Surveillance Camera Scheduling: A Virtual Vision Approach

Faisal Z. Qureshi[1] and Demetri Terzopoulos[2,1]

[1]Department of Computer Science, University of Toronto, Toronto ON M5S 3G4, Canada
[2]Courant Institute of Mathematical Sciences, New York University, New York, NY 10003, USA

ABSTRACT

We present a surveillance system, comprising wide field-of-view (FOV) passive cameras and pan/tilt/zoom (PTZ) active cameras, which automatically captures and labels high-resolution videos of pedestrians as they move through a designated area. A wide-FOV stationary camera can track multiple pedestrians, while any PTZ active camera can capture high-quality videos of a single pedestrian at a time. We propose a multi-camera control strategy that combines information gathered by the wide-FOV cameras with weighted round-robin scheduling to guide the available PTZ cameras, such that each pedestrian is viewed by at least one active camera during their stay in the designated area.

A distinctive centerpiece of our work is the exploitation of a visually and behaviorally realistic virtual environment simulator for the development and testing of surveillance systems. Our research would be more or less infeasible in the real world given the impediments to deploying and experimenting with an appropriately complex camera sensor network in a large public space the size of, say, a train station. In particular, we demonstrate our surveillance system in a virtual train station environment populated by autonomous, lifelike virtual pedestrians, wherein easily reconfigurable virtual cameras generate synthetic video feeds that emulate those generated by real surveillance cameras monitoring richly populated public spaces.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Motion,Tracking*; I.5.4 [**Pattern Recognition**]: Applications—*Computer Vision*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Scheduling*

General Terms

Design

Keywords

Surveillance Systems, Virtual Vision, Camera Scheduling, Camera Control, Sensor Coordination

VSSN'05, November 11, 2005, Singapore.
Copyright 2005 ACM 1-59593-242-9/05/0011 ...$5.00.

1. INTRODUCTION

We regard the design of an autonomous visual sensor network as a problem in resource allocation and scheduling, where the sensors are treated as resources required to complete the required sensing tasks. Imagine a situation where the camera network is asked to capture high-resolution videos of every pedestrian that passes through a region of interest.[1] Passive cameras alone cannot satisfy this requirement. Active *pan/tilt/zoom* (PTZ) cameras are needed to capture high-quality videos of pedestrians. Often there will be more pedestrians in the scene than the number of available cameras, so the PTZ cameras must intelligently allocate their time among the different pedestrians, and a resource management strategy can enable the cameras to decide autonomously how best to allocate their time to viewing the various pedestrians in the scene. The dynamic nature of the sensing task further complicates the decision making process; e.g., the amount of time a pedestrians spends in the designated area can vary dramatically between different pedestrians, an attempted video recording by a PTZ camera might fail due to occlusion, etc.

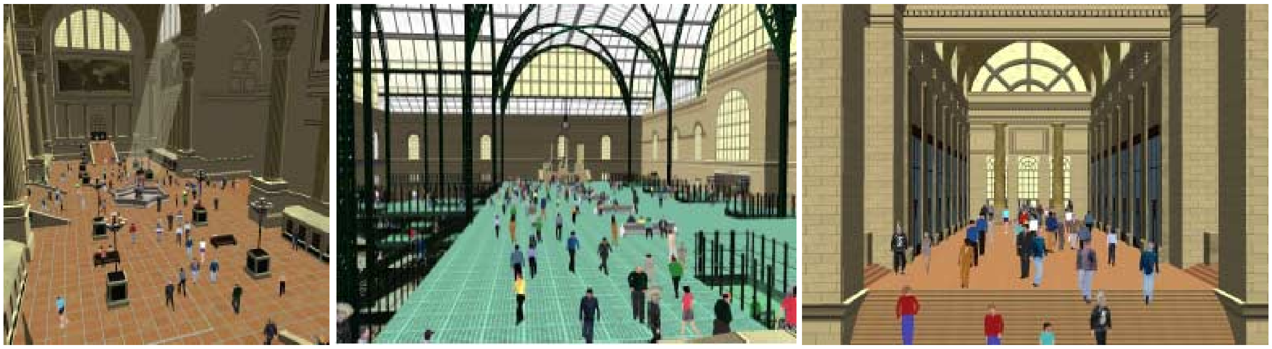
1.1 The Virtual Vision Paradigm

Even if there were no legal obstacles to monitoring people in public spaces for experimental purposes, the cost of deploying a large-scale camera network in the real world and experimenting with it can easily be prohibitive for computer vision researchers. As was argued in [1], however, computer graphics and virtual reality technologies are rapidly presenting viable alternatives to the real world for developing computer vision systems. Legal impediments and cost considerations aside, the use of a virtual environment can also offer greater flexibility during the system design and evaluation process. Terzopoulos [2] proposed a *Virtual Vision* approach to designing surveillance systems using a virtual train station environment populated by fully autonomous, lifelike virtual pedestrians that perform various activities (Figure 1) [3]. Within this environment, virtual cameras generate synthetic video feeds (Figure 2). The video streams emulate those generated by real surveillance cameras, and low-level image processing mimics the performance characteristics of a state-of-the-art surveillance video system. The virtual vision approach to surveillance in sensor networks was developed further in our recent work [4].

1.2 The Virtual Sensor Network

Within the virtual vision paradigm, we propose a sensor network consisting of wide field-of-view (FOV) stationary cameras and PTZ cameras to capture automatically and label high-quality video for every pedestrian that passes through a designated region. The net-

[1] The captured video can subsequently be used for further biometric analysis, e.g., by a facial, gesture, or gait recognition routine.



Waiting room

Concourses and platforms

Arcade

Figure 1: A large-scale virtual train station populated by self-animating virtual humans.

