Surveillance in Virtual Reality: System Design and Multi-Camera Control

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

[1]Department of Computer Science, University of Toronto, Toronto, ON, Canada
[2]Computer Science Department, University of California, Los Angeles, CA, USA

Abstract

This paper advocates a Virtual Vision paradigm and demonstrates its usefulness in camera sensor network research. Virtual vision prescribes the use of a visually and behaviorally realistic virtual environment simulator in the design and evaluation of surveillance systems. Impediments to deploying and experimenting with appropriately complex camera networks makes virtual vision an attractive alternative for many vision researchers who are motivated to investigate high level multi-camera control issues within such networks. In particular, we present two prototype surveillance systems comprising passive and active pan/tilt/zoom cameras. We deploy these systems in a virtual train station environment populated by autonomous, lifelike virtual pedestrians. The easily reconfigurable virtual cameras situated throughout this environment generate synthetic video feeds that emulate those acquired by real surveillance cameras monitoring extensive public spaces. Our novel multi-camera control strategies enable the cameras to collaborate in persistently observing pedestrians of interest that move across their fields of view and in capturing close-up videos of pedestrians as they travel through designated areas. The sensor networks support task-dependent camera node selection and aggregation through local decision-making and inter-node communication. Our approach to multi-camera control is robust to node failures and message loss.

1. Introduction

Recent advances in camera and video technologies have made it possible to network numerous video cameras together in order to provide visual coverage of extensive public spaces such as airports and train stations. As the size of the camera network grows and the level of activity in the public space increases, it becomes infeasible for human operators to monitor the multiple video streams and identify all events of possible interest, nor even to control individual cameras in performing advanced surveillance tasks, such as zooming in on a moving subject of interest to acquire one or more facial snapshots. Consequently, a timely challenge for computer vision researchers is to design camera sensor networks capable of performing visual surveillance tasks automatically, or at least with minimal human intervention.

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 desired sensing tasks. Imagine a situation where the camera network is asked to capture high-resolution videos of every pedestrian that passes through a designated area.[1] Passive cameras alone cannot satisfy this requirement. Active pan/tilt/zoom (PTZ) cameras must be recruited 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. A resource management strategy can enable the cameras to decide autonomously how best to allocate their time in collectively observing 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 spent 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

Deploying a large-scale surveillance system is a major undertaking whose cost can easily be prohibitive for most computer vision researchers interested in designing and experimenting with multi-camera systems. Moreover, privacy laws impede the monitoring of people in public spaces for experimental purposes. To overcome such obstacles, we advocate *Virtual Vision*, a paradigm that prescribes visually and behaviorally realistic virtual environments for the design of simulated surveillance systems and the meaningful experimentation with such systems. Cost considerations and legal impediments aside, the use of sufficiently realistic virtual environments also offers significantly greater flexibility during the design and evaluation cycle, thus enabling many more iterations of the scientific method.

[1]The captured video can subsequently serve in further biometric analysis; e.g., with facial, gesture, and/or gait recognition routines.

1-4244-1180-7/07/$25.00 ©2007 IEEE

Concourses and platforms Waiting room Arcade

Figure 3. A large-scale virtual train station (Penn Station) populated by self-animating virtual humans.

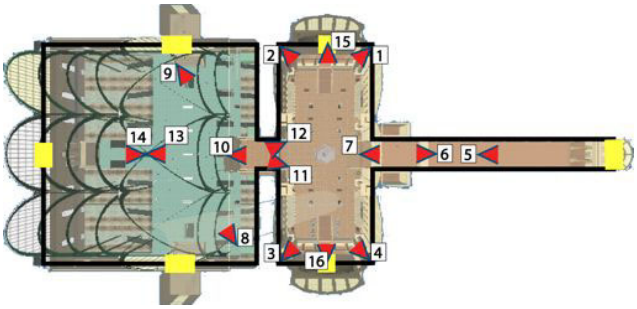


Figure 1. Plan view of the virtual Penn Station environment with the roof not rendered, revealing the concourses and train platforms (left), the main waiting room (center), and the long shopping arcade (right). (The yellow rectangles indicate station pedestrian portals.) An example visual sensor network comprising 16 active (pan-tilt-zoom) simulated video surveillance cameras is illustrated.

