

A Scalable Image-Based Multi-Camera Visual Surveillance System

Ser-Nam Lim

Larry S. Davis

Ahmed Elgammal

University of Maryland, College Park
Computer Vision Laboratory, UMIACS
{sernam,lsd,Elgammal}@umiacs.umd.edu

Abstract

In this paper, we aim to achieve scalability and wider scene coverage through the use of multiple cameras in an outdoor visual surveillance system. Only image-based information is used to control the cameras, making the system highly scalable. We show that when a Pan-Tilt-Zoom camera pans and tilts, a given image point moves in circular and linear trajectory respectively. We also create a scene model using a directly top-view image of the scene to eliminate perspective foreshortening. The scene model allows us to handle occlusion occurring as a result of line of sight between cameras and tracked objects and to determine the view of an object that a camera is acquiring images of. In addition, we also use a maximum weight matching algorithm to assign cameras to tasks when the number of cameras are limited. Finally, the system is tested by simulations and a prototype system.

1 Introduction

Most prior research on video surveillance, including our own, has focused on low and intermediate level vision problems associated with detection, classification and tracking of moving objects from single, fixed surveillance platforms. When multiple camera surveillance systems are considered, it is typically either to increase the reliability of tracking algorithms (e.g., to overcome effects of occlusion, or to utilize 3D information for tracking), or to track an object in hand-off situations as it moves out of the Field of View (FOV) of one camera and into the FOV of another (and the emphasis here has been principally on model transference from one (camera, illumination, view direction), to another). In this paper, we present an image-based camera control technique that handle handoff situations.

Here, we also consider problems related to more general tasking of multiple camera visual surveillance systems, and to the control and scheduling of those cameras to satisfy as

many of those tasks as possible (or to optimize some other figure of merit with respect to tasks completed). We envision an environment in which multiple cameras have to monitor a site so large that it cannot be continuously and completely monitored at a resolution sufficient to conduct surveillance anywhere at any time. Instead, objects (people, vehicles, etc.) are detected and tracked using time-varying and dynamic subsets of the cameras at a wide field of view, and based on surveillance task specifications, predictive models of object position and sensor availability, cameras are scheduled in space-time-resolution to acquire images and short video segments that vision algorithms (e.g., biometric algorithms) can use to satisfy tasks.

We desire an approach that is general with respect to the set of physical camera assets - so that a camera can be added to the system with minimal effort, and the system can continue to operate if one or more cameras are removed (due to failure or maintenance, for example). We accomplish this by scheduling "logical sensors" [8], and then mapping such logical sensors onto available physical cameras.

We consider three general classes of surveillance tasks; their collective scheduling using available cameras employing one unified scheduling mechanism constitutes our surveillance system. All tasks are defined in an scene model, which includes the ground plane, vehicular and human pathways on the ground plane, Region of Interest (ROI) such as vertical walls of building with entrance/exit locations identified, and possibly some important fixed objects such as receptacles, etc. The three types of tasks are:

1. Initial moving object acquisition: These tasks are defined with respect to a set of fixed and dynamic locales (e.g., a building door would be a fixed locale, while a car or human that enters a parking lot would be a dynamic locale).
2. Position updating of moving objects: These tasks are defined with respect to predicted positions of moving targets (which generally degrade with time). Targets are reacquired and briefly tracked to update their posi-

tion and velocity estimates. Currently, simple Kalman filters are used as predictive models.

- Acquisition of surveillance images and videos for monitoring activities. Based on our predictive models of object motion, we create a dynamic model of object visibility from virtual cameras in space-time. This is used to schedule collections of physical cameras to satisfy surveillance tasks. Examples of such tasks might include collecting different views (facial, front, back, left, right) of an object at a given spatial resolution or better which involve predicting when that person would be unobstructed and sufficiently close to some camera to allow us to acquire the image at an appropriate zoom setting. The system must be able to determine the views acquired by a given camera. The system should also favor cluster of objects that are close to one another since we expect a higher probability of object interaction to occur in such situations and it allows a single camera resource to acquire images of multiple objects at the same time. Again, path prediction and occlusion modeling can be used to construct virtual sensors to acquire this video; we then attempt to map these virtual sensors to physical cameras to satisfy the task.