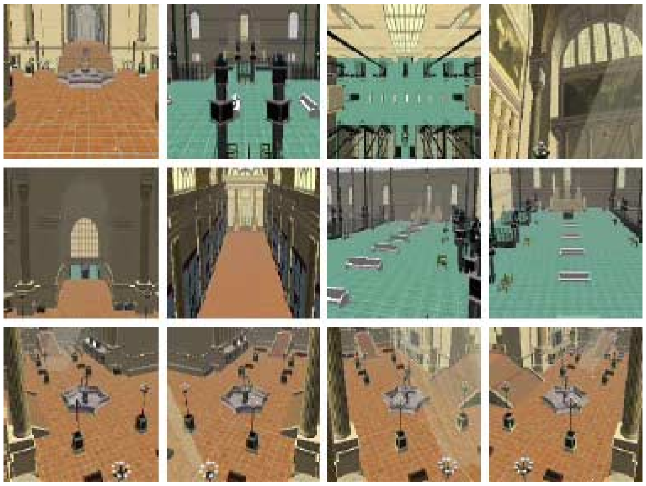


Figure 2: Virtual Vision. Synthetic video feeds from multiple virtual surveillance cameras situated in the (empty) Penn Station environment.

work described here is a special instance of the sensor network architecture proposed in [4]. The network is capable of performing common visual surveillance tasks through local decision making at each node, as well as internode communication, without relying on camera calibration, a detailed world model, or a central controller.

Unlike [4], we assume in our current work that the wide-FOV stationary cameras are calibrated,[2] which enables the network to estimate the 3D locations of the pedestrians through triangulation. However, we do not require the PTZ cameras to be calibrated. Rather, during a learning phase, the PTZ cameras learn a coarse mapping between the 3D locations and the gaze-direction by observing a single pedestrian in the scene. A precise mapping is unnecessary since we model each PTZ camera as an autonomous agent that can invoke a search behavior to find the pedestrian using only coarse hints about the pedestrian's position in 3D. The network uses a weighted round-robin strategy to assign PTZ cameras to the various pedestrians. Each pedestrian creates a new sensing request in the task queue. Initially, each sensing request is assigned the same priority; however, the decision making process uses domain-specific heuristics, such as the distance of the pedestrian from a camera or the heading of the pedestrian, to evaluate continuously the priorities of the sensing requests. The PTZ cameras handle each task in priority sequence. A warning is issued when a sensing request cannot be met.

1.3 The Virtual World Simulator

Our visual sensor network is deployed and tested within the virtual train station simulator that was developed in [3]. The simulator incorporates a large-scale environmental model (of the original Pennsylvania Station in New York City) with a sophisticated pedestrian animation system that combines behavioral, perceptual, and cognitive human simulation algorithms. The simulator can efficiently synthesize well over 1000 self-animating pedestrians performing a rich variety of activities in the large-scale indoor urban environment. Like real humans, the synthetic pedestrians are fully autonomous. They perceive the virtual environment around them, analyze environmental situations, make decisions and behave naturally within the train station. They can enter the station, avoiding collisions when proceeding though portals and congested areas, queue in lines as necessary, purchase train tickets at the ticket booths in the main waiting room, sit on benches when they are tired, purchase food/drinks from vending machines when they are hungry/thirsty, etc., and eventually proceed downstairs in the concourse area to the train tracks. Standard computer graphics techniques enable a photorealistic rendering of the busy urban scene with considerable geometric and photometric detail (Figure 1).

1.4 Contributions and Overview

The contributions of the research reported herein are as follows: First, we demonstrate the advantages of implementing, experimenting with, and evaluating our sensor network system within the virtual vision paradigm. Furthermore, we develop new gaze-direction controllers for active PTZ cameras. Next, we propose a sensor management scheme that appears well suited to the challenges of designing camera networks for surveillance applications capable of fully automatic operation. Finally, we also demonstrate how our system can be used for semantic labeling (or thematic grouping) of the recorded video.

The remainder of the paper is organized as follows: Section 2 covers relevant prior work. We explain the low-level vision emulation in Section 3. In Section 4, we describe PTZ active camera controllers and propose a scheme for learning the mapping between 3D locations and gaze directions. Section 5 introduces our scheduling strategy. We present our initial results in Section 6 and our conclusions and future research directions in Section 7.

[2] This assumption is justifiable given the success of numerous automatic static camera calibration schemes [5, 6].

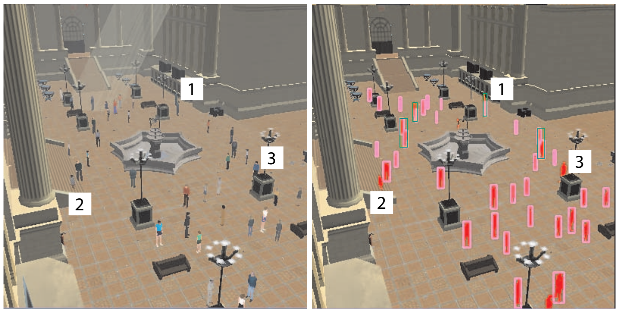


Figure 3: Pedestrian segmentation and tracking. (1) Multiple pedestrians are grouped together due to poor segmentation. (2) Noisy pedestrian segmentation results in a tracking failure. (3) Pedestrian segmentation and tracking failure due to occlusion.

2. RELATED WORK

Previous work on multi-camera systems has dealt with issues related to low and medium-level computer vision, namely identification, recognition, and tracking of moving objects [7, 8, 9, 10, 11]. The emphasis has been on tracking and on model transference from one camera to another, which is required for object identification across multiple cameras [12]. Numerous researchers have proposed camera network calibration to achieve robust object identification and classification from multiple viewpoints, and automatic camera network calibration strategies have been proposed for both stationary and actively controlled camera nodes [5, 6].

Little attention has been paid, however, to the problem of controlling or scheduling active cameras when there are more objects to be monitored in the scene than there are active cameras. Some researchers employ a stationary wide-FOV camera to control an active tilt-zoom camera [13, 14]. The cameras are assumed to be calibrated and the total coverage of the cameras is restricted to the FOV of the stationary camera. Zhou *et al.* [14] track a single person using an active camera. When multiple people are present in the scene, the person who is closest to the last tracked person is chosen. The work of Hampapur *et al.* [15] is perhaps closest to ours in that it deals with the issues of deciding how cameras should be assigned to various people present in the scene. Costello *et al.* [16] evaluates various strategies for scheduling a single active camera to acquire biometric imagery of the people present in the scene.

The problem of online scheduling has been studied extensively in the context of scheduling jobs on a multitasking computer [17, 18] as well as for packet routing in networks [19, 20].