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

Specifically, we demonstrate a surveillance system comprising static and active simulated video cameras that provide perceptive coverage of a large virtual public space; a reconstruction of the original Pennsylvania Train Station in New York City, which was demolished in 1963 (Fig. 1). The virtual cameras situated throughout the expansive chambers of the station generate multiple synthetic video feeds that emulate those generated by real surveillance cameras monitoring public spaces (Fig. 2). The train station is populated by self-animating virtual pedestrians (Fig. 3). The advanced pedestrian animation system combines behavioral, perceptual, and cognitive human simulation algorithms [17]. The simulator can efficiently synthesize well over 1000 pedestrians performing a rich variety of activities in the extensive 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, avoid collisions when locomoting (even though portals and other congested areas), queue in lines as necessary, say, to purchase train tickets at the ticket booths in the main waiting room, sit on benches when they are tired, purchase food from vending machines when they are hungry, etc., and eventually proceed down the stairs from the concourse area to the tracks to catch a train. A graphics pipeline (OpenGL) renders the busy urban scene with considerable geometric and photometric detail, as shown in Fig. 3.

Our unique combination of advanced vision and graphics technologies offers several advantages. First, the virtual cameras are very easily relocated and reconfigured in the virtual environment. Second, the virtual world provides readily accessible ground-truth data for the purposes of surveillance algorithm/system validation. Third, surveillance experiments in the virtual environment are perfectly repeatable. Fourth, simulation time can be prolonged relative to real, "wall-clock time"; i.e., arbitrary amounts of computation can be performed per simulation time unit, thereby enabling competence assessments of collections of sophisticated visual algorithms that cannot run in real time on current surveillance hardware. Finally, our simulator runs on (high-end) commodity PCs, obviating the need to grapple with special-purpose hardware and associated software.

1.2. The Surveillance System

Within the virtual vision paradigm, we develop and evaluate a visual sensor network consisting of fixed, wide field-of-view (FOV) passive cameras and PTZ active cameras.

We develop novel multi-camera control strategies that enable the simulated camera nodes to collaborate both in tracking pedestrians of interest that move across the FOVs of different cameras and in capturing close-up videos of pedestrians as they travel through designated areas. The network supports task-dependent node selection and aggregation through local decision-making and inter-node communication. Treating node assignment conflicts as a constraint satisfaction problem, we propose a solution that is robust against node failures and message loss as it lacks a central controller.

For the task of capturing high-quality videos of pedestrians as they move through a designated area, we assume that the wide-FOV stationary cameras are calibrated,[2] which enables the network to estimate the 3D locations of pedestrians through triangulation. However, we do not require the PTZ cameras to be calibrated. Rather, during an initial learning phase, the PTZ cameras learn a coarse mapping between 3D world 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 3D position. 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. Contributions and Overview

The contributions of the research reported herein are as follows: First, we develop a pedestrian tracker that operates upon the synthetic video captured by our virtual cameras, essentially to faithfully emulate a state-of-the-art tracker operating upon real video. Second, we develop new gaze-direction controllers for active PTZ cameras. Third, we propose a sensor management scheme that appears well suited to the challenges of designing camera networks for surveillance applications that are potentially capable of fully automatic operation. Finally, we demonstrate the advantages of the virtual vision paradigm in designing, experimenting with, and evaluating a prototype large-scale surveillance system.

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 the PTZ active camera controllers and propose a scheme for learning the mapping between 3D world locations and gaze directions. Section 5 introduces our scheduling strategy. We present our results in Section 7 and our conclusions and future research directions in Section 8.

