Department of Electrical
and
Computer Systems Engineering

Technical Report
MECSE-30-2004

Automatic Video Surveillance and Robotic Early Response: An Evaluation of ObjectVideo VEW

Geoffrey Taylor

MONASH
UNIVERSITY

Automatic Video Surveillance and Robotic Early Response: An Evaluation of ObjectVideo VEW

Geoffrey Taylor
ARC Centre for Perceptive and Intelligent Machines in Complex Environments
Monash University 3800 VIC, Australia
Geoffrey.Taylor@eng.monash.edu.au

Abstract

ObjectVideo VEW (Video Early Warning) is a commercial surveillance system that automatically analyses real-time video to detect security events based on user-defined rules. The aim of this study is to firstly evaluate the performance of VEW in typical applications, and secondly to extend the automatic surveillance concept by triggering a robotic early response to security events. The evaluation of VEW covers various user-defined rules in both indoor and outdoor environments with several different cameras, including wide-angle and omnidirectional sensors. Surveillance scenarios include scenes with crowds, vehicles and attempts at deliberate deception. To extend VEW to robotic applications, we first develop a visual servo controller that allows a mobile robot to be driven to any location in an image using only visual feedback from an off-board camera. A VEW rule is then constructed to detect an unattended object, and the resulting alert triggers the robot to intercept the target. Such a system could provide an automatic early response for airport security and other immediate applications.

Contents

1 Introduction

Automatic video surveillance (AVS) promises to improve the effectiveness of surveillance camera networks by augmenting the role of the human observer. Video footage is continuously analyzed using computer vision and artificial intelligence techniques to determine the behaviour of visual targets and alert the operator to suspicious events. Benefits of AVS include higher detection rates, greater coverage with fewer operators, and the ability to automatically index logged data to facilitate forensic queries. AVS has immediate relevance to several important applications, which include perimeter security at borders and coastlines, detecting unattended baggage in airports and train stations and identifying suspicious behaviour near key infrastructure.

The main computer vision problems in AVS are the detection, classification and tracking of visual targets. Detection is usually based on some form of adaptive background subtraction [16, 18], assuming a static camera. Classification is the problem of attaching semantic labels to objects to aid in the interpretation of meaningful events (such as a person leaving a vehicle). A wide variety of classification algorithms using a range of image features and pattern matching techniques have been demonstrated in previous work [7, 8, 12]. Finally, tracking is the problem of determining spatio-temporal behaviour, and is complicated by occlusions, changes in lighting and appearance, erratic motion and other distractions. Various tracking algorithms have also been demonstrated, ranging from the use of *a priori* models [11] to data-driven features [5]. A good review of the computer vision issues in AVS can be found in [6].

The level of maturity reached by computer vision has recently resulted in a push to transfer the technology from the research domain into commercial markets. Large-scale AVS projects sharing this goal include the European PASSWORDS (Parallel and real time Advanced Surveillance System With Operator assistance for Revealing Dangerous Situations) project [3], the DARPA funded VSAM[1] (Video Surveillance And Monitoring) project at several institutions including MIT and CMU [4], and the Video Surveillance and Analysis Project[2] at the Australian CRC for Sensor Signal and Information Processing [9]. ObjectVideo VEW (Video Early Warning) is a commercial product based on VSAM research[3] and represents state-of-the-art commercial AVS. One of the aims of this research is to evaluate the performance of commercial AVS using the small-scale ObjectVideo installation at the ARC Centre for Perceptive and Intelligent Machines in Complex Environments (PIMCE), Monash University.

In addition to AVS systems, considerable research effort has been directed towards the development of security robots. These systems are sometimes developed as fully autonomous robots with a rich set of multimodal sensors and the ability to detect and respond to security threats, providing an alternative to fixed surveillance cameras [13, 15]. Other systems employ human-operated security robots as a teleoperated extension to conventional security infrastructure [2]. In this report, we explore the possibility of integrating autonomous robots with AVS to gain the benefits of both wide surveillance coverage and an early robotic response without requiring human intervention. In addition, it will be shown that this approach can utilize robots with minimal on-board sensing by exploiting the network of fixed surveillance sensors.

The remainder of this report is divided into three parts. Section 2 provides a brief overview of the ObjectVideo VEW architecture and operation. The performance of VEW in a variety of surveillance scenarios is qualitatively evaluated and discussed in

[1] See http://www-2.cs.cmu.edu/ vsam/index.html
[2] See http://www.cssip.edu.au/research/VSAProject.html
[3] See http://www.objectvideo.com

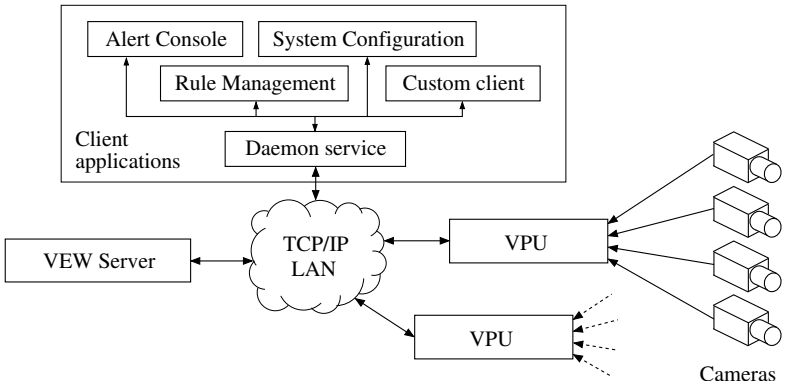


Figure 1: ObjectVideo VEW architecture.

Section 3. Finally, a robotic early response extension to AVS is described and demonstrated in Section 4.

2 Overview of ObjectVideo VEW

The architecture of ObjectVideo VEW is illustrated in Figure 1. ObjectVideo markets three versions of the system: VEW Standard, VEW HiRes and VEW FlowControl (this actually refers to the type of VPU installed, as described below). Standard and HiRes differ only in the resolution at which video is analyzed, while FlowControl introduces algorithms for analysing the movement of crowds. The following discussion refers mainly to VEW Standard, which is the version installed at PIMCE.

VEW is comprised of both hardware and software components, both of which are designed to be highly scalable. Automatic video analysis is performed by the Video Processing Units (VPUs), each of which can process up to four cameras simultaneously at a resolution of 320x240 pixels. In addition to live feeds, the VPU inputs can be individually configured to accept stored video footage (AVI files), which is helpful for debugging and evaluation. The ObjectVideo Server manages communication between the VPUs and client software, and logs security alerts in a central database. A single Server manages up to 80 cameras (20 VEW Standard VPUs), each of which can be programmed to respond to different security events. The Server and VPUs communicate over a standard TCP/IP network. The installation at PIMCE is comprised of a single VPU and ObjectVideo Server.

The software components of the system include the System Configuration, Alert Console and Rule Management applications. Custom client applications can also be created using the ObjectVideo Client Application Programming Interface (API) for the Microsoft .NET Framework, which is important for the robot early response extension described in Section 4. Client applications communicate with the Server through a Windows component called the ObjectVideo Daemon service. The System Configuration tool is used for system related tasks during installation, the Alert Console displays real-time security alerts and provides database search tools, and the Rule Management tool allows views and rules to be created and managed as described below. *Rules* define the conditions that must be satisfied for a particular security event, and are central to the VEW architecture.

The VPUs detect potential objects of interest using statistical background modeling techniques. Each target is tracked on the image plane, and simultaneously classified as a *person*, *vehicle*, *unknown* or *transient* object based on appearance and motion. Unknown objects are those that cannot be classified as either person or vehicle, and transient objects appear too briefly or are too small for reliable classification. VEW rules can be defined for a particular object class (such as people) or all classes.