3. LOCAL VISION ROUTINES

As we described in [4], each camera has its own suite of visual routines for pedestrian recognition, identification, and tracking, to which we refer as Local Vision Routines (LVRs). The LVRs are computer vision algorithms that directly operate upon the synthetic video generated by virtual cameras and the information readily available from the 3D virtual world. The virtual world affords us the benefit of fine tuning the performance of the recognition and tracking modules by taking into consideration the readily available ground truth. Our imaging model emulates camera jitter and imperfect color response; however, it does not yet account for such imaging artifacts as depth-of-field and image vignetting. More sophisticated rendering schemes would address this limitation.

We employ appearance-based models to track pedestrians. Pedestrians are segmented to construct unique and robust color-based pedestrian signatures, which are then matched across the subsequent frames. Pedestrian segmentation is carried out using 3D geometric information as well as background modeling and subtraction. The quality of the segmentation depends upon the amount of noise introduced into the process, and the noise is drawn from Gaussian distributions with appropriate means and variances. Color-based signatures, in particular, have found widespread use in tracking applications [21]. Unfortunately, color-based signatures are sensitive to illumination changes; however, this shortcoming can be mitigated by operating in HSV space instead of RGB space.

The tracking module mimics the performance of a state-of-the-art tracking system (Figure 3). For example, it can lose track due to occlusions, poor segmentation, or bad lighting. Tracking sometimes locks onto the wrong pedestrian, especially if the scene contains multiple pedestrians with similar visual appearance; i.e., wearing similar clothes. Tracking also fails in group settings when the pedestrian cannot be segmented properly.

For the purposes of this paper, we assume that the scene is viewed by more than one calibrated wide-FOV passive camera plus at least one PTZ active camera. Multiple calibrated static cameras allow the system to use triangulation to compute the location of a pedestrian in 3D, when the pedestrian is simultaneously visible in two or more cameras. For PTZ cameras, zooming can drastically change the appearance of a pedestrian, thereby confounding conventional appearance-based schemes, such as color histogram signatures. We tackle this problem by maintaining HSV color histograms for several camera zoom settings for each pedestrian. Thus, a distinctive characteristic of our pedestrian tracking routine is its ability to operate over a range of camera zoom settings.

4. PTZ ACTIVE CAMERA CONTROLLER

We implement each PTZ active camera as a behavior-based autonomous agent [4]. The overall behavior of the camera is determined by the LVR and the current task. The camera controller is modeled as an augmented finite state machine. At the highest level, the camera can be in one of the following states: *free*, *tracking*, *searching*, and *lost* (Figure 4). When a camera is free, it selects the next sensing request in the task pipeline. The sensing requests are of the form, "look at the pedestrian i at location (x, y, z) for t seconds." When performing the new sensing request, the camera selects its widest FOV setting and chooses an appropriate gaze direction using the estimated 3D location of the pedestrian. Upon the successful identification of the pedestrian in question within the FOV, the camera uses image-driven fixation and zooming algorithms to follow the subject.

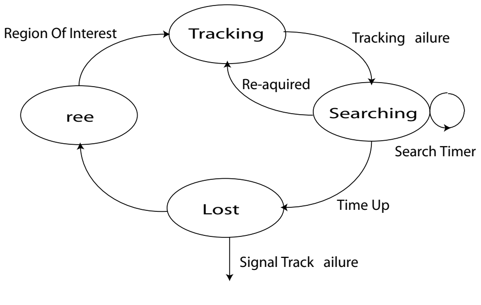


Figure 4: Top-level camera controller.

Each camera can fixate on and zoom in on an object of interest. Fixation and zooming routines are image driven and do not require any 3D information, such as camera calibration or a global frame

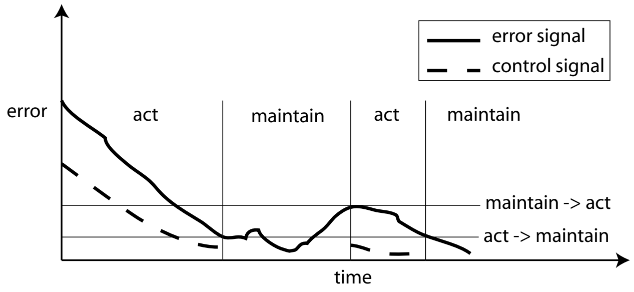


Figure 5: Dual-state controller for fixation and zooming.

of reference. We discovered that traditional proportional derivative (PD) controllers generate unsteady control signals resulting in jittery camera motion. The noisy nature of tracking forces the PD controller to strive to minimize the error metric continually without ever succeeding, so the camera keeps servoing. Hence, we model the fixation and zooming routines as dual-state controllers. The states are used to activate/deactivate the PD controllers. In the *act* state the PD controller tries to minimize the error signal; whereas, in the *maintain* state the PD controller ignores the error signal altogether and does nothing (Figure 5).

The fixate routine brings the region of interest—e.g., the bounding box of a pedestrian—into the center of the image by tilting the camera about its local X and Y axes (Figure 6, Row 1). The zoom routine controls the FOV of the camera such that the region of interest occupies the desired percentage of the image. This is useful in situations where, for example, the operator desires a closer look at a suspicious pedestrian (Figure 6, Row 2).

4.1 Gaze Direction Computation

Computing an appropriate gaze direction in order to bring the desired pedestrian within the FOV of a camera requires a mapping between the 3D locations in the world and the internal gaze-direction parameters (i.e., the pan-tilt settings) of the camera. This mapping is established automatically by tracking and following a single pedestrian in the scene during an initial learning phase.

During learning, a pedestrian is directed to move around in the scene. The pedestrian is tracked by the calibrated stationary cameras and 3D location of the pedestrian is estimated continuously through triangulation. The PTZ cameras are instructed to track and follow the pedestrian and a lookup table is computed for each PTZ camera, which associates the 3D (x, y, z) location of the pedestrian with the corresponding internal pan-tilt settings of the camera. This yields n tuples of the form (x, y, z, α, β), where α and β are the camera pan and tilt angles.

Subsequent to the learning phase, given any new 3D point $\vec{p}$, the system can estimate the values for α and β of any camera that can observe the point by using the nearest neighbor approximation. This process provides only a coarse mapping between the 3D points and the camera pan-tilt settings; however, in practice the mapping is accurate enough to bring the pedestrian within the field of view of the camera.