[2]This assumption is justifiable given the success of automated static camera calibration schemes [13, 6].

2. Related Work

The Virtual Vision approach to designing surveillance systems with the help of a virtual environment wherein simulated cameras generate synthetic video feeds of a population of lifelike pedestrians autonomously performing various activities was first proposed in [18]. The paradigm was developed further in our recent work on sensor networks [14].

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 [3, 9]. The emphasis has been on tracking and on model transference from one camera to another, which is required for object identification across multiple cameras [10]. Many 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 [13, 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 [4, 19]. 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.* [19] 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.* [8] 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.* [5] evaluate various strategies for scheduling a single active camera to acquire biometric imagery.

The problem of online scheduling has been studied extensively in the context of scheduling jobs on multitasking computer systems [1, 16] as well as for packet routing in networks [11, 7].

3. Pedestrian Tracking

Our system employs appearance-based models to track pedestrians. Pedestrians are segmented in order to construct color-based signatures (appearance models), which are then matched across subsequent frames. Zooming can drastically change the appearance of a pedestrian, thereby confounding conventional appearance-based schemes. We address this problem by maintaining HSV color histograms for several camera zoom settings for each pedestrian. Thus, an important feature of our pedestrian tracking routine is its ability to operate over a range of camera zoom settings.

(a) (b) (c)

Figure 4. Tracking pedestrians 1 and 3. Pedestrian 3 is tracked successfully; however, (a) track is lost of pedestrian 1 who blends in with the background. (b) The tracking routine loses pedestrian 3 when she is occluded by pedestrian 2, but it regains track of pedestrian 3 when pedestrian 2 moves out of the way (c).

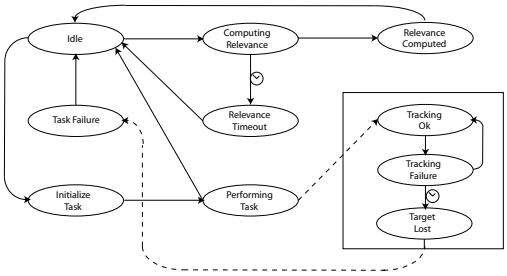


Figure 5. Camera behavioral controller.

The tracking module emulates the abilities and, importantly, the limitations of a state-of-the-art tracking system. In particular, it can lose track due to occlusions, poor segmentation (the quality of segmentation depends upon the amount of noise introduced into the process), or bad illumination (Fig. 4). 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.

The implementation details of our pedestrian tracking module are presented elsewhere [15].

4. PTZ Active Camera Controller

We treat every PTZ active camera as a behavior-based autonomous agent. The overall behavior of the camera is determined by the pedestrian tracking module and the current task. The camera behavioral controller, which we model as an augmented finite state machine (Fig. 5), enables an autonomous camera to achieve its high-level sensing goals as determined by the current task. Typical sensing goals might be, "look at the pedestrian i at location (x, y, z) for t seconds," or "track the pedestrian whose appearance signature is h." Our approach severs the ubiquitous master-slave relationship between the originator of the sensing goal and the camera in the sensor network that will perform the sensing action [19]. Communication requirements and scalability considerations aside, the master-slave relationship between multiple cameras is undesirable as it requires the camera network to be calibrated. Unfortunately, active PTZ cameras are notoriously difficult to calibrate; moreover, the calibration deteriorates over time and needs to be recomputed. Our camera network model does not require calibrated active cameras, so it is easier to change the topology of the network by adding, removing, and/or modifying cameras.[3]

When carrying out a new sensing request, the camera selects a suitable FOV setting and either chooses an appropriate gaze direction using the estimated 3D location of the pedestrian, or performs an exploratory sweep when the pedestrian's 3D location is unavailable. Upon the successful identification of the pedestrian within the FOV, the camera uses fixation and zooming algorithms to follow the subject [15]. The fixation and zooming routines are image driven and do not require any 3D information such as camera calibration or a global frame of reference.