Since VEW employs statistical background modeling, the system requires a *view* (or background model) to be defined for each static camera before creating rules. Several views may be defined for different static poses of a pan/tilt/zoom camera, and the system automatically selects the appropriate view for the current camera settings. Once a view is initialized, rules can be defined to detect security events. A rule is composed of an *event* such a person entering a region of interest, a *schedule* to indicate when the rule is applied, and a *response* such as an email and/or Alert Console message. When an event is detected, VEW creates an *alert image* with annotations for the target and rule parameters, and archives the event in the Server database. The following events are supported by VEW Standard:

Tripwire: triggered when an object crosses a line (drawn on the image plane by the user) in a particular direction.

Multi-line tripwire: triggered when an object crosses two user-defined lines in succession within a threshold time.

Partial view: triggered when an object enters, exits, appears, disappears, loiters, is left behind or removed from a polygonal sub-region of the image plane.

Full view: similar to a partial view, but applied to the entire image.

Scene change: triggered for any significant change in the field of view, such as the lights being turned off or a camera malfunction.

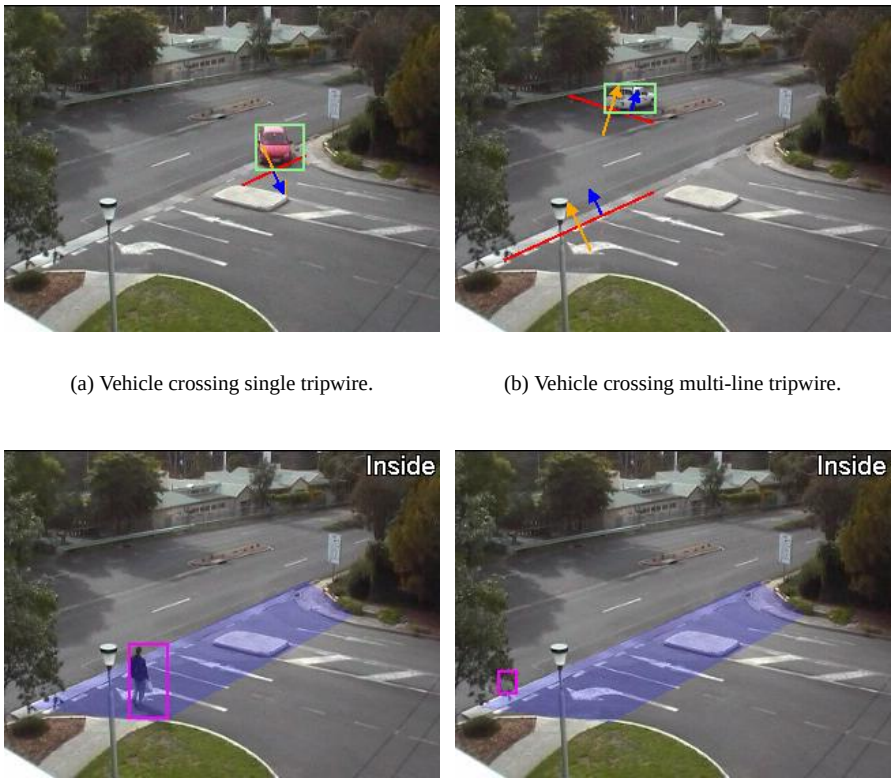
One of the difficulties with motion-based detection in AVS is the possibility of false alarms due to swaying branches, wave on water, reflections, stray animals and other spurious distractions. To reduce false alarms, VEW allows rules to be supplemented with *object filters* to eliminate targets that do not satisfy a maximum size, minimum size, maximum rate of change in size, consistent shape and consistent direction of motion. The maximum and minimum size filters provide a mechanism to scale the threshold depending on the distance along the ground plane of the target from the camera. Object filters are defined for a particular view, and can be applied to all or a subset of the associated rules.

3 ObjectVideo VEW in Surveillance Applications

The following sections describe a series of experiments that were carried out with several different cameras in a variety of conditions to evaluate the performance of ObjectVideo VEW. It should be noted that the results presented here are qualitative in nature, and serve to highlight the capabilities and weaknesses of the system.

3.1 Car Park Entrance Sequence

The first sequence represents a typical surveillance scenario involving a static security camera overlooking the entrance to a multi-level car park. The sequence was captured



(a) Vehicle crossing single tripwire.

(b) Vehicle crossing multi-line tripwire.

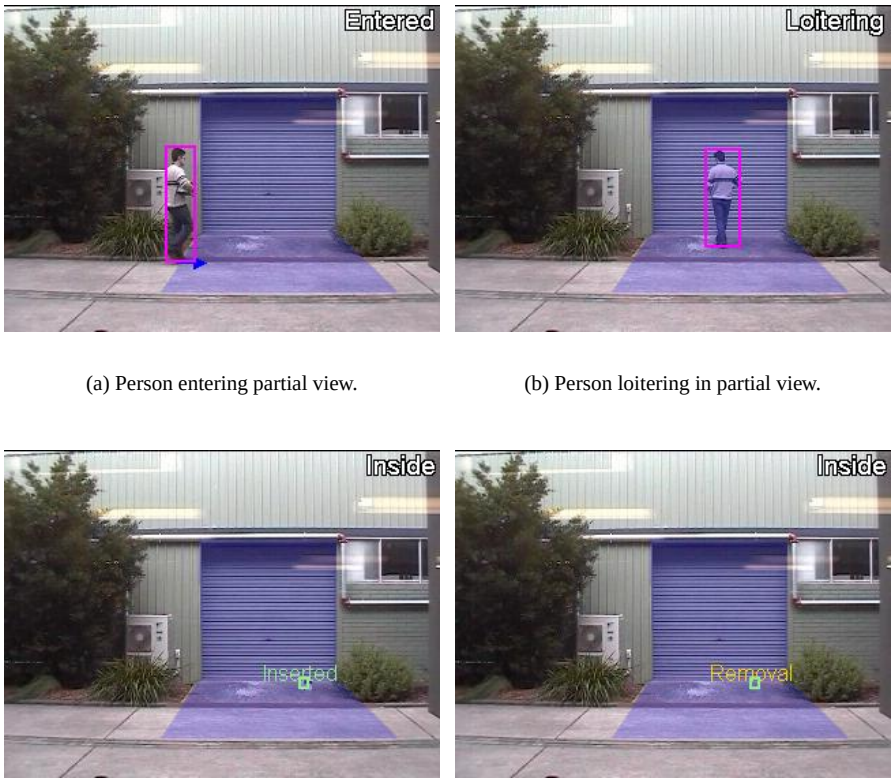
(c) Person entering partial view.

(d) False alarm caused by swaying tree.

Figure 2: Alert images from car park video sequence.

off-line with a tripod-mounted handycam, before digitizing and playing the footage through the VPU masqueraded as a live camera. Discrete events included pedestrians crossing the road and cars entering and leaving the lot. While these events were not particularly suspicious, they nevertheless test the ability of VEW to detect common events in the presence of distractions such as swaying trees and shadows.

Three rules were created, and the resulting alert images are shown in Figure 2. In Figure 2(a) a car trips a single tripwire upon entering the car park, while Figure 2(b) shows a car tripping a multi-line tripwire while leaving. The complete video sequence shows three different cars entering and two leaving the car park, and VEW successfully detects each event. Figure 2(c) shows a third rule triggered by a pedestrian entering the polygonal partial view covering the car park entrance. While VEW successfully detects this event, Figure 2(d) shows the same rule producing a false positive, triggered by the swaying branch of a tree. This false positive can be eliminated using minimum size and consistent motion filters provided by VEW. In any case, it should be noted that the partial view alert was never triggered by a car; VEW successfully discriminated between vehicles and people for every legitimate target.



