), 1992.

[Tsai *et al.*, 1994] P. Tsai, M. Shah, K. Ketter, and T. Kasparis. Cyclic motion detection for motion based recognition. *Pattern Recognition*, 27(12):1591–1603, 1994.

[Wixson and Selinger, 1998] L. Wixson and A. Selinger. Classifying moving objects as rigid or non-rigid. In *Proc. DARPA Image Understanding Workshop*, 1998.

[Wixson *et al.*, 1998] L. Wixson, J. Eledath, M. Hansen, R. Mandelbaum, and D. Mishra. Image alignment for precise camera fixation and aim. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 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.