4.1. Gaze Direction Computation

Computing an appropriate gaze direction in order to bring a subject 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 during an initial learning phase by tracking and following a single pedestrian in the scene.

During learning, a pedestrian is directed to move around in the scene. The pedestrian is tracked by the calibrated stationary cameras and the 3D location of the pedestrian is estimated continuously through triangulation. The PTZ cameras are instructed to track the pedestrian and a look-up 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 (α, β) of the camera. We model the relationship between (x, y, z) and (α, β) as a radial basis function (RBF) network that is trained by using the stored (x, y, z) and (α, β) values [2].

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 learned RBF model. This technique provides only a coarse mapping between the 3D points and the camera pan-tilt settings. In practice, however, the mapping is accurate enough to bring the pedestrian within the field of view of the camera.

5. Camera Scheduling

The camera scheduling problem shares many characteristics with a network packet routing problem [5], where network packets are serviced by a router upon arrival. The packet routing problem is an online scheduling problem, as

[3] For the camera scheduling scheme, we assume that the stationary cameras are calibrated in order to estimate the 3D position of a pedestrian. It should, however, be noted that the 3D location of the pedestrian is not required by a PTZ camera for the purposes of fixation/zooming/tracking.

the arrival times of packets are not known *a priori*. Moreover, a packet must be served for a finite duration before it expires and is subsequently dropped by the router. Similarly, in our context, the arrival times of pedestrians entering the scene are not known beforehand and a pedestrian must be observed for some minimum duration by one of the PTZ cameras before (s)he leaves the scene. That minimum time serves as the deadline.

The packet routing problem, however, does not account for all aspects of the problem we confront. 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 observe 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. Third, not every PTZ camera is equally suitable for observing any particular pedestrian, and the suitability of a PTZ camera with respect to observing a pedestrian changes over time.

We propose a weighted round-robin scheduling scheme with a static *First Come, First Serve* (FCFS+) priority policy that strikes a balance between two competing goals: 1) to capture high-quality video for as many as possible, preferably all, pedestrians in the scene and 2) to view each pedestrian for as long or as many times as possible. At one extreme, the camera can follow a pedestrian for his 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 its time transitioning between different pan/tilt/zoom settings.

We model each PTZ camera as a processor whose weights are adjusted dynamically. The weights quantify the suitability of a camera for viewing a particular pedestrian. They are determined by two factors: 1) the adjustments the camera must make in its PTZ settings to look at the pedestrian and 2) the distance separating the pedestrian from the camera. A camera that requires small adjustments in its PTZ settings 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 must adjust itself 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. 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 intervening between the camera and the subject being recorded. We assume that the sensor network stores information about the pedestrians present in the scene, including their arrival times and the most current estimates of their positions and headings. Scene information is available to the scheduler, which assigns cameras to the various pedestrians present in the scene. We specify the minimum length of time that a PTZ camera must spend observing a pedestrian. The cameras use the 3D information to choose an appropriate gaze direction in order to bring the pedestrian into view.

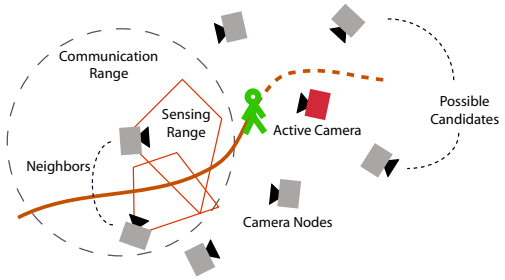
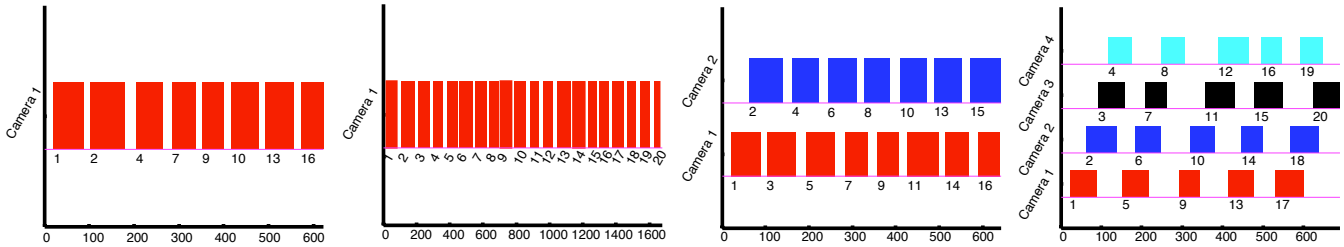