(a) Person entering partial view.

(b) Person loitering in partial view.

(c) Object left in partial view.

(d) Object removed from partial view.

Figure 3: Alert images from suspicious package sequence.

3.2 Suspicious Package Sequence

This sequence was also captured off-line with a tripod-mounted handycam, before digitizing and feeding the footage through the VPU. Unlike the previous sequence, this case is a deliberately suspicious scenario involving a person loitering near a door, placing a package on the ground, leaving the area and finally returning to remove the package. Rules were initiated to detect a person entering, leaving, or loitering in a partial view surrounding the door, and for anything left behind or removed from the same area. Figure 3 shows the alert images generated by VEW, which successfully detected all key events in the sequence (some alerts have been omitted from Figure 3 for conciseness). While this sequence was largely unchallenging in terms of distractions and occlusions, it nevertheless represents a realistic scenario in which the perpetrator targets a secluded location in order to avoid attention.

3.3 Wide Angle and Omnidirectional Cameras

This section evaluates the performance of VEW with cameras that are less conventional but nevertheless useful in surveillance applications. The first camera is fitted with a

(a) Initial size at first tripwire. (b) Final size at second tripwire.

Figure 4: Multi-tripwire event tracked over rapid scale change in wide angle camera.

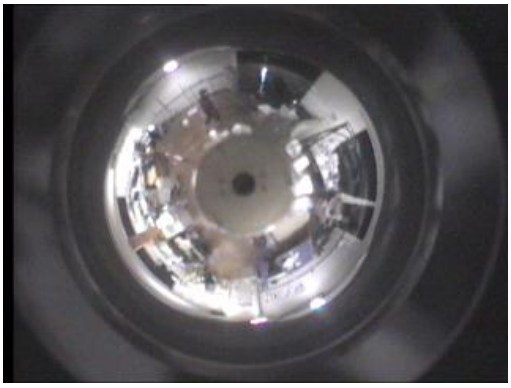
wide-angle lens (approximately 90 degree field of view), which allows a large area to be placed under surveillance as shown in Figure 4. Most wide-angle cameras, including this one, typically feature severe radial distortion and perspective foreshortening, and the main challenge is to track objects through rapid changes in shape and scale.

To test this issue, we constructed the multi-line tripwire rule shown in Figure 4. The appearance of the person at the first tripwire is shown in Figure 4(a), while the appearance at the second is shown in Figure 4(b). It should be noted that the physical distance separating these locations was about four metres, leading to a rapid change in scale with only a few steps. Despite this change, Figure 4(b) shows that the event was successfully detected, as validated by several repetitions. When the first tripwire was moved further down the road (to the left of the frame) to produce particularly severe foreshortening, the success rate was reduced to about half.

Next, VEW was tested with the omnidirectional camera shown in Figure 5(a), which features an upward pointing camera and curved mirror mounted in a clear plastic cylinder. The mirror was developed by Srinivasan at the Australian National University and is specially shaped so that radial distance from the centre of the captured frame is directly proportional to angle of elevation [17]. The camera was elevated on a tripod in the middle of a cluttered indoor area to give the view shown in Figure 5(b). For clarity, an “unwrapped” and mirror imaged version is shown in Figure 5(c), although this post-processing was not used by VEW. Much of the frame is occupied by the ceiling and the camera base, leaving only a narrow band of useful pixels. To operate successfully, VEW must be capable of classifying and tracking objects comprised of only a few tens of pixels. Furthermore, targets may undergo a change in orientation of up to 180 degrees while passing by the camera.

The rules shown in Figure 6 were constructed to test the performance of VEW in the presence of these difficulties. Figure 6(a) shows a multi-line tripwire alert image triggered by a person passing near the camera, and a similar multi-line tripwire rule was also constructed to detect vehicles. In Figure 6(b), a partial view rule detects the appearance of a person entering through a door, and again a similar rule was also constructed for vehicles. Over several trials, VEW successfully detected every person satisfying either rule, and more significantly the parallel alerts for vehicles were never

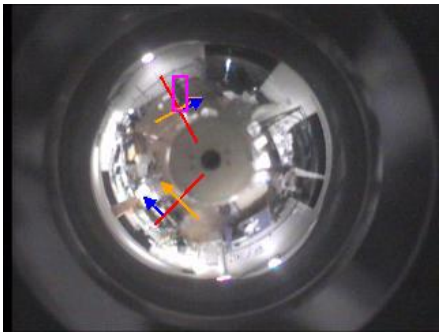
(a) Mirror and camera arrangement.



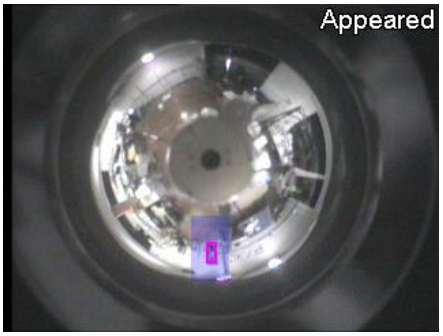
(b) Typical indoor view from elevated viewpoint.

(c) Unwrapped 360 degree view of lab from elevated viewpoint.

Figure 5: Omnidirectional camera.

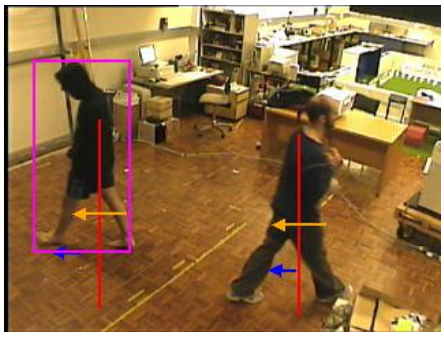


(a) Person crossing multi-line tripwire.

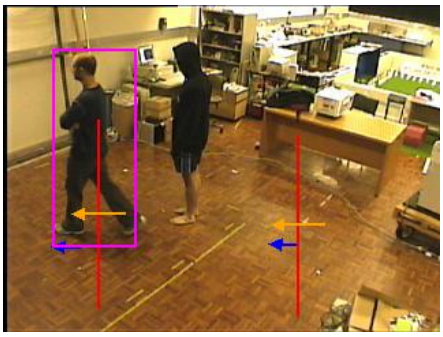


(b) Person appearing in partial view.

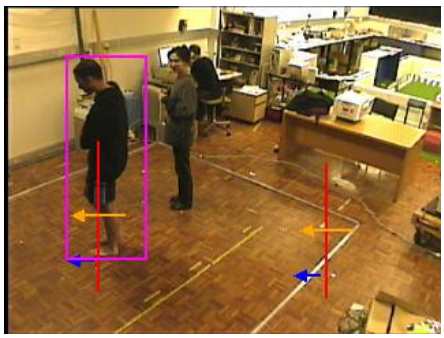
Figure 6: ObjectVideo alerts with omnidirectional camera.



(a) People walking in opposite directions.



(b) Multi-line tripwire triggered by two people.



(c) Multi-line tripwire triggered by two people.

(d) Crowded scene with multiple occlusions.

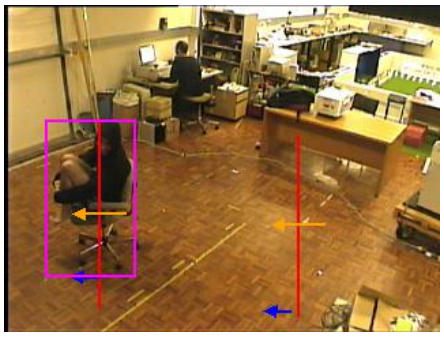
Figure 7: Alert images for occlusion tests with multi-line tripwires.

triggered. These encouraging results suggest that VEW is ready to handle the wide variety of cameras types that are useful in surveillance applications.