The distance of $\vec{p}$ from the nearest (x, y, z) in the lookup table is a good indicator of the accuracy of the computed angles. If this distance is large, the PTZ camera invokes a search behavior to locate the pedestrian. In order to minimize the reliance on the initial learning phase, the lookup table is continuously updated when the PTZ camera is following a pedestrian whose 3D location is known.

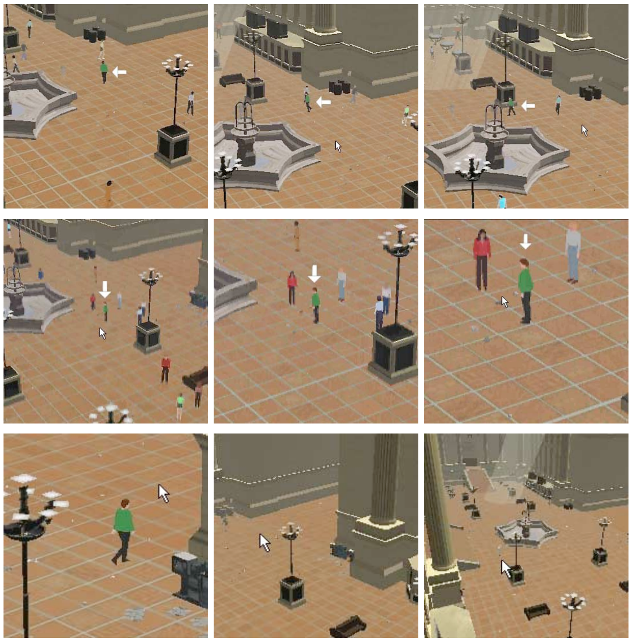


Figure 6: Row 1: A fixate sequence. Row 2: A zoom sequence. Row 3: Camera returns to its default settings upon losing the pedestrian; it is now ready for another task.

5. CAMERA SCHEDULING

The sensor network maintains an internal world model that reflects the current state of the world. The internal world model stores information about the pedestrians present in the scene, including their arrival times and the most current estimates of their positions and headings. The world model is available to the scheduling routine which assigns cameras to the various pedestrians present in the scene. The cameras use the 3D information stored in the world model to choose an appropriate gaze direction when viewing a particular pedestrian.

Following the reasoning presented in [16], the camera scheduling problem shares many characteristics with the network packet routing problem. Network packet routing is an online scheduling problem where the arrival times of the packets are not known *a priori* and where each packet must be served for a finite duration before a deadline, when it is dropped by the router. Similarly, in our case, the arrival times of pedestrians entering the scene is not known beforehand and a pedestrian must be observed for some minimal amount of time by one of the PTZ cameras before (s)he leaves the scene. That time serves as the deadline.

However, the problem addressed here differs from the packet routing problem in several significant ways. First, continuing with network terminology, we have multiple routers (one for every PTZ camera) instead of just one. This aspect of our problem is better modeled using scheduling policies for assigning jobs to different processors. Second, we typically must deal with additional sources of uncertainty: 1) it is difficult to estimate when a pedestrian might leave the scene and 2) the amount of time for which a PTZ camera should track and follow a pedestrian to record high-quality video that is suitable for further biometric analysis can vary depending upon multiple factors, e.g., a pedestrian suddenly turning away from the camera, a tracking failure, an occlusion, etc.

The scheduling algorithm must find a compromise between two competing ends: 1) to capture high-quality video for as many as possible, preferably all, of pedestrians in the scene and 2) to view each pedestrian for as long or as many times as possible. The second goal is supported by the observation that the chances of identifying a pedestrian are directly proportional to the amount of data collected for that pedestrian. At one extreme, the camera can follow a pedestrian for their entire stay in the scene, essentially ignoring all other pedestrians, whereas, at the other extreme, the camera would repeatedly observe every pedestrian in turn for a single video frame, thus spending most of the time transitioning between different pan, tilt, and zoom settings.

We propose a weighted round-robin scheduling scheme with a static *First Come, First Serve* (FCFS+) priority policy that strikes a balance between these two goals. The weighted round-robin scheduling scheme is a variant of the round-robin scheduling scheme used for assigning jobs to multiple processors with different load capacities. Each processor is assigned a weight indicating its processing capacity and more jobs are assigned to the processors with higher weights. We model each PTZ camera as a processor whose weights are adjusted dynamically. The weights quantify the suitability of a camera with respect to viewing a pedestrian. They are determined by two factors: 1) the amount of adjustments the camera needs to make in the PTZ coordinates to look at the pedestrian and 2) the distance separating the pedestrian from the camera.

A camera that requires small adjustments in the PTZ coordinates to look in the direction of a pedestrian usually needs less *lead* time (the total time required by a PTZ camera to locate and fixate on a pedestrian and initiate the video recording) than a camera that needs to turn more drastically in order to bring the pedestrian into view. Consequently, we assign a higher weight to a camera that needs the least amount of redirection to observe the pedestrian in question. On the other hand, a camera that is closer to a pedestrian is more suitable for observing this pedestrian, as such an arrangement can potentially avoid occlusions, tracking loss, and subsequent re-initialization, by reducing the chance of another pedestrian coming in-between the camera and the subject being recorded.

A danger of using weighted round-robin scheduling is that there is a possibility that a majority of the jobs will be assigned to the processor with the highest weight. We avoid this situation by sorting the PTZ cameras according to their weights with respect to a given pedestrian and assigning the free PTZ camera with the highest weight to that pedestrian. The FCFS+ policy breaks ties by selecting the pedestrian who entered the scene first. The arrival times of the pedestrians are maintained by the network and are made available to the PTZ cameras. We did not choose other possible tie breaking options—e.g., *Earliest Deadline First* (EDF+)—since they require an estimate of the exit times of the pedestrians from the scene, which are difficult to predict.

The amount of time a PTZ camera spends viewing a pedestrian depends upon the number of pedestrians in the scene; however, we have specified a minimum length of time that a PTZ camera must spend looking at a pedestrian. This is determined by the minimum length of the video sequence required by the biometric routines that perform further evaluation plus the average time it takes a PTZ camera to lock onto and zoom in on a pedestrian.