Figure 6. A camera network for video surveillance consists of camera nodes that can communicate with other nearby nodes. Collaborative tracking requires that cameras organize themselves to perform camera handover when the tracked subject moves out of the sensing range of one camera and into that of another.

6. Collaborative Tracking

Let us consider how a sensor network of dynamic cameras may be used in the context of video surveillance (Fig. 6). A human operator spots one or more suspicious pedestrians in one of the video feeds and, for example, requests the network to "track this pedestrian," "zoom in on that pedestrian," or "track the entire group." The successful execution and completion of these tasks requires intelligent allocation and scheduling of the available cameras; in particular, the network must decide which cameras should track the pedestrian and for how long. In our approach, we assume only that a pedestrian can be identified by different cameras with reasonable accuracy and that the camera network topology is known *a priori*. A direct consequence of this approach is that the network can easily be modified through removal, addition, or replacement of camera nodes.

In response to a sensing task, such as, "observe pedestrian i during his stay in the region of interest," wide-FOV passive and PTZ active cameras organize themselves into groups with the aim of fulfilling the task. The *group*, which formalizes the collaboration between member cameras, evolves throughout the lifetime of the task; i.e., member cameras that are not relevant to the task are dropped and new cameras are recruited as they are needed. At any given time, multiple groups can be active, each performing its respective task. Group formation is carried out through local processing at each camera and inter-camera communication. Unlike the camera scheduling mechanism, which assumes calibrated stationary cameras to maintain the scene



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

Figure 7. (a) Single camera, 20 pedestrians, (b) single camera, 20 pedestrian that tend to linger, (c) two cameras, 20 pedestrians, and (d) four cameras, 20 pedestrians.

model and a central scheduler that uses the scene model to assign PTZ cameras to different pedestrians, the collaborative tracking strategy does away altogether with any scene model, camera calibration, and central controller. A camera node can communicate with nearby camera nodes (those that are within its wireless communication range). Furthermore, we assume that each camera node can independently compute its *relevance* to a task [15]. Inspired by the behavior-based autonomous agent design philosophy, we leverage the interaction between the individual nodes to generate global task-directed behavior.

When a suspicious pedestrian is selected (either by a human operator, or automatically by a video analysis procedure) in a camera c, a group is initiated. Initially, the group has only one member, camera c, which also acts as the group's supervisor. To recruit new cameras for the current task, camera c asks nearby cameras to compute their *relevance* to the task. Some of the nearby cameras send their relevance to camera c, and those cameras with relevance values greater than a predefined threshold are asked to join the group. One of the member cameras acts as the multi-camera group supervisor, and this camera decides which new nodes should be asked to join the group. The supervisor node removes a member camera from the group when the camera ceases to be relevant to the task; e.g., when the pedestrian has moved out of the sensing range of a camera. Group formation is relatively straightforward when there is no resource contention—i.e., when multiple tasks do not require the same camera for successful operation—the supervisor simply chooses cameras with higher relevance values with respect to the current task. A group vanishes when none of the cameras can perform the current task; e.g., when the tracked pedestrian leaves the designated area.