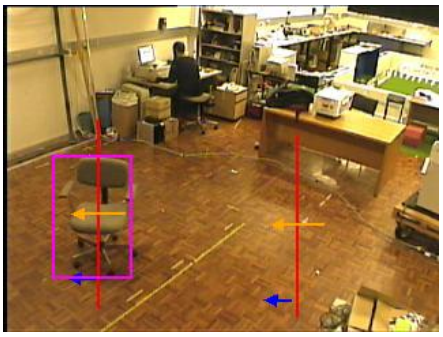
3.4 Crowded Room Tests

The previous sequences did not feature occlusions or distractions to hinder tracking. However, these difficulties can be particularly severe in many important surveillance scenarios including crowded airport lounges and hotel lobbies. To evaluate tracking performance in the presence of occlusions, we created a simple multi-line tripwire rule to detect people in the overhead view shown in Figure 7. The first test in Figure 7(a) shows two people crossing paths, with the background figure triggering the multi-line tripwire despite occlusion by the foreground figure. This event was repeated several times and the occluded person was successfully detected in each case.

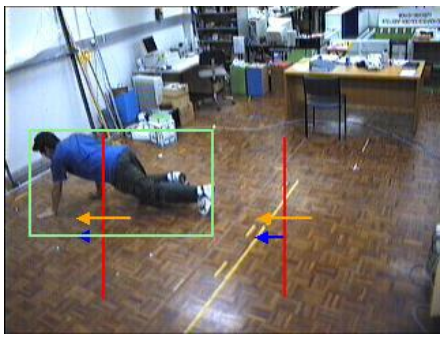
Figures 7(b) and 7(c) show a more difficult case, in which one person crosses the first tripwire and stops behind a second person, who is initially stationary between the tripwires. The second person then begins walking forward and crosses the second tripwire. In almost every case, VEW detects this as a single event despite the tripwires being triggered by different people. Such accidental switching of targets could be a



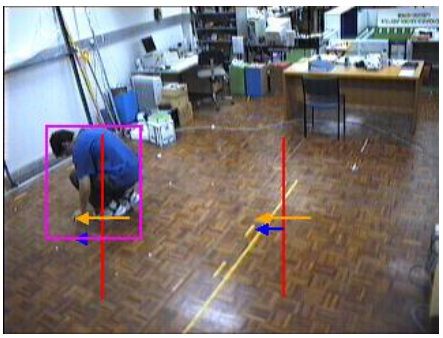
(a) Person on wheeled chair, detected as person.



(b) Empty wheeled chair, detected as person.



(c) Person crawling, detected as vehicle.



(d) Person crouching, detected as person.

Figure 8: Malicious deception by altering shape and motion.

shortcoming in some surveillance tasks.

Finally, Figure 7(d) shows the same rule applied to a crowded scene involving severe occlusions between several targets. While the alert was triggered, the target bounding box (purple rectangle) shows that the tracker did not distinguish between individual targets. Several trials of this scenario produced similar results. In such crowded situations, it can only be suggested that tracking may be improved by viewing the scene from a higher vantage point to minimize the severity of occlusions.

3.5 Malicious Deception Tests

In the previous tests, the subjects had no knowledge of being observed and made no attempt to deceive the system. However, deliberate attempts to avoid detection based on operational knowledge of the detection algorithms are possible in a realistic security scenario. The tests shown in Figures 8 and 9 were conceived to evaluate the performance of ObjectVideo VEW in this situation.

Figure 8 shows an overhead view with multi-line tripwires to detect both people and vehicles to highlight false classification results. In this test, VEW must correctly detect the person while the subject makes deliberate attempts to change his shape and

(a) Box on table swapped for identical box. (b) Object removed from partial view alert.

Figure 9: Detection of swapped object in partial view.

motion. The subject first crosses the scene in a wheeled chair, and Figure 8(a) shows that the correct event is detected despite the rigid motion. Interestingly, Figure 8(a) shows that the same classification is obtained for an empty wheeled chair. Figures 8(c) and 8(d) show attempts to disguise shape by crawling with the legs first extended and then retracted. In the first case, the subject is incorrectly classified as a vehicle (shown by the colour of the bounding box) despite the non-rigid motion. From these results, it would appear that target classification is mainly influenced by aspect-ratio. This leads to the possibility of malicious deception (as shown in Figure 8(c)) if the VEW rules are not constructed appropriately for the surveillance task.

For the final test shown in Figure 9, a rule was constructed to detect the removal of any object from a desktop. The item of interest is a cardboard box, which the subject attempts to swap for an identical box in his possession. Furthermore, the swap is made while obscuring the objects from view. Figure 9(a) shows the subject swapping the boxes, and the alert image in Figure 9(b) indicating the removal of the first box was generated by VEW only a short time later. The experiment was repeated several times with the same result in each case. From these results it can be concluded that VEW is sufficiently sensitive to scene changes to overcome this casual deception.

4 ObjectVideo VEW and Robotic Early Response

In the following sections, we investigate the possibility of triggering a robotic early response to surveillance alerts generated by VEW. Figure 10 illustrates the surveillance scenario that will serve as an example application of our system. The mobile robot operates in an indoor environment on a planar floor, under the observation of a static security camera. A VEW rule is created to detect any object left unattended in the accessible workspace (approximate dimensions shown). When an alert is received, the robot is automatically driven to the location of the unattended object using a technique known as *visual servoing*; that is, a differential wheel velocity is calculated purely from the relative location of the robot and target as viewed by the surveillance camera. The advantage of this approach is that the robot does not need any local sensing, odometric measurements or maps to be accurately driven to any location in the visible workspace.

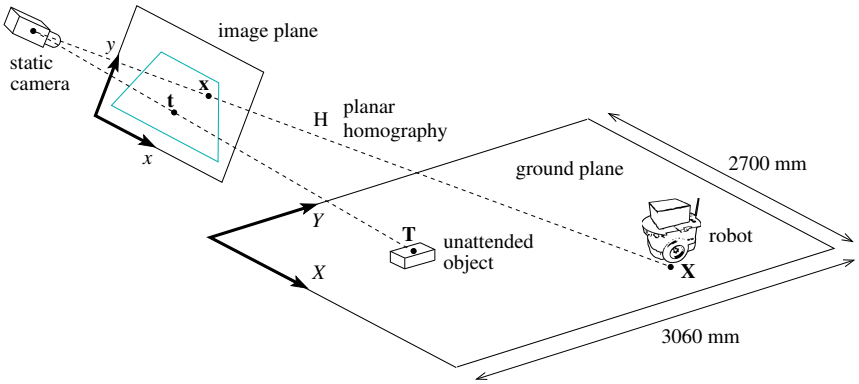


Figure 10: Application of robotic early response with ObjectVideo VEW.

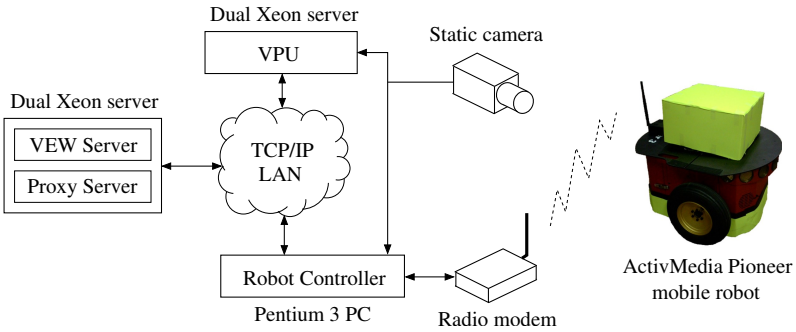


Figure 11: Architecture of experimental robotic surveillance system.