6. RESULTS

We populated the train station with up to twenty autonomous pedestrians, entering, wandering, and leaving the waiting room of their own volition. We tested our scheduling strategy in various scenarios using anywhere from 1 to 18 PTZ active cameras. For example, Figure 7 shows our prototype surveillance system consisting of five wide-FOV stationary cameras situated within the waiting room of the virtual train station. The system behaved as expected

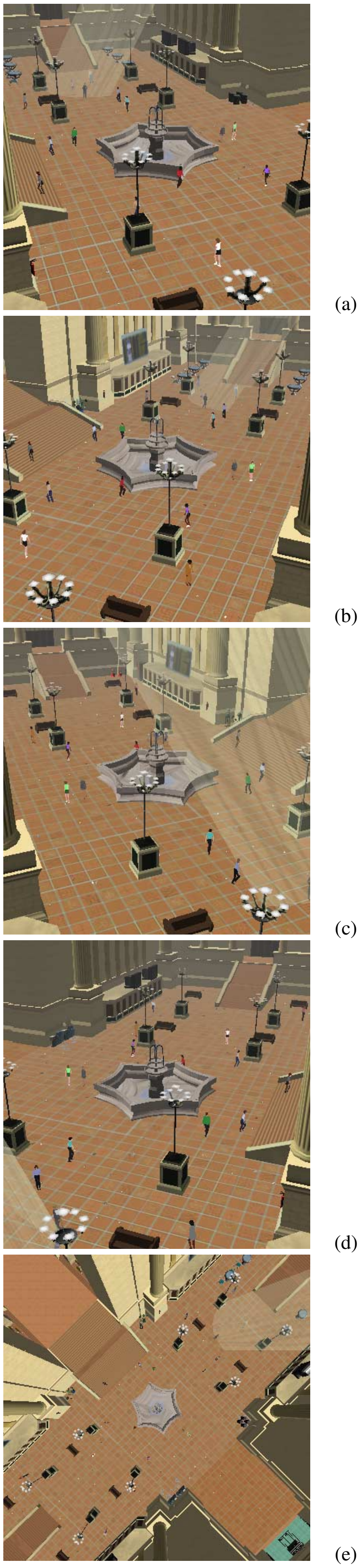


Figure 7: (a)-(d) Wide-FOV stationary cameras situated at the 4 corners of the main waiting room in the train station. (e) A fish-eye camera mounted at the ceiling of the waiting room. These cameras are calibrated and the 3D position of a pedestrians is estimated through triangulation.

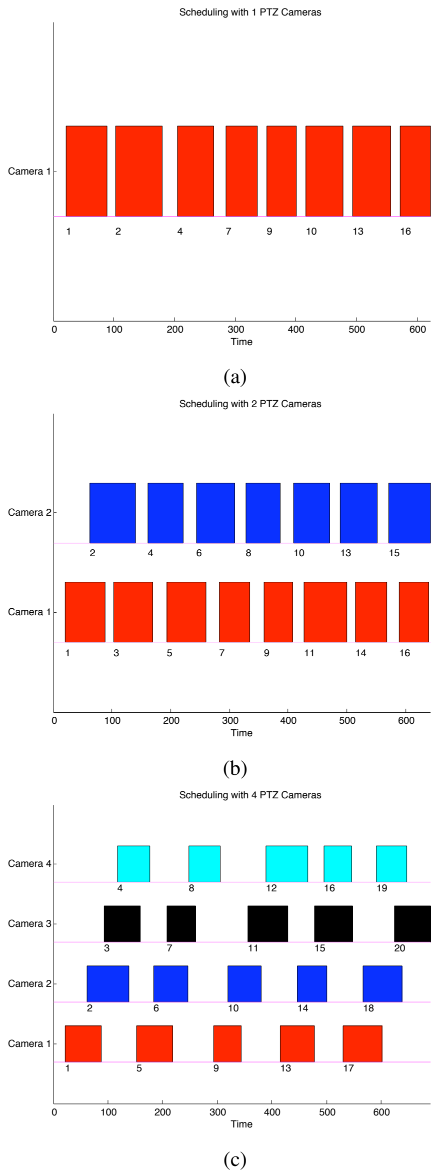


Figure 8: Pedestrians are assigned unique identifiers based on their entry times; e.g., Pedestrian 1 always enters the scene at the same time or before the arrival of Pedestrian 2. (a)–(c) Twenty pedestrians are present in the scene. (a) The scheduling policy for one camera: Camera 1 successfully recorded pedestrians 1, 2, 4, 7, 9, 10, 13, and 16. (b)–(c) Adding more cameras improves the chances of viewing more pedestrians. Only pedestrians 12, 17, 18, 19, and 20 go unnoticed when two cameras are handy; whereas, with four cameras all pedestrians are observed.