Inter-group conflicts, which arise when multiple groups require the same cameras, are resolved within a Constraint Satisfaction Problem (CSP) framework [12]. Here, each group is treated as a variable whose domain consists of non-empty subsets of the set of relevant cameras. The CSP is centralized at the supervisor of one of the conflicting groups and solved using *backtracking*. Under the assumption that the quality of a solution can only increase as we assign values to more variables, our scheme guarantees optimal sensor assignment. The solution is then sent to every affected camera. A limitation of our approach to conflict resolution is that, currently, a camera can be engaged only in a single task at a time.

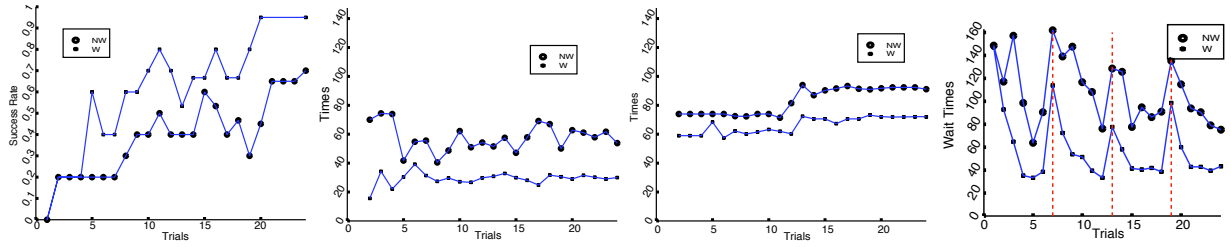
Our proposed communication model also takes into consideration camera and inter-camera communication failures. A communication failure is treated as a node failure. The supervisor responds to a member camera failure by simply removing it from the group. On the other hand, supervisor failure resolution is more involved. When a member camera detects a supervisor camera failure, it selects itself to be the group supervisor, thereby initiating a single-camera group. An actual or perceived supervisor camera failure can therefore give rise to multiple single-node groups performing the same task. These groups are later merged to form one group, establishing collaboration between these cameras. For the technical details, we refer the reader to [15].

The proposed scheme lies between a fully distributed and a totally centralized scheme. Group formation is distributed and independent of the size of the network, while group conflict resolution is centralized within groups. We conclude that our scheme is scalable when group sizes are kept small. Indeed, we expect group sizes to be small due to spatial constraints.

7. Results

To conduct camera scheduling experiments, we populated the virtual train station with up to twenty autonomous pedestrians, who enter, wander, and exit the main waiting room of their own volition. We tested our scheduling strategy in various scenarios using anywhere from 1 to 18 PTZ active cameras. For each trial, we placed a wide-FOV passive camera at each corner of the main waiting room. We also affixed a fish-eye camera to the ceiling of the waiting room. These passive cameras were used to estimate the 3D location of the pedestrians. As expected, the chances that a given set of cameras can observe the pedestrians present in the scene increase when there are fewer pedestrians or when pedestrians tend to linger longer in the area (Fig. 7).

In Fig. 8, we compare the weighted and non-weighted scheduling schemes (averaged over multiple runs). The weighted scheduling scheme outperforms its non-weighted



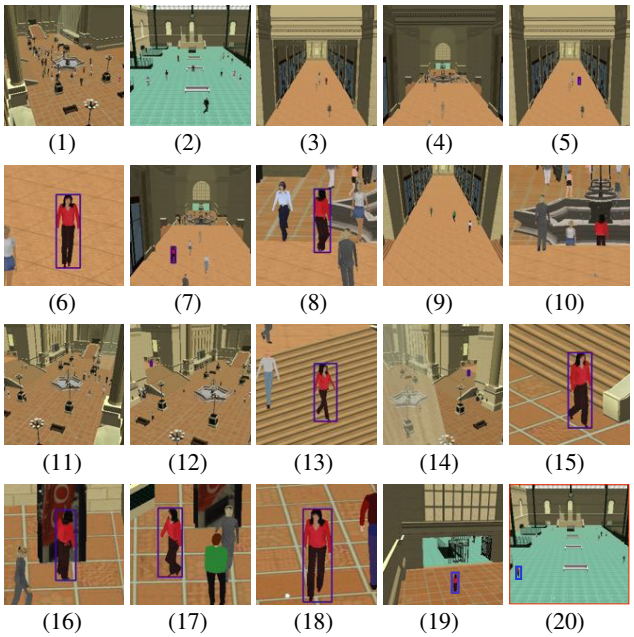
(a) (b) (c) (d)