The architecture of the experimental robotic surveillance system is outlined in Figure 11. VEW is implemented on two Xeon workstations hosting the VPU and ObjectVideo Server respectively. Image processing and visual servo control are implemented on a 450 MHz Pentium 3 desktop PC, which is off-board the robot for simplicity. The robot controller communicates with an ActivMedia Pioneer 3 mobile robot via an RS-232 radio modem. A single static camera is used for both VEW surveillance and visual servo control by splitting and feeding the video signal into both the VPU and robot controller. The VPU, ObjectVideo Server and robot controller communicate via a TCP/IP network. One of the drawbacks of the VEW architecture is that any program receiving alerts directly from the ObjectVideo Server must communicate through the Windows-based Daemon service. However, the robot controller was written for Linux and thus required an alternative communication channel. To accommodate this requirement, a proxy server was created to receive alerts from the ObjectVideo Server on the Server PC, and then forwarded these across the network to the robot controller through a TCP socket.

The implementation of our visual servo robot controller is detailed below, followed in Section 4.4 with experimental results to demonstrate the feasibility of robotic early response in surveillance applications with ObjectVideo VEW.

4.1 Calibration

As shown in Figure 10, the purpose of our visual servo controller is to drive the robot from an arbitrary initial location to the location of the unattended object, based only on image plane measurements (ie. so that **x** and **t** coincide). To determine how the robot should be actuated to reduce the position error, the controller requires knowledge of the mapping between the image plane where the error is observed and ground plane on which the robot moves. As will be shown below, this mapping is a linear transformation that can be calculated using a simple image-based calibration procedure.

In *homogeneous* (or *projective*) coordinates, a camera can be modelled as a linear device. Let $\mathbf{X} = (X,Y,Z,1)^\top$ represent the homogeneous coordinates of a 3D point in real space, and $\mathbf{x} = (x,y,1)^\top$ represent the homogeneous coordinates of the corresponding 2D projection on the image plane (see Figure 10). Then, the operation of the camera can be expressed as $\lambda\mathbf{x} = \mathrm{P}\mathbf{X}$, where P is a 3×4 camera projection matrix and λ is an arbitrary scale. CCD cameras are typically approximated by the *central projection* model, which in the simplest case has a projection matrix of the form

$$\mathrm{P} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} (\mathrm{R}|\mathbf{T}) \tag{1}$$

where f is the focal length, and vector **T** and rotation matrix R describe the position and orientation of the camera in the world frame [10]. However, as will be shown below, the camera model need not be explicitly known for visual servoing.

Now, consider the arrangement of coordinate frames in Figure 10. Since the robot only moves on the XY-plane of the ground frame, the position of the robot can be written in homogeneous coordinates as $\mathbf{X} = (X,Y,0,1)^\top$. Then, the corresponding projection of the robot on the image plane is located at

$$\lambda\mathbf{x} = \mathrm{P}\mathbf{X} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} \tag{2}$$

Writing $\widetilde{\mathbf{X}} = (X,Y,1)^\top$, and eliminating the third column of P, equation (2) reduces to

$$\lambda\mathbf{x} = \begin{pmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \mathrm{H}\widetilde{\mathbf{X}} \tag{3}$$

where H is the 3×3 *planar homography* relating points on the ground and image planes. For simplicity, the location of the robot on the ground plane will henceforth be written as $\mathbf{X} = (X,Y,1)^\top$ (dropping the tilde).

The planar homography can be determined by manually measuring the location and intrinsic parameters of the camera and calculating the required elements of P from equation (1). However, a more robust and direct solution is to solve equation (3) directly for H from image plane measurements of known targets using standard linear least squares methods. The solution presented here follows the method outlined in [14] for finding the planar homography between stereo image planes. Due to the unknown factor λ, H is known only to an arbitrary scale and therefore has just 8 unconstrained degrees of freedom. Each image plane measurement provides two constraints (x and y),

and H is thus completely constrained by four or more general (non-collinear) points. To eliminate λ, both sides of equation (3) are first cross-multiplied by $\mathbf{x}$ to give

$$\mathbf{x} \times \mathrm{H}\mathbf{X} \equiv \widehat{\mathbf{x}}\mathrm{H}\mathbf{X} = 0 \tag{4}$$

where $\widehat{\mathbf{x}}$ is the skew symmetric matrix given by

$$\widehat{\mathbf{x}} = \begin{pmatrix} 0 & -1 & y \\ 1 & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

Equation (4) is known as the *planar homography constraint*. Writing the elements of H as the column vector

$$\mathbf{h} = (h_{11}, h_{21}, h_{31}, h_{12}, h_{22}, h_{32}, h_{13}, h_{23}, h_{33})^\top$$

and defining the 9×3 matrix A as the *Kronecker product* of $\mathbf{X}$ and $\widehat{\mathbf{x}}$, given by

$$\mathrm{A} = \mathbf{X} \otimes \widehat{\mathbf{x}} = \begin{pmatrix} 0 & X & -Xy & 0 & Y & -Yy & 0 & 1 & -y \\ -X & 0 & Xx & -Y & 0 & Yx & -1 & 0 & x \\ Xy & -Xx & 0 & Yy & -Yx & 0 & y & -x & 0 \end{pmatrix}^\top \tag{5}$$

we can rewrite the planar homography constraint as

$$\mathrm{A}^\top \mathbf{h} = 0$$

Now, let $\{\mathbf{X}_i, \mathbf{x}_i\}$, $i = 1 \ldots n$, represent a set of n points at known locations $\mathbf{X}_i$ on the ground plane, with corresponding measurements $\mathbf{x}_i$ on the image plane. For each corresponding pair, the Kronecker product A_i can be calculated using equation (5). Stacking $\mathrm{A}_i^\top$ into a single $3n \times 9$ matrix $\chi = (\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_n)^\top$, the planar homography constraint over all points can be written as

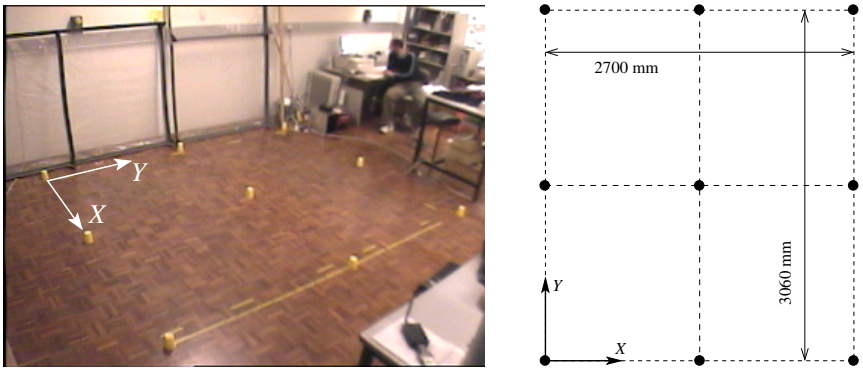
$$\chi \mathbf{h} = 0 \tag{6}$$

For $n > 4$, equation (6) is an over-constrained linear system (noting that A has rank 2) and can be solved for $\mathbf{h}$ using least squares methods. The solution that minimizes $\|\chi \mathbf{h}\|$ is the eigenvector of the 9×9 matrix $\chi^\top \chi$ corresponding to the smallest eigenvalue.

Figure 12(a) shows the calibration image used to calculate the experimental image/ground planar homography. Nine targets are placed in a regular grid at known locations in the ground frame, as shown in Figure 12(b). The targets locations are measured manually on both the ground and image planes, and the resulting set, $\{\mathbf{X}_i, \mathbf{x}_i\}$ $i = 1 \ldots 9$, is used to calculate first χ and finally the planar homography H using the least squares solution described above.