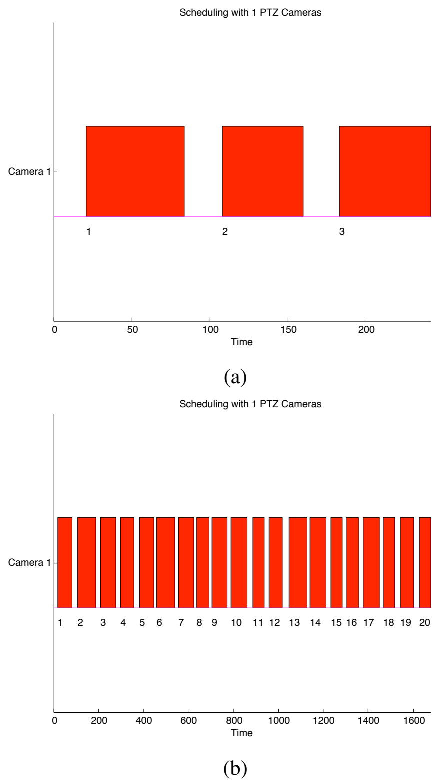


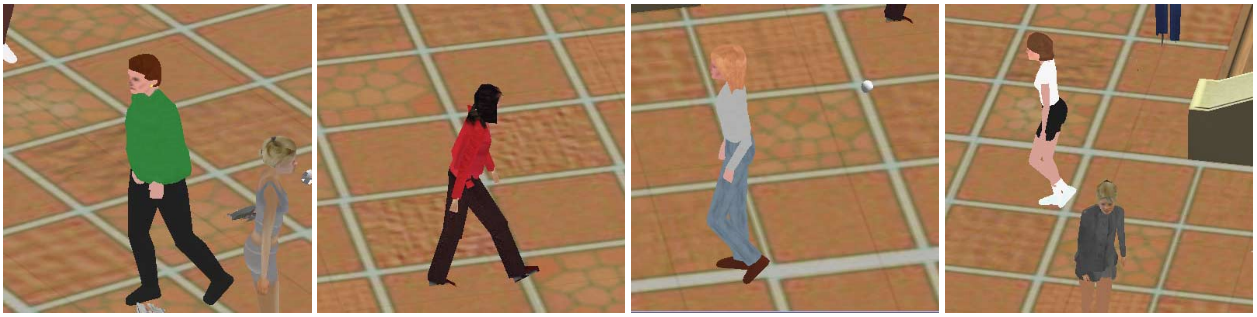
Figure 9: (a) The scene is populated with only three pedestrians. (b) Twenty pedestrians, who tend to stick around, are simulated. The chances of a given set of cameras to view the pedestrians present in the scene increase (a) when there are fewer pedestrians or (b) when pedestrians tend to linger longer in the area.

and it correctly scheduled the available cameras using a weighted round-robin scheduling with an FCFS+ priority policy for all cases.

When only one PTZ camera is available, pedestrians 1, 2, 4, 7, 9, 10, 13, and 16 are recorded (Figure 8(a)); however, pedestrians 3, 5, 6, 8, 11, 12, 14, 15, 17, 18, 19, and 20 go unnoticed, because they left the scene before the camera had an opportunity to observe them. Figure 8(b) and (c) shows the results from the same run with two and four active cameras, respectively. In the two-camera case, even though the performance has improved significantly from the addition of a camera, pedestrians 12, 17, 18, 19, and 20 still go unnoticed. With four active cameras, the system is now able to observe every pedestrian. These results support the intuitive expectation that the chances of viewing multiple pedestrians improve as more cameras become available.

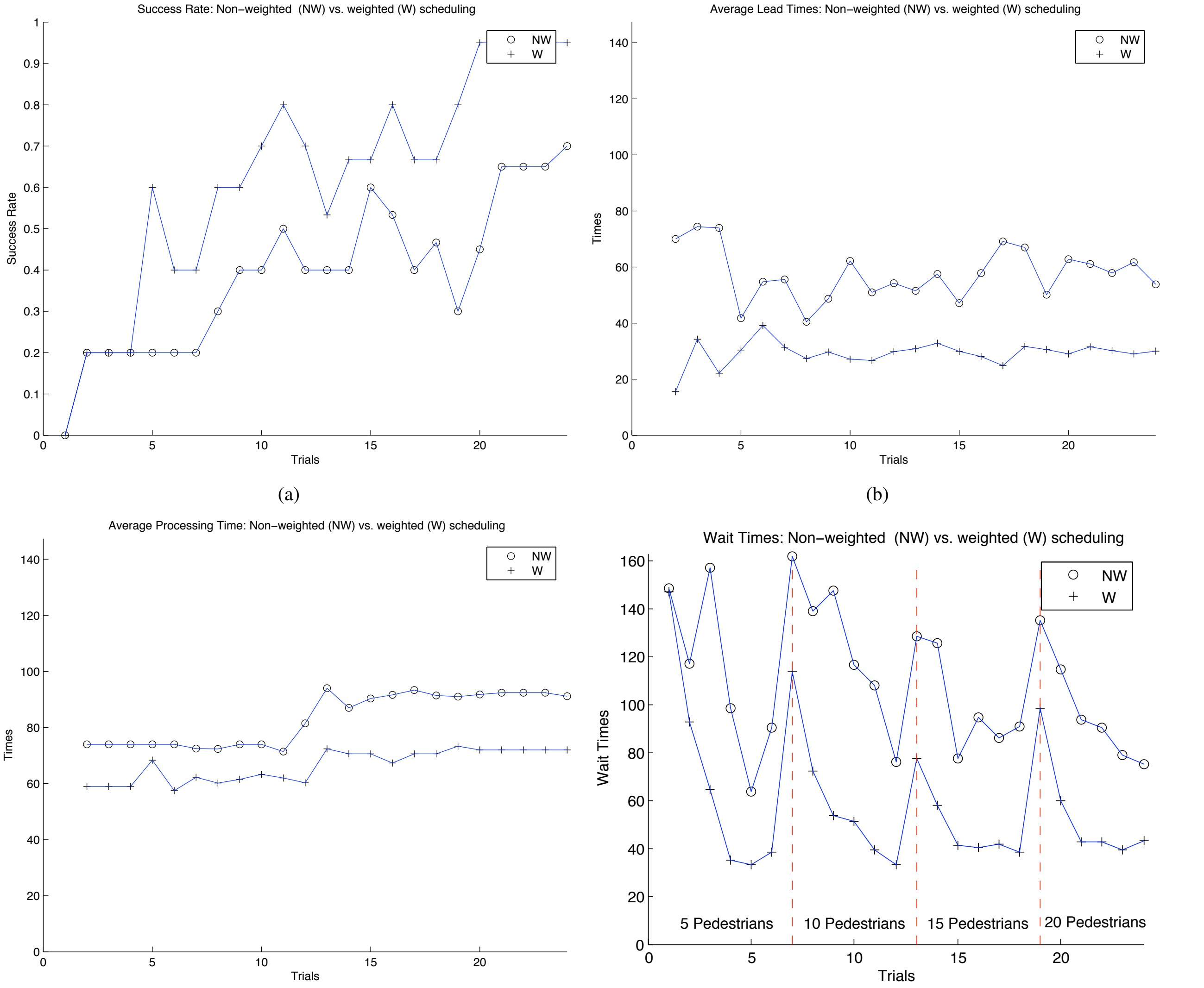
In Figure 9(a), we have populated the virtual train station with only three autonomous pedestrians, leaving all other parameters unchanged. Given that there are now only three pedestrians in the scene, even a single camera successfully observes them. Next, we ran the simulation with twenty pedestrians (Figure 9(b)). This time, however, we changed the behavior settings of the pedestrians, so the pedestrians tend to linger in the waiting room. Here too, a single camera successfully observed each of the twenty pedestrians. We conclude that even a small number of cameras can perform satisfactorily when there are either few pedestrians in the scene or when the pedestrians tend to spend considerable time in the area.