Figure 8. Comparisons of Weighted (W, circled curve) and Non-Weighted (NW) 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 is evident from its (a) higher success rates and (b) shorter lead, (c) processing, and (d) wait times. The displayed results are averaged over several runs of each trial scenario. Trials 1–6 involve 5 pedestrians and 1, 2, 3, 4, 5, and 6 cameras, respectively. Trials 7–12 involve 10 pedestrians and 3, 4, 5, 6, 7, and 8 cameras, respectively. Trials 13-18 involve 15 pedestrians and 5, 6, 9, 10, 11, and 12 cameras, respectively. Trials 19–24 involve 20 pedestrians with 5, 8, 10, 13, 15, and 18 cameras, respectively.

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 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. As expected, the average wait times typically decrease as we increase the number of cameras.

In our collaborative tracking experiments to date, we have tested our visual sensor network system with up to 16 stationary and pan-tilt-zoom cameras (Fig. 1), and we have populated the virtual Penn Station with up to 100 pedestrians. The sensor network correctly assigned cameras in most cases. As the number of pedestrians that appear similar grows, the tracking module has increasing difficulty following the same pedestrian, and poor pedestrian tracking adversely affects the performance of the camera network.

For the example shown in Fig. 9, we placed 16 active PTZ cameras in the train station, as shown in Fig. 1. An operator selects the pedestrian with the red shirt in Camera 7 (Fig. 9(5)) and initiates the "follow" task. Camera 7 forms the task group and begins tracking the pedestrian. Subsequently, Camera 7 recruits camera 6, which in turn recruits Cameras 2 and 3 to track the pedestrian. Camera 6 becomes the supervisor of the group when Camera 7 loses track of the pedestrian and leaves the group. Subsequently, Camera 6 experiences a tracking failure, promotes Camera 3 to group supervisor, and leaves the group. Cameras 2 and 3 track the pedestrian during her stay in the main waiting room, where she also visits a vending machine. When the pedestrian starts walking towards the concourse, Cameras 10 and 11 take over the group from Cameras 2 and 3. Cameras 2 and 3 leave the group and return to their default modes. Later, Camera 11, which is now acting as the group's supervisor, recruits Camera 9, which tracks the pedestrian as she enters the concourse.



(1) (2) (3) (4) (5)
(6) (7) (8) (9) (10)
(11) (12) (13) (14) (15)
(16) (17) (18) (19) (20)

Figure 9. A pedestrian is successively tracked by multiple cameras (see Fig. 1) for 15 minutes as she makes her way through the station to the concourse. (1-4) Cameras 1, 9, 7, and 8 observing the station (elapsed time: 30 sec). (5) Operator selects a pedestrian in feed 7 (1.7 min). (6) Camera 7 has zoomed in on the pedestrian (2 min). (7) Camera 6, which is recruited by Camera 7, acquires the pedestrian (2.2 min). (8) Camera 6 zooms in on the pedestrian (3 min). (9) Camera 7 reverts to its default mode after losing track of the pedestrian—it is now ready for another task (3.5 min). (10) Camera 6 has lost track of the pedestrian (4.2 min). (11) Camera 2 (3 min). (12) Camera 2, which is recruited by Camera 6, acquires the pedestrian (4 min). (13) Camera 2 tracking the pedestrian (4.3 min). (14) Camera 3 is recruited by the Camera 6; Camera 3 has acquired the pedestrian (4 min). (15) Camera 3 zooming in on the pedestrian (5 min). (16) Pedestrian is at the vending machine (6 min). (17) Pedestrian is walking towards the concourse (13 min). (18) Camera 10 is recruited by Camera 3; Camera 10 is tracking the pedestrian (13.4 min). (19) Camera 11 is recruited by Camera 10 (14 min). (20) Camera 9 is recruited by Camera 10 (15 min).