The task scheduling problem can be posed as a general temporal logic-programming problem, where we populate a "database" of current camera schedules and imaging parameters (e.g. camera position, field of regard, range of focal lengths, latencies associated with Pan/Tilt/Zoom (PTZ) of each camera), along with positional predictions together with the associated uncertainties and visibility predictions of moving objects from the site model. Tasks of all three types can then be represented as theorems to prove in an appropriate temporal logic. The proofs then correspond to possible assignments of cameras to individual tasks (in space-time-resolution), which are then subsequently analyzed by a general scheduling mechanism with respect to a scheduling figure of merit.

2 System Overview

Figure 1 illustrates the overall architecture of the proposed system. The detection technique used is an adaptive model of the scene background to detect moving object using kernel density estimation [5, 12]. Tracking is accomplished using approach in [7, 4]. All cameras are able to detect and track and the camera that picks up an object in a handoff situation will be used to provide tracking information for assigning and scheduling other cameras to acquire images of that object. Occlusion arising from objects being closed together (as opposed to that occurring from line of sight between cameras and tracked objects) is also handled

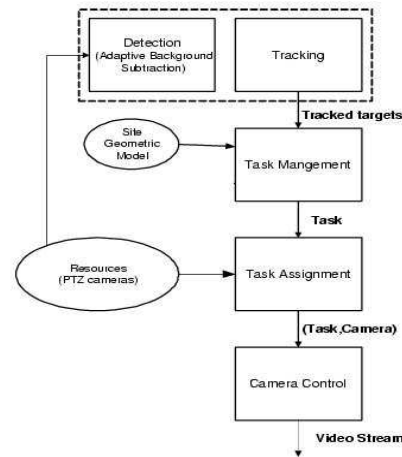


Figure 1. System architecture

in the detection and tracking. Readers should refer to the references for details.

The system is first required to position the camera (camera control) effectively and accurately, even when handoff situations arise. Views acquired are required to be unobstructed, which leads to the requirement for a scene model (task management) that allows us to resolve any occlusion situations that might arise between tracked objects and static background objects by providing positional information about objects in the scene.

Moreover, we are faced with a scheduling problem (task assignment) whenever we have more tracked objects than the number of cameras. Considerations for task scheduling include occlusion, high priority regions (ROI), high priority objects (e.g. objects exiting scene soon) etc. Uncertainties present in such information make task assignment heuristic and probabilistic in nature.

Our system uses a set of PTZ cameras mounted in nadir orientations with respect to the scene. Section 3 describes the camera handoff technique for controlling and facilitating camera positioning in handoff situations, Section 4 describes the role a scene model plays in our system and how one is created, Section 5 describes how tasks are assigned, when the number of cameras are limited, using a graph representation with a scheduling figure of merit for each edge and Section 6 will discuss the contributions and other issues presented in this paper. Results of the techniques will be shown where applicable. Relevant references will also be given throughout the paper where necessary. All techniques described are image-based and non-metric i.e. no three dimensional information was used in our analysis.

3 Image-based Camera Control

Previous work on camera control include [10] that setup a ground-plane coordinate system by watching the objects entering and leaving the scene and then recovering the image-plane to local-ground-plane transformation of each camera, [9] which used the color model of objects to integrate information from multiple cameras and others like [1, 3] which used the more traditional approach of calibrating the cameras by providing different approaches to determine the intrinsic and extrinsic parameters. In this paper, we propose an image-based technique that uses the trajectories of image points to determine how to position a camera. This approach essentially uses a frame in the reference camera (which pickups the object) to communicate with other cameras by mapping the image coordinates of objects from the reference camera to the assigned camera.