4.2 Robot Detection and Tracking

During visual servoing, the position of the robot is extracted from each captured frame using the series of image processing operations illustrated in Figure 13. The process is based on finding a foreground object with yellow markers matching those shown in Figure 11. Thus, the first step is to subtract a static background image pixel-wise from the newly captured frame. The difference is thresholded and morphologically eroded and dilated to reduce noise, and a binary connectivity analysis segments candidate foreground objects. A typical binary foreground image is shown in Figure 13(a).



(a) Experimental calibration image.

(b) Arrangement of targets.

Figure 12: Calibration of the image/ground planar homography.

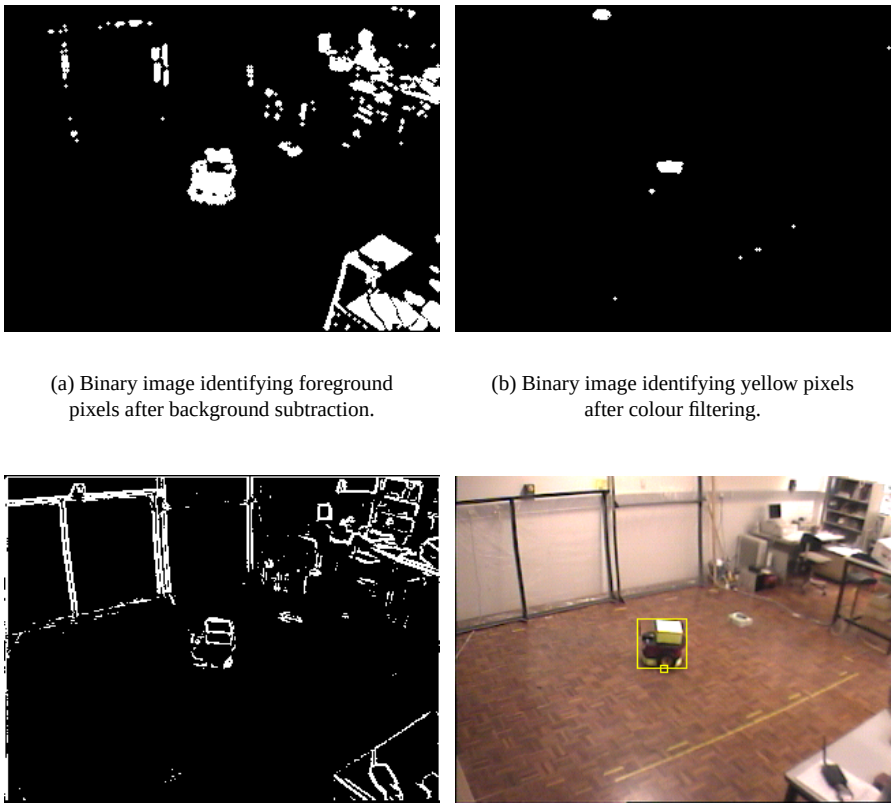
Next, the captured frame is passed through a hand-tuned yellow colour filter to identify pixels matching the colour of the robot's markers. A typical colour filtering result is shown in Figure 13(b). The colour filter and foreground images are combined to identify the robot by searching for the foreground region with the greatest number of yellow pixels. Let F_i represent the set of pixels in the ith fully connected foreground blob, and let C represent the set of yellow pixels in the entire image. Then, the foreground region F representing the robot is identified as:

$$F = \text{argmax}_{F_i} |F_i \cap C|$$

The selected region is validated by ensuring that sufficient pixels of the correct colour are found: $|F \cap C| > n_{th}$ for a fixed threshold n_{th}.

In practice, it was found that F typically included background pixels darkened by shadowing under the robot. A final image processing step was therefore designed to reduce this effect by observing that the shadows generally have significantly softer edges than the outline of the robot (this assumption would likely be violated outdoors). The sharp edges of the robot are identified by applying Sobel operators to the captured frame, as shown in Figure 13(c). Finally, the bounding box of the robot is calculated over the edge pixels lying within the foreground blob, given by the set $F \cap E$ where E is the set of all edge pixels. The bounding box of the robot in this example is overlaid in Figure 13(d). The centre of the lower edge of the bounding box (shown by the small yellow square in Figure 13(d)) serves as the measured position of the robot.

To drive the robot to a given location, both the position and direction of motion must be known. While position is directly observable as described above, orientation must be estimated by tracking the motion of the robot. The tracker is implemented in an extended Kalman filter (EKF) framework [1]. The extimated state of the robot is described by the state vector $\mathbf{x} = (\mathbf{X}_R, \theta_R, v_L, v_R)^\top$ where $\mathbf{X}_R = (X_R, Y_R)^\top$ and θ are the position and orientation (relative to the X-axis) of the robot on the ground plane, and v_L and v_R and linear velocities due to the left and right wheels. The measurement vector is $\mathbf{z} = (\mathbf{x}_R, \tilde{v}_L, \tilde{v}_R)^\top$, where $\mathbf{x}_R = (x_R, y_R)^\top$ is the image plane location of the robot, and $\tilde{v}_L$ and $\tilde{v}_R$ and the linear wheel velocities commanded by the visual servo controller (see



(a) Binary image identifying foreground pixels after background subtraction.

(b) Binary image identifying yellow pixels after colour filtering.

(c) Binary edge image for eliminating shadows.

(d) Bounding box of robot after modifying the lower bound using edge pixels.

Figure 13: Image processing steps for detection and tracking of the robot.

Section 4.3). While the wheel velocities are actually system inputs, they are treated as measurements due to uncertainty caused by inconsistent actuation lag.

Following the development in the previous section, the measurement model for the EKF is derived from the planar homography constraint in equation (3), which is projected into real space by normalizing the unknown scale λ. Thus, the non-linear measurement prediction equations are

$$x_R = (h_{11}X_R + h_{12}Y_R + h_{13})/(h_{31}X_R + h_{32}Y_R + h_{33}) \quad (7)$$

$$y_R = (h_{21}X_R + h_{22}Y_R + h_{23})/(h_{31}X_R + h_{32}Y_R + h_{33}) \quad (8)$$

where h_{ij} are elements of the planar homography matrix H.

The EKF also requires a dynamic model to predict the current state based on the estimated state from the previous measurement cycle. For simplicity we assume the wheel velocities remain constant, while the position and orientation are predicted using the motion model shown in Figure 14. The robot always moves along an instantaneous arc, with zero to infinite radius depending on the differential wheel velocity. Let R represent the arc radius, w represent the wheel base of the robot, $\Delta\mathbf{X}_R$ represent the

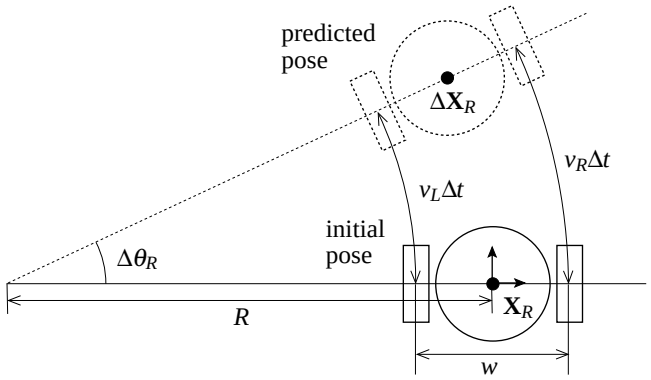


Figure 14: Motion model for state prediction.