In Figure 11, we compare the scheduling scheme that treats all cameras equally with the weighted scheduling scheme that takes into account the suitability of any camera in observing a pedestrian. As expected, the weighted scheduling scheme outperforms its non-weighted counterpart. The weighted scheduling scheme has higher success rates, which is defined as the fraction of pedestrians successfully recorded, and lower average lead time, processing time (the time spent recording the video of a pedestrian), and wait time (the time elapsed between the entry of a pedestrian and when the



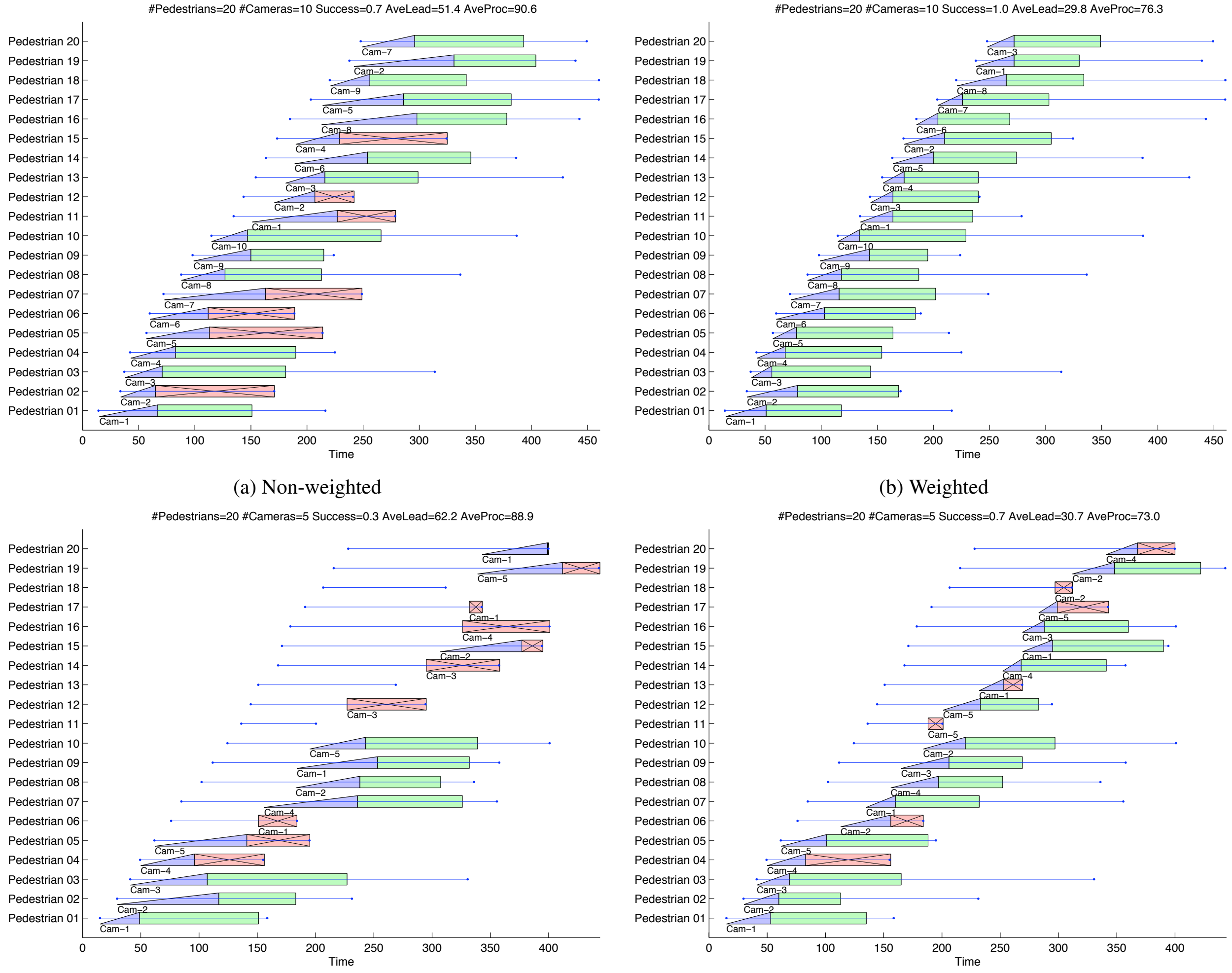
(a) (b) (c) (d)

Figure 10: A sampling of close-up images captured by the PTZ cameras.



(a) (b) (c) (d)

Figure 11: A comparison of Weighted and Non-weighted scheduling schemes. The weighted scheduling strategy, which takes into account the suitability of a camera for recording a particular pedestrian, outperforms its non-weighted counterpart as evident from its higher success rates (a) and shorter lead (b), processing (c), and wait (d) times. The displayed results are averaged over several runs of each trial scenario. Trials 1–6 comprise 5 pedestrians and 1, 2, 3, 4, 5, and 6 cameras, respectively. Trials 7–12 comprise 10 pedestrians and 3, 4, 5, 6, 7, and 8 cameras, respectively. Trials 13-18 comprise of 15 pedestrians and 5, 6, 9, 10, 11, and 12 cameras, respectively. Trials 19–24 comprise of 20 pedestrians with 5, 8, 10, 13, 15, and 18 cameras, respectively.



(a) Non-weighted

(b) Weighted

(d) Non-weighted

(d) Weighted

Figure 12: Scheduling results for Trials 19 and 21. Blue lines represent the entry and exit times, the blue triangles represent the lead times, the Green rectangles represent the processing times, and the Red crossed rectangles represent an aborted attempt at capturing the video of a pedestrian.

camera begins fixating on the pedestrian). The lower average lead and processing times are a direct consequence of how we compute the suitability of a camera for recording a pedestrian. An interesting observation is that the average wait times do not necessarily decrease as we increase the number of cameras. Figure 12 shows detailed results for two scenarios, one with 20 pedestrians and 5 available cameras and the other with 20 pedestrians and 10 cameras.