Each camera is assigned a pan-tilt zero-position, ϕ_0 and ψ_0 [11]. 3×3 homographies with 8 d.o.f. between these zero-positions are pre-computed, without assuming affine, by solving a linear system using four pairs of image landmarks where (x_i, y_i) denotes the landmarks in one image and (x'_i, y'_i) denotes the corresponding landmarks in another image

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \times \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix}. \quad (1)$$

We assume objects are moving on the ground plane so that the homographies are sufficiently accurate. This provides only the calibration between different cameras in their zero-positions. Different pan-tilt positions of the same camera are related to each other by projective rotations. [2, 11] provides details about projective rotations, which is defined to be the homography that corresponds to pure Euclidean rotation. We show that the projective rotations conjugated to pan and tilt are approximately circular and linear trajectories in the image plane respectively. As a result, cameras can be positioned accurately by first mapping the image coordinates of a detected object in the reference camera's zero-position to the assigned camera's zero-position using the pre-computed homographies in Equation 1. The mapped image coordinates is in turn mapped to the current position of the assigned camera by using the projective rotations relating different pan-tilt positions of the same camera. Cameras are then controlled by panning first follow by tilting, based on projective rotations.

3.1 Relation Between Image Trajectory and Projective Rotations

Figure 2 shows a top view of the camera geometry. Let the change in pan angle be ψ_c starting from any initial pan

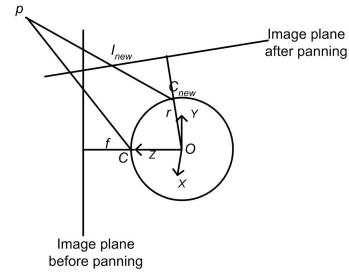


Figure 2. Top view - camera geometry

position and the tilt angle be ϕ , where the tilt angle is defined as the angle between the rotation axis and the ray joining the pan-tilt joint and the camera's center of projection.

Given a point p with world coordinate $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$, we start with the camera at $\psi_c = 0$ and some tilt ϕ . We have the world coordinate system axes orientated as in Figure 2 and r and f as the radius of rotation and focal length respectively. At the current tilt ϕ , the center of projection is

$$C = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = \begin{bmatrix} 0 \\ -r \cos \phi \\ r \sin \phi \end{bmatrix} \quad (2)$$

If we change the pan angle by panning with respect to the world coordinate system's y-axis, we get a rotation matrix R , for a right-handed system that gives C_{new}

$$R = \begin{bmatrix} \cos \psi_c & 0 & -\sin \psi_c & 0 \\ 0 & 1 & 0 & 0 \\ \sin \psi_c & 0 & \cos \psi_c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$C_{new} = \begin{bmatrix} -r \sin \phi \sin \psi_c \\ -r \cos \phi \\ r \sin \phi \cos \psi_c \end{bmatrix} \quad (4)$$

The coordinate of p in the camera coordinate system thus changed by first applying translation opposite to the initial world coordinate of C and then R in the reverse direction

$$\begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} = \begin{bmatrix} X \cos \psi_c + Z \sin \psi_c \\ Y + r \cos \phi \\ -X \sin \psi_c + Z \cos \psi_c - r \sin \phi \\ 1 \end{bmatrix} \quad (5)$$

Let the camera-to-image mapping M_{CI} be

$$M_{CI} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

where (u_0, v_0) is the principle point, $\alpha_u = fk_u, \alpha_v = -fk_v, ku =$ horizontal inter-pixel distance and $k_v =$ vertical inter-pixel distance. We map Equation 5 from 3D to 2D

$$\begin{bmatrix} x_{cam} \\ y_{cam} \\ f \end{bmatrix} = \begin{bmatrix} f \frac{X \cos \psi_c + Z \sin \psi_c}{-X \sin \psi_c + Z \cos \psi_c - r \sin \phi} \\ f \frac{Y + r \cos \phi}{-X \sin \psi_c + Z \cos \psi_c - r \sin \phi} \\ f \end{bmatrix} \quad (7)$$

followed by applying Equation 6 to get the image coordinate of p

$$I_{new} = \begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} \alpha_u \frac{X \cos \psi_c + Z \sin \psi_c}{-X \sin \psi_c + Z \cos \psi_c - r \sin \phi} + u_0 \\ \alpha_v \frac{Y + r \cos \phi}{-X \sin \psi_c + Z \cos \psi_c - r \sin \phi} + v_0 \end{bmatrix} \quad (8)$$