change in position of the robot with respect to the initial pose, $\Delta\theta_R$ represent the change in orientation, and Δt represent the update period. Assuming constant velocity, the left and right wheels trace arcs of length $v_{L,R}\Delta t$ between updates. These arc lengths can also be expressed in terms of R, $\Delta\theta_R$ and w to give the simultaneous equations

$$\begin{aligned} v_L\Delta t &= \Delta\theta_R(R-w/2) \\ v_R\Delta t &= \Delta\theta_R(R+w/2) \end{aligned}$$

Solving for the unknown turning radius and change in orientation gives:

$$R = \frac{1}{2}w(v_R+v_L)/(v_R-v_L) \quad (9)$$

$$\Delta\theta_R = (v_R-v_L)\Delta t/w \quad (10)$$

The change in position $\Delta\mathbf{X}_R = (\Delta X_R, \Delta Y_R)^\top$ is then found by rotating the initial position (at the origin) by $\Delta\theta_R$ about the centre of the turning circle at $X = -R$:

$$\Delta X_R = R\cos\Delta\theta_R - R \quad (11)$$

$$\Delta Y_R = R\sin\Delta\theta_R \quad (12)$$

In the case of straight-line motion ($v_L = v_R$), equations (11)-(12) must be replaced with $\Delta X_R = X_R$ and $\Delta Y_R = Y_R + \frac{1}{2}(v_L+v_R)\Delta t$ to avoid the singularity in equation (9). Finally, transforming the change in pose from the initial robot frame to the world frame, the new position $\mathbf{X}'_R = (X'_R, Y'_R)^\top$ and orientation θ'_R are

$$X'_R = X_R + \Delta X_R\cos\Delta\theta_R - \Delta Y_R\sin\Delta\theta_R \quad (13)$$

$$Y'_R = X_R + \Delta X_R\sin\Delta\theta_R + \Delta Y_R\cos\Delta\theta_R \quad (14)$$

$$\theta'_R = \theta_R + \Delta\theta_R \quad (15)$$

The EKF is implemented using the measurement prediction model in equations (7)-(8) and the state prediction model in equations (13)-(15). Tracking is initialized with the state vector set to an arbitrary value (zero) with a large initial state error covariance. For each captured frame, the image plane position of robot (using the detection process above) and commanded wheel velocities form a new measurement vector, which is

processed by the standard EKF equations to update the estimated state and covariance. If the robot is not detected, the predicted state and covariance are used as the current estimate without applying the update equations. Due to the large initial covariance, the filter quickly converges on the real state of the robot as it moves in the field of view.

4.3 Visual Servo Control

As described earlier, the goal of this system is to drive the robot towards an unattended object under visual control. Let **t** represent the target location on the image plane as reported by ObjectVideo VEW. The corresponding location on the ground plane **T** is found by applying the inverse planar homography, $\mathbf{T} = \mathrm{H}^{-1}\mathbf{t}$. Using this measurement and the estimated pose of the robot, the visual servo controller drives the robot towards the target by modulating the differential wheel velocity to regulate the bearing error to zero. The bearing error is calculated by first finding the position of the target relative to the robot, $\mathbf{D} = \mathbf{T} - \mathrm{X}_R$. It should be noted that the estimated (instead of measured) position of the robot is used since the robot may not always be detected. Converting to polar coordinates, let θ_D represent the angle between **D** and the X-axis of the ground frame. Then, the bearing error is defined as $\phi = \theta_D - \theta_R - \pi/2$, where θ_R is the robot orientation estimated by the EKF, and the $\pi/2$ offset is necessary since the robot moves in the direction of the local Y-axis.

Let v_{max} represent the maximum linear velocity of the robot (the experimental implementation uses $v_{max} = 200$ mm/s). The differential wheel velocity is then modulated as follows: if the target is to the left of the robot (positive bearing error), the left wheel velocity is reduced to turn the robot anti-clockwise, and *vice versa* for a negative bearing error. Analytically, the control law is expressed as follows:

$$\tilde{v}_L = \begin{cases} v_{max} - 2v_{max}\phi/\pi & \text{for } \phi > 0 \\ v_{max} & \text{for } \phi \leq 0 \end{cases} \tag{16}$$

$$\tilde{v}_R = \begin{cases} v_{max} & \text{for } \phi \geq 0 \\ v_{max} + 2v_{max}\phi/\pi & \text{for } \phi < 0 \end{cases} \tag{17}$$

A drawback of this control approach is that a large bearing error may not converge to zero if **D** is sufficiently small, in which case the robot may simply orbit the target. To avoid this issue, the control task is considered to have converged when **D** is below a threshold (300 mm in the experimental implementation).

4.4 Results

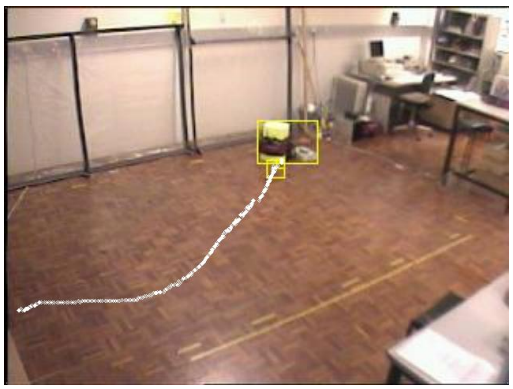
The principles and algorithms described above were used to experimentally implement the robotic early response extension to VEW, and the results are presented below[4]. As described earlier, a VEW rule was created to detect any object left behind on a planar workspace patrolled by the mobile Pioneer robot. Figures 15(a) and 15(b) show the placement of a suspicious object in the corner of the workspace, and the subsequent alert image generated by VEW. Along with the alert image, ObjectVideo provides "base64" encoded *Windows Metafile* data containing graphics primitives for displaying mark-up information including the bounding box of the target. The centroid of this bounding box provides the target location **t** for the robot.

Next, the robot was driven to the target by minimizing the estimated bearing error. Figure 15(c) shows the observed path of the robot (white trail) and the final position

[4] A video of these results is available online at http://www.irrc.monash.edu.au/gtaylor/AVS/index.html

(a) Placement of target.

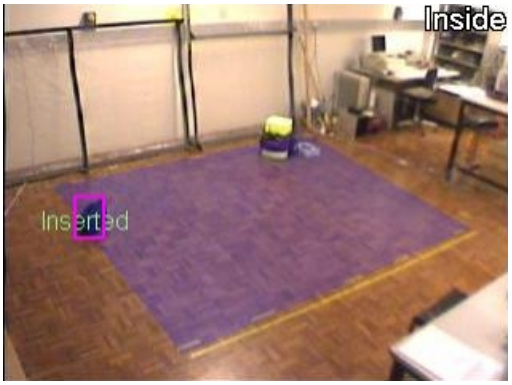
(b) ObjectVideo alert image. The region of interest is shaded in purple, and a yellow bounding box gives the location of the target.



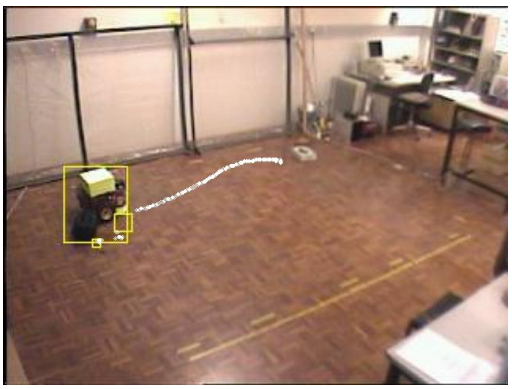
(c) Robotic response under visual servo control. The path of the robot is shown in white.

Figure 15: Placement, detection and robotic response for the first target.