8. 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. As was our earlier work in [14], the work presented in this paper is a step toward the realization of these new generations of sensor networks.

We have described two prototype surveillance systems capable of autonomously carrying out high-level visual surveillance tasks. Our first surveillance system comprised calibrated passive and uncalibrated active cameras, and it relied upon a scheduling strategy for managing multiple active cameras in order to capture close-up videos of pedestrians as they travel through designated areas. The second surveillance system intelligently managed multiple passive and active cameras to track pedestrians of interest that move across the FOVs of different cameras. Here, we assumed uncalibrated passive and active cameras.

We have demonstrated our prototype surveillance system in a virtual train station environment populated by autonomous, lifelike pedestrians. This simulator has facilitated our ability to design large-scale visual 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 facile experimentation.

In future work, we intend to tackle the scalability issue by investigating distributed scheduling and conflict resolution strategies. Additionally, we are investigating the combination of camera scheduling and camera grouping within a unified framework.

9. 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 Tom Strat, formerly of DARPA, for his generous support and encouragement. We gratefully acknowledge Wei Shao and Mauricio Plaza-Villegas for their invaluable contributions to the implementation of the Penn Station simulator. Finally, we appreciate the encouragement and goodwill that we have received from Larry Davis.

References

[1] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber. Approximating the throughput of multiple machines in real-time scheduling. *SIAM Journal on Computing*, 31(2):331–352, 2002.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Number 0-19-853849-9. Oxford University Press, Nov. 1995.

[3] R. Collins, O. Amidi, and T. Kanade. An active camera system for acquiring multi-view video. In *Proc. International Conference on Image Processing*, pages 517–520, Rochester, NY, USA, Sept. 2002.

[4] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89(10):1456–1477, Oct. 2001.

[5] 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*, pages 39–45, New York, NY, 2004. ACM Press.

[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*, pages 12–19, New York, NY, 2004. ACM Press.

[7] R. Givan, E. Chong, and H. Chang. Scheduling multiclass packet streams to minimize weighted loss. *Queueing Systems: Theory and Application*, 41(3):241–270, July 2002.

[8] 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*, pages 13–21, Washington, DC, USA, 2003.

[9] 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*, pages 21–33, New York, NY, 2003. ACM Press.

[10] 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*, 25(10):1355–1360, Oct. 2003.

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

[12] J. K. Pearson and P. G. Jeavons. A survey of tractable constraint satisfaction problems. Technical Report CSD-TR-97-15, Royal Holloway, University of London, July 1997.

[13] F. Pedersini, A. Sarti, and S. Tubaro. Accurate and simple geometric calibration of multi-camera systems. *Signal Processing*, 77(3):309–334, 1999.

[14] 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 (VS-PETS05)*, pages 177–184, Beijing, China, Oct. 2005.

[15] F. Z. Qureshi. *Intelligent Perception in Virtual Camera Networks and Space Robotics*. PhD thesis, Department of Computer Science, University of Toronto, January 2007.

[16] J. Sgall. Online scheduling: A survey. In *On-Line Algorithms: The State of the Art, Lecture Notes in Computer Science*, pages 192–231. Springer-Verlag, 1998.

[17] W. Shao and D. Terzopoulos. Autonomous pedestrians. In *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 19–28, Los Angeles, CA, July 2005.

[18] D. Terzopoulos. Perceptive agents and systems in virtual reality. In *Proc. 10th ACM Symposium on Virtual Reality Software and Technology*, pages 1–3, Osaka, Japan, Oct. 2003.

[19] 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*, pages 113–120, New York, NY, 2003. ACM Press.