3.2 Projective Rotations Conjugated to Pure Camera Pan

Equation 8 can be used to derive the relation between I_x and I_y

$$I_y = \frac{\alpha_v(Y + r \cos \phi)}{\alpha_u \sqrt{X^2 + Z^2} \cos(\psi_c - \tan^{-1} \frac{Z}{X})} (I_x - u_0) + v_0 \quad (9)$$

Since we are considering pure camera pan, ϕ is a constant. Hence as the pan angle changes, I_y and I_x change with respect to each other in the form of a $\frac{1}{\cos}$ curve. Note that if the world point is behind the image plane, I_x and I_y becomes undefined, so we are only concerned with the part of the curve that corresponds to trajectory while the world point is in front of the image plane. To model this characteristic of the projective rotation for pure camera pan, we propose using a circular trajectory that best fits the part of the curve when the world point is in front of the image plane. This is because the term v_0 in Equation 9 would flatten out the curve largely since we would expect $|v_0|$ to be comparatively large, making it close to a circular curve. Figure 3 shows the partial plot of the trajectory of an image point while it is in front of the image plane. Note that the image coordinate system used here has x-value increasing leftward due to the projective matrix.

Different image points have different centers of trajectory. However, we know that both r and f is very small compared to the coordinates of p in Figure 2 i.e. the change in distance between two image points as the camera pans is very small. It also means that the angle between two different image points and the center of circle for one of the image points will change very slightly when the camera pans. In Figure 4, this means that $d \approx d'$ and $\theta \approx \theta'$ even as the camera is panned, where p_1 and p_2 are the image points of two different points in the scene, p'_1 and p'_2 are the corresponding image points after the camera panned, r is now

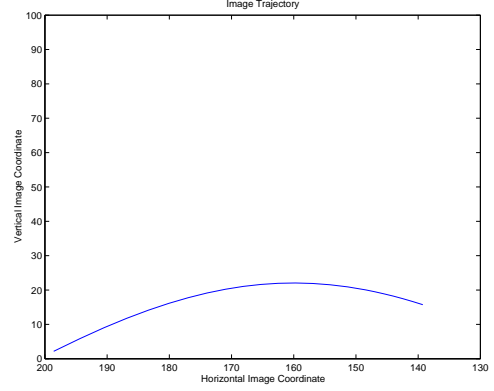


Figure 3. Image trajectory

the radius of trajectory for p_1 and p'_1 and C is the center of trajectory for p_1 and p'_1 . Therefore, $R \approx R'$ i.e. different trajectories for panning have the same circle center.

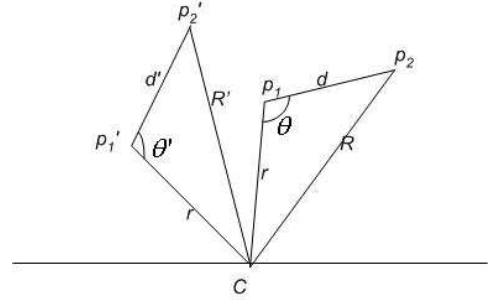


Figure 4. Trajectories for panning have approximately same circle center

3.3 Projective Rotations Conjugated to Pure Camera Tilt

If the pan angle is fixed while we change the tilt angle, it is clear that the horizontal image coordinate remains constant if r is small and therefore $r \sin \phi \ll -X \sin \psi_c + Z \cos \psi_c$. Using Equation 8, we have

$$I_x \approx \alpha_u \frac{X \cos \psi_c + Z \sin \psi_c}{-X \sin \psi_c + Z \cos \psi_c} + u_0 \quad (10)$$

Similarly, for the vertical image coordinate, the term $-r \sin \phi$ in the denominator can be eliminated. However, we cannot do so in the numerator since $\alpha_v r \cos \phi$ cannot be ignored because we expect α_v to be large

$$I_y \approx \alpha_v \frac{Y + r \cos \phi}{-X \sin \psi_c + Z \cos \psi_c} + v_0 \quad (11)$$

Equation 11 thus tells us that given a constant-sized tilt step δ , the change in I_y remains constant. This characteristic is useful for controlling the tilt position of a camera.