(a) Placement of target.



(b) ObjectVideo alert image. The region of interest is shaded in purple, and a yellow bounding box gives the location of the target.

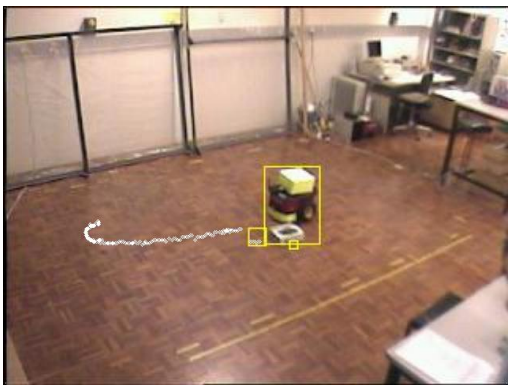


(c) Robotic response under visual servo control. The path of the robot is shown in white.

Figure 16: Placement, detection and robotic response for the second target.

(a) Placement of target.

(b) ObjectVideo alert image. The region of interest is shaded in purple, and a yellow bounding box gives the location of the target.



(c) Robotic response under visual servo control. The path of the robot is shown in white.

Figure 17: Placement, detection and robotic response for the third target.

Figure 18: Spurious ObjectVideo alert caused by stationary robot.

under visual control. The robot actually starts outside the field of view, and is therefore initially driven in an unknown direction based on an incorrect pose estimate. However, the path is quickly corrected and converges on the target when the robot fully enters the field of view. Once the first target was reached, a second object was placed in the workspace as shown in Figure 16(a). The corresponding alert image and response of the robot are shown in Figures 16(b) and 16(c). Finally, the first two targets were removed and a third object was detected as pursued as shown in Figure 17.

The three targets used in this experiment demonstrate significant coverage of the workspace by the robot. Furthermore, the system successfully handled bearing errors of up to 180 degrees, non-detection of the robot and other image processing difficulties such as overlap between the robot and target object. The main problem in the experimental implementation arose from false VEW alerts triggered by the motion of the robot. Figure 18 shows on such alarm generated when the robot halted after reaching the first target. To avoid responding to false alarms, the system was modified to require a human operator to validate each target (by responding to a simple dialogue window) before triggering the automatically controlled response.

5 Summary and Conclusions

In this research, we have qualitatively evaluated the performance of a commercial automatic video surveillance system, ObjectVideo VEW, and experimentally extended the concept to include automatic robotic early response. Using a rule-based approach to event detection, ObjectVideo VEW requires the user to specify image-based areas of interest (including tripwires and polygonal regions), events (crossing, loitering, entering, *etc*) and object classes (people and vehicles) to generate alert triggers. This intuitive approach was generally found to facilitate a short learning curve and efficient deployment of the fully functional installation.

Through a series of test video sequences, ObjectVideo VEW was found to detect most specified events in a variety of conditions both indoors and outdoors with several different cameras. Test sequences using wide-angle and omnidirectional cameras demonstrated good tracking performance despite variations in the scale and orientation of targets. More difficult sequences demonstrated that occlusions could cause the tracker to jump between different targets or merge several targets together and generate

false alarms. While revealing a weakness of VEW, this result highlights the importance of viewpoint planning in any AVS installation.

The final series of sequences involved attempts to deliberately deceive VEW using basic knowledge of the underlying algorithms. In the first test, VEW successfully detected a person crossing a multi-line tripwire despite squatting or using a wheeled chair to disguise the shape and motion. However, crawling with legs extended was found to fool VEW into classifying the person as a vehicle, leaving the system vulnerable to deception in a poorly conceived installation. In the final test, VEW successfully detected several attempts to exchange two identical objects.

Robotic early response takes the AVS concept further by enabling a roaming mobile robot to quickly and autonomously respond to a security event while additional human support is organized. In this work, a robotic early response system was implemented using VEW to detect unattended items on the floor and a Pioneer robot to move to the object under visual control from a single surveillance camera. Visual control allows the system to employ robots without sophisticated internal sensing or mapping capabilities by exploiting the existing sensor network of surveillance cameras. Experimental trials demonstrate the feasibility of the approach, directing the robot to detected targets throughout the area under surveillance.

The image processing algorithms used for this implementation allow the system to work in a static indoor environment with controlled lighting and low clutter. Clearly, more sophisticated techniques must be developed for visual robot control to robustly handle real-world issues such as lighting variations and occlusions. Furthermore, robotic early response is only useful if the robot can perform subsequent tasks such as erecting perimeters, identifying threats and removing or securing offending items. These capabilities are the subject of ongoing research.

Acknowledgment

This work was funded by the *Australian Research Council Centre for Perceptive and Intelligent Machines in Complex Environments*. Special thanks to Ray Jarvis, Lindsay Kleeman, Gideon Kowadlo, David Rawlinson, David Fernandez, Wai Ho Li and Anies Purnamadjaja for use of equipment and floor space, and participating in VEW trials.

References

[1] Y. Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association*. Academic Press, San Deigo, 1988.

[2] A. Birk and H. Kenn. Roboguard, a teleoperated mobile security robot. *Control Engineering Practice*, 10(11):1259–1264, 2002.

[3] M. Bogaert, N. Chelq, P. Cornez, C. S. Regazzoni, A. Teschioni, and M. Thonnat. The PASSWORDS project. In *Proc. International Conference on Image Processing*, volume 3, pages 675–678, 1996.

[4] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, 2000.

[5] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color and pattern detection. *Int. Journal of Computer Vision*, 37(2):175–185, 2000.

[6] A. Dick and M. J. Brooks. Issues in automated visual surveillance. In *International Conference on Digital Image Computing: Techniques and Applications*, 2003.

[7] C. P. Diehl and J. B. Hampshire II. Real-time object classification and novelty detection for collaborative video surveillance. In *Proc. 2002 International Joint Conference on Neural Networks*, volume 3, pages 2620–2625, 2002.

[8] G. L. Foresti. Object recognition and tracking for remote video surveillance. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(7), October 1999.

[9] D. Gibbins, G. Newsam, and M.J. Brooks. Detecting suspicious background changes in video surveillance of busy scenes. In *Third IEEE Workshop on Applications of Computer Vision Sarasota*, pages 22–26, 1996.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[11] D. Koller. Moving object recognition and classification based on recursive shape parameter estimation. In *Proc. 12th Israel Conference on Artificial Intelligence, Computer Vision and Neural Networks*, pages 359–368, 1993.

[12] A. J. Lipton, H. Fujiyoshi, and R. S. Patil. Moving target classification and tracking from real-time video. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 8–14, October 1998.

[13] R. C. Luo and K. L. Su. A multiagent multisensor based real-time sensory control system for intelligent security robot. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 2394–2399, 2003.

[14] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An invitation to 3-D vision: from images to geometric models*, volume 26 of *Interdiciplinary Applied Mathematics*. Springer-Verlag, New York, 2004.

[15] P. E. Rybski, N. P. Papanikolopoulos, S. A. Stoeter, D. G. Krantz, K. B. Yesin, M. Gini, R. Voyles, D. F. Hougen, B. Nelson, and M. D. Erickson. Enlisting rangers and scouts for reconnaissance and surveillance. *IEEE Robotics Automation Magazine*, 7(4):14–24, December 2000.

[16] N. L. Seed and A. D. Haughton. Background updating for real-time image processing at tv rates. *SPIE Image Processing, Analysis, Measurement and Quality*, 901, 1988.

[17] M.V. Srinivasan. A new class of mirrors for wide angle imaging. In *Proc. IEEE Workshop on Omnidirectional Vision and Camera Networks*, Madison, Wisconsin, USA, June 2003.

[18] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, June 1999.