7. CONCLUSION

We envision future surveillance systems to be networks of stationary and active cameras capable of providing perceptive coverage of extended environments with minimal reliance on a human operator. Such systems will require not only robust, low-level vision routines, but also novel sensor network methodologies. The work presented in this paper is a step toward the realization of these new sensor networks, as was our work in [4].

We have presented a scheduling strategy for intelligently managing multiple PTZ cameras in order to satisfy the challenging task of capturing without human assistance close-up biometric videos of pedestrians present in a scene. We assume that the stationary cameras are calibrated, but that the PTZ cameras are uncalibrated. At present, predicting pedestrian behaviors is at best an inexact science, so we have intentionally avoided scheduling policies that depend on predictions about the future, as the results will degrade when predictions are poor. Instead, we have found the FCFS+ tie breaking policy to be the most suitable one for our purposes.

We have demonstrated our prototype surveillance system in a virtual train station environment populated by autonomous, lifelike pedestrians. This simulator facilitates our ability to design large-scale sensor networks and experiment with them on commodity personal computers. The future of such advanced simulation-based approaches appears promising for the purposes of low-cost design and experimentation.

In future work, we intend to evaluate our scheduling policy more rigorously. Also, since scalability is an issue when dealing with numerous active cameras spread over a large area, we hope to tackle the scalability issue by investigating distributed scheduling strategies.

8. ACKNOWLEDGEMENTS

The research reported herein was supported in part by a grant from the Defense Advanced Research Projects Agency (DARPA) of the Department of Defense. We thank Dr. Tom Strat of DARPA for his generous support and encouragement. We also thank Wei Shao and Mauricio Plaza-Villegas for their invaluable contributions to the implementation of the Penn Station simulator. Mauricio Plaza-Villegas implemented a prototype virtual vision infrastructure within the simulator.

9. REFERENCES

[1] D. Terzopoulos and T. Rabie, "Animat vision: Active vision in artificial animals," *Videre: Journal of Computer Vision Research*, vol. 1, pp. 2–19, September 1997.

[2] D. Terzopoulos, "Perceptive agents and systems in virtual reality," in *Proc. 10th ACM Symposium on Virtual Reality Software and Technology*, (Osaka, Japan), pp. 1–3, October 2003.

[3] W. Shao and D. Terzopoulos, "Autonomous pedestrians," in *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, (Los Angeles, CA), pp. 19–28, July 2005.

[4] F. Qureshi and D. Terzopoulos, "Towards intelligent camera networks: A virtual vision approach," in *Proc. The Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, (Beijing), October 2005. In press.

[5] F. Pedersini, A. Sarti, and S. Tubaro, "Accurate and simple geometric calibration of multi-camera systems," *Signal Processing*, vol. 77, no. 3, pp. 309–334, 1999.

[6] T. Gandhi and M. M. Trivedi, "Calibration of a reconfigurable array of omnidirectional cameras using a moving person," in *Proc. 2nd ACM International Workshop on Video Surveillance and Sensor Networks*, (New York, NY), pp. 12–19, ACM Press, 2004.

[7] R. Collins, O. Amidi, and T. Kanade, "An active camera system for acquiring multi-view video," in *Proc. International Conference on Image Processing*, (Rochester, NY), pp. 517–520, September 2002.

[8] J. Kang, I. Cohen, and G. Medioni, "Multi-views tracking within and across uncalibrated camera streams," in *Proc. First ACM SIGMM International Workshop on Video Surveillance*, (New York, NY), pp. 21–33, ACM Press, 2003.

[9] D. Comaniciu, F. Berton, and V. Ramesh, "Adaptive resolution system for distributed surveillance," *Real Time Imaging*, vol. 8, no. 5, pp. 427–437, 2002.

[10] M. Trivedi, K. Huang, and I. Mikic, "Intelligent environments and active camera networks," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 804–809, October 2000.

[11] S. Stillman, R. Tanawongsuwan, and I. Essa, "A system for tracking and recognizing multiple people with multiple cameras," Tech. Rep. GIT-GVU-98-25, Georgia Institute of Technology, Graphics, Visualization, and Usability Center, 1998.

[12] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1355–1360, October 2003.

[13] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proceedings of the IEEE*, vol. 89, pp. 1456–1477, October 2001.

[14] X. Zhou, R. T. Collins, T. Kanade, and P. Metes, "A master-slave system to acquire biometric imagery of humans at distance," in *Proc. First ACM SIGMM International Workshop on Video Surveillance*, (New York, NY), pp. 113–120, ACM Press, 2003.

[15] A. Hampapur, S. Pankanti, A. Senior, Y.-L. Tian, L. Brown, and R. Bolle, "Face cataloger: Multi-scale imaging for relating identity to location," in *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance*, (Washington, DC, USA), pp. 13–21, 2003.

[16] C. J. Costello, C. P. Diehl, A. Banerjee, and H. Fisher, "Scheduling an active camera to observe people," in *Proc. 2nd ACM International Workshop on Video Surveillance and Sensor Networks*, (New York, NY), pp. 39–45, ACM Press, 2004.

[17] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber, "Approximating the throughput of multiple machines in real-time scheduling," *SIAM Journal on Computing*, vol. 31, no. 2, pp. 331–352, 2002.

[18] J. Sgall, "Online scheduling - a survey," in *On-Line Algorithms: The State of the Art, Lecture Notes in Computer Science*, pp. 192–231, Springer-Verlag, 1998.

[19] T. Ling and N. Shroff, "Scheduling real-time traffic in ATM networks," in *Proc. IEEE Infocom*, pp. 198–205, 1996.

[20] R. Givan, E. Chong, and H. Chang, "Scheduling multiclass packet streams to minimize weighted loss," *Queueing Systems: Theory and Application*, vol. 41, no. 3, pp. 241–270, 2002.

[21] N. T. Siebel, *Designing and Implementing People Tracking Applications for Automated Visual Surveillance*. PhD thesis, Dept. of Computer Science. The University of Reading., UK, March 2003.