3.4 Using the Camera Control Technique

To find the center of circular trajectory for panning, we collect n image coordinates of the same point as the camera is panned, where $n \geq 3$ and then run the algorithm in [13] to get the center of the circular trajectory that these points lie on. Each camera is also fitted with a zoom function that gives the magnification factor. This zoom function is used to compute the amount of zooming in required in order to acquire images of certain resolution and is specific to each camera.

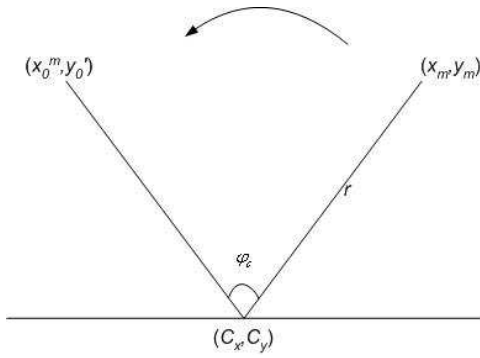


Figure 5. Moving camera between different pan-tilt positions

The camera control technique starts by computing homographies between zero-positions of the cameras as in Equation 1. The center of image trajectories, (C_x, C_y) , for each camera (all of which could potentially pickup an object in a handoff situation) is first computed. With all cameras starting in their respective zero-positions, when a camera first detects an object and predicts its future image coordinate to be (x_m, y_m) , given its pan as ψ_m and tilt as ϕ_m , it will first have to convert (x_m, y_m) to (x_0^m, y_0^m) , the corresponding image coordinate in its zero-position. This can be achieved by first panning with respect to (C_x, C_y) . Since we know the pan angle for the zero-position, ψ_c is known. In addition, we know the radius r of trajectory is the Euclidean distance between (C_x, C_y) and (x_m, y_m) . Referring to Figure 5, we have

$$r = \sqrt{(C_x - x_m)^2 + (C_y - y_m)^2} \quad (12)$$

$$x_0^m = r \cos(180 - \psi_c - \cos^{-1} \frac{|C_x - x_m|}{r}) + C_x \quad (13)$$

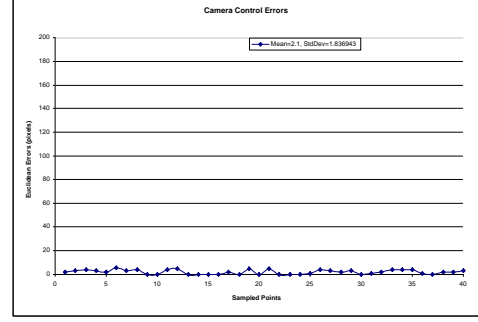


Figure 6. Error in camera control method is small

$$y_0' = C_y - r \sin(180 - \psi_c - \cos^{-1} \frac{|C_y - y_m|}{r}) \quad (14)$$

where y_0' is not yet the vertical image coordinate for the zero-position, but since we know the change in I_y is constant with a constant-sized tilt step from Section 3.3, the change in tilt angle to get to the zero-position can be used to compute y_0^m . If the assigned camera is i , then the image coordinate (x_0^i, y_0^i) in the zero-position of camera i is

$$I_i = H \begin{bmatrix} x_0^m \\ y_0^m \\ 1 \end{bmatrix} \quad (15)$$

where $I_i = \begin{bmatrix} x_0^i \\ y_0^i \\ 1 \end{bmatrix}$ and H is the homography between zero-positions of the reference camera and camera i . Using Equation 12 to Equation 14 again but on I_i should give us the image coordinate in the current position of camera i . To move camera i so that the detected object is in a vantage position in camera i 's image plane, such as the center of the image, we can let X_0^m be the image center. We can then derive y_0' in Equation 14 and compute the change in tilt so that the vertical image coordinate is in the center of the image.

To show the effectiveness of the camera control technique, we randomly sample a number of different image points in some reference camera and move the assigned camera so that the image points would be in some desired position. Euclidean differences between the resulting and desired positions are then recorded. We used three PTZ cameras, one of which is designated as the reference camera. They are mounted at high positions and have common field of view. Figure 6 shows that the error of our camera control method is very small, where the maximum error is less than 6 pixels.

Two sets of images showing the camera control technique in action is shown in Figure 7. In each set, the refer-

ence camera detects the object and assigns another camera to acquire images of the object. The assigned camera uses the camera control technique to compute the position of the object and zoomed in on the object.



Figure 7. (a) Reference camera detects (b) Assigned camera positions (c) Assigned camera zooms

4 Role of a Scene Model

The scene model contains positional information about the cameras, static background objects, ROI and tracked objects in the scene. In this paper, we create a single image of the scene from a direct top viewpoint that is fronto-parallel. We will refer to this as the blueprint. Homographies between the cameras' zero-positions and the blueprint are pre-computed using Equation 1. A fronto-parallel image is necessary to prevent any perspective foreshortening. All further analysis is performed by first mapping the position of the object in the camera's image to the blueprint using Equation 1. Figure 8 shows a manually configured scene model used in the experimentations.

Section 1 mentioned one task is to acquire images of an object from different viewing directions for uses in gait analysis. Using the scene model, we can determine the direction from which an object is viewed by a given camera using the direction of movement relative to the camera. A Kalman-based path predictor [7] is used to predict the future position of an object, (x_f, y_f) . Since we know the cur-

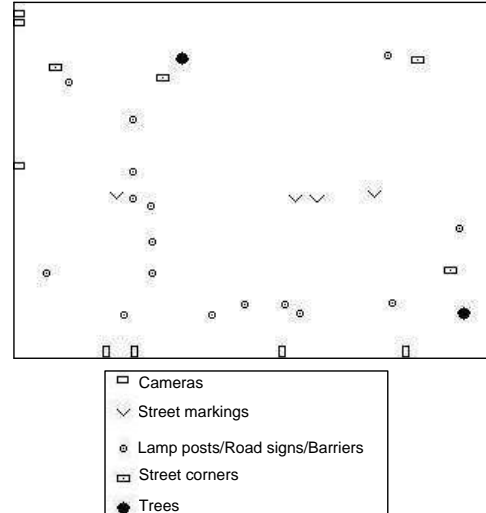


Figure 8. Scene model

rent position (x_c, y_c) of the object, the directed vector from $y_f - y_c$ to $x_f - x_c$ is the direction of motion. If we assume that objects are moving facing the direction of motion, we can infer the view taken by a given camera. Figure 9 shows the system in real time action and determining views correctly, while Figure 10 shows simulations using the same system but objects are inserted synthetically at random positions. The current positions of the objects are denoted by the white bounding boxes while red, blue, yellow and green boxes represent that the predicted positions are front, back, left and right views respectively as determined by the system. The corresponding scene models on the right show the predicted direction of object movement with black dots representing the assigned cameras and other color dots representing the predicted positions which are colored accordingly to the determined views.

We also use positional information from the scene model to detect occlusions that result from the line of sight between the cameras and static background objects and different tracked objects in the scene. Let the predicted position of object i as (x_i, y_i) and object j as (x_j, y_j) and the position of camera assigned to acquire images of object i be (C_x, C_y) . Possible occlusion situations between the two objects with respect to the camera might occur if both object i and j are in the same viewing direction with respect to the camera and $|\tan^{-1}(\frac{y_i - C_y}{x_i - C_x}) - \tan^{-1}(\frac{y_j - C_y}{x_j - C_x})| < \epsilon$ where ϵ is some tolerance level that depends on the average width of the objects (e.g. humans) we are tracking. To prevent tracked objects that are assimilated into the background after remaining stationary for long period, we do not add dynamic background objects to the background model but instead keep their status as being tracked to retain their

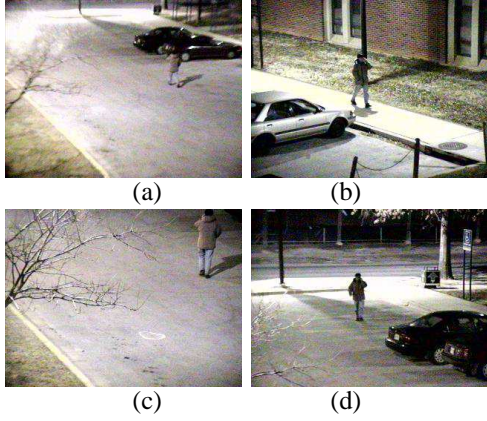


Figure 9. Views were determined accurately (a) Back-left view (b) Front-right view (c) Back view (d) Front view

positional information in the scene model.

We use a random number generator to synthetically create different number of objects denoted by blue boxes in Figure 11 in the scene at random positions. Objects occluded in the same line of sight with the assigned camera and other objects are bounded in red bounding boxes. Figure 11 shows randomly created objects that are occluded bounded in red boxes with the corresponding scenarios in the scene model. Black dots in the scene model represent cameras, blue dots represent unobstructed objects and red dots represent occluded objects.

5 Task Scheduling

Task scheduling in a visual surveillance system deals with the issue of limited resources i.e. cameras are assigned in a way that optimizes the throughput of the system. Factors affecting the system throughput include higher priorities for objects in ROI and exiting objects. The task scheduling problem can be abstracted into the logical sensor architecture/specifications as described in [8]. We also need to consider the camera's associated latency including time for positioning and zooming. Uncertainties involved must also be considered, such as the accuracy of the predicted position.

Let some initial time instance be t_0 . In $t_0 + \delta_t$, we wanted to be able to schedule cameras to acquire images of objects in the scene requiring that they are unobstructed, the view has not already been acquired and $t_0 + \delta_t > t_0 + t_m + t_z + t_r$ where t_m is time taken by a camera to position, t_z is the time taken by a camera to zoom in based on its zoom function and t_r is the time taken for running the scheduling. Let such requirements that must be met and has value of

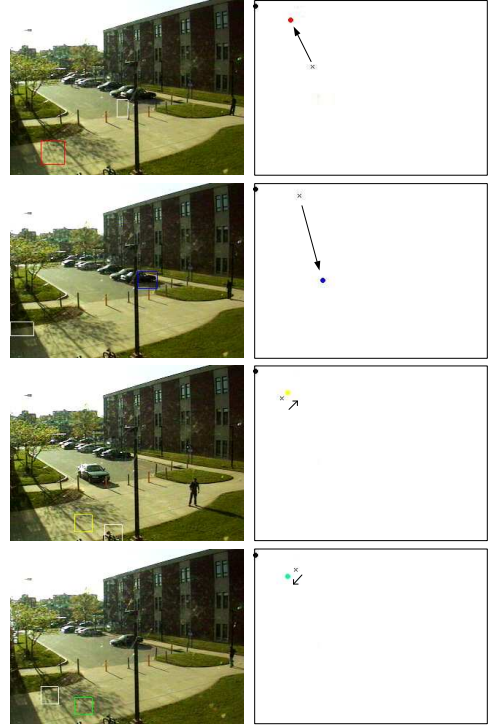


Figure 10. Role of scene model in determining views

0 or 1 (requirement is either met or not) be represented by R_i . We can also have other factors that are not required but will improve the throughput of the system if met. Let these factors be r_j and the associated weights be w_j . As mentioned, such factors include higher priorities for objects in ROI and objects exiting the scene and objects proximity indicating possible interactions and that camera can acquire images of multiple objects at the same time. In addition, uncertainties in predicting object position in the future have to be handled as well.

In the general case, we can derive for each camera, the scheduling figure of merit B for each of the tracked objects

$$B = \prod_i R_i \times \sum_j w_j r_j \quad (16)$$

Equation 16 is general enough to handle issue such as uncertainties of predicted position, where we expect $P(k) \propto \frac{1}{t_k}$ where t_k is the length of time since the path prediction of object k was last updated. Hence, $r_k = \frac{1}{t_k}$ will be included as one of r_j while evaluating Equation 16. Matching cameras with tasks using B is performed using a maximum weight matching technique from Gabow's N-cubed weighted matching [6]. A bipartite graph, one set being the tasks and the other set the cameras is set up. We label each

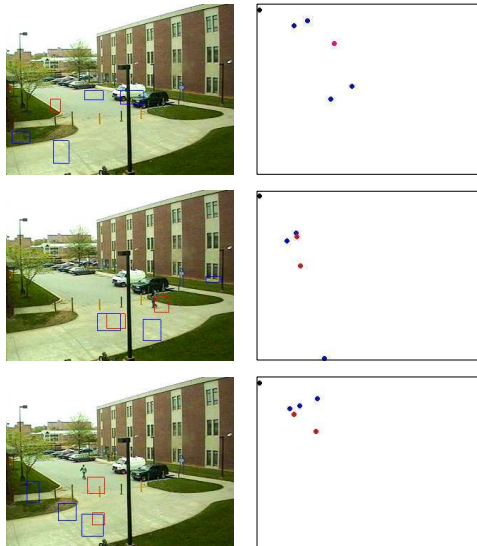


Figure 11. Role of the scene model in occlusion handling

edge in the bipartite graph with the associated B and then perform Gabow's algorithm to generate the maximum (and hence optimum) possible weights among all the edges in the set of camera-task pairs.

6 Discussions

In this paper, our contributions include the proposal of an image-based camera control technique and scene modeling, that are the two main components of a scalable surveillance system described in this paper. As the camera control technique uses only two dimensional information, it is highly scalable. The scene model is also used solely for significantly simplifying occlusion handling and determination of views acquired of an object by a camera. However, a direct top view of the scene is sometimes not readily available. Although possible sources can come from aerial photos, satellite images, their resolutions are usually not sufficiently high enough for establishing accurate homographies. In this paper, we have manually configured a scene model with some given scale factor. This approach while effective is not very scalable. Further studies on this issue are necessary. We have also shown how tasks can be scheduled using a logical sensor abstraction. We provide a way to compute a scheduling figure of merit that is general enough to include different requirements such as uncertainties associated with predicted positions and higher priorities in certain cases. A bipartite graph maximum weight matching approach is then used for generating the best set of camera-task pairs in terms

of the scheduling figure of merit. Effectiveness and reliability of the system have been tested rigorously both in a real-time prototype system and simulations.

References

- [1] P. N. abd R.J. Valkenburg. Calibrating a pan-tilt camera head. In *Image and Vision Computing Workshop, New Zealand*, 1995.
- [2] P. A. B. Andrew Zisserman and I. D. Reid. Metric calibration of a stereo rig. In *IEEE Workshop on Representation of Visual Scenes, Cambridge, Massachusetts*, Jun 1995.
- [3] R. T. Collins and Y. Tsin. Calibration of an outdoor active camera system. In *IEEE Computer Vision and Pattern Recognition, Fort Collins, Colorado*, Jun 1999.
- [4] A. Elgammal and L. S. Davis. Probabilistic framework for segmenting people under occlusion. In *Proc. of IEEE 8th International Conference on Computer Vision*, 2001.
- [5] A. Elgammal, D. Harwood, and L. S. Davis. Nonparametric background model for background subtraction. In *Proc. of 6th European Conference of Computer Vision*, 2000.
- [6] H. Gabow. Implementation of algorithms for maximum matching on nonbipartite graphs. *Ph.D. thesis, Stanford university, California*, 1973.
- [7] I. Haritaoglu, D. Harwood, and L. S. Davis. W4:who? when? where? what? a real time system for detecting and tracking people. In *International Conference on Face and Gesture Recognition*, 1998.
- [8] T. Henderson and E. Shilcrat. Logical sensor systems. *Journal of Robotic System*, 1(2):169–193, 1984.
- [9] P. R. J. Orwell and G. Jones. Multi-camera color tracking. In *Second IEEE Workshop on Visual Surveillance, Fort Collins, Colorado*, Jun 1999.
- [10] G. A. Jones, J.-P. Renno, and P. Remagnino. Auto-calibration in multiple-camera surveillance environments. In *Proceedings of IEEE International Workshop on Performance Evaluation and Surveillance*, 2002.
- [11] A. Ruf and R. Horaud. Projective rotations applied to a pan-tilt stereo head. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.*, pages 144–150, 1999.
- [12] D. W. Scott. *Multivariate Density Estimation*. Wiley-Interscience, 1992.
- [13] G. Taubin. Estimation of planar curves, surfaces and non-planar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991.