Surveillance in Virtual Reality: System Design and Multicamera Control

Anonymous CVPR submission

Paper ID 1400

Abstract

We describe a prototype surveillance system featuring a visual sensor network comprising wide field-of-view passive cameras and pan/tilt/zoom active cameras. Novel multicamera control strategies enable the 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 sensor network supports task-dependent node selection and aggregation through local decision-making and inter-node communication. We treat node selection as a constraint satisfaction problem. Lacking a central controller, our solution is scalable and robust against node failures. Impediments to deploying and experimenting with appropriately complex camera networks in large, busy public spaces would make the work reported in this paper more or less infeasible for computer vision researchers like ourselves. Hence, a unique centerpiece of our work is the exploitation of a visually and behaviorally realistic virtual environment simulator in the design and evaluation of surveillance systems. In particular, we have developed our 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 acquired by real surveillance cameras monitoring public spaces.

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 large 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, or 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 region of interest.[1] Passive cameras alone cannot satisfy this requirement and 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 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 subject 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

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 multicamera systems. Moreover, privacy laws impede the monitoring of people in public spaces for experimental purposes. To overcome these 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.

Specifically, we demonstrate a surveillance system comprising static and active simulated video cameras that pro-

[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 3. A large-scale virtual train station populated by self-animating virtual humans.

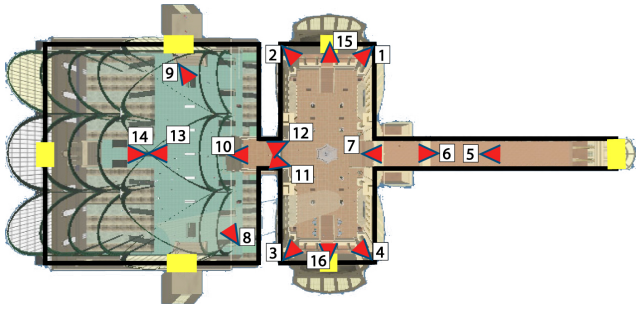

Figure 1. Plan view of the virtual Penn Station environment with the roof not rendered, revealing the concourses and train tracks (left), the main waiting room (center), and the long shopping arcade (right). (The yellow rectangles indicate station pedestrian portals.) An example visual sensor network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

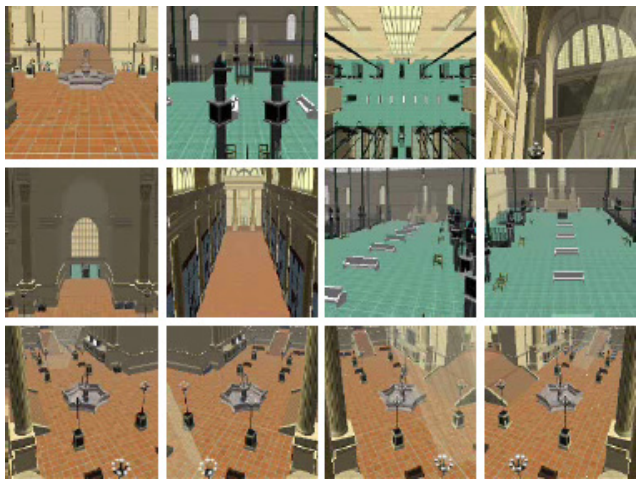

Figure 2. Synthetic video feeds from multiple virtual surveillance cameras situated in the (empty) Penn Station environment.

vide perceptive coverage of a large virtual public space; in our case, a train station (Fig. 1), a reconstruction of the original Pennsylvania Station in New York City, which was demolished in 1963. The virtual cameras situated throughout the expansive chambers of the station generate multiple synthetic video feeds (Fig. 2) that emulate those generated by real surveillance cameras monitoring public spaces. The station is populated by autonomously self-animating virtual pedestrians (Fig. 3). The advanced pedestrian animation system combines behavioral, perceptual, and cognitive human simulation algorithms [16]. 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 to the concourse area and down to the train tracks. A graphics pipeline renders the busy urban scene with considerable geometric and photometric detail, as shown in Figure 3.

Our unique combination of vision and advanced 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, 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 one to evaluate the competence of collections of sophisticated visual surveillance algorithms that cannot currently be expected to run in real time. Finally, our simulator runs on (high-end) commodity PCs, obviating the need to grapple with special-purpose hardware and 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 multicamera 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 selection as a constraint satisfaction problem, we propose a solution that is scalable and robust against node failures, since 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 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 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 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 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 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 7 and our conclusions and future research directions in Section 8.

[2] This assumption is justifiable given the success of numerous automatic static camera calibration schemes [14, 7].

(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).

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 [3, 10, 5, 18, 17]. The emphasis has been on tracking and on model transference from one camera to another, which is required for object identification across multiple cameras [11]. 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 [14, 7].

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.* [9] 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.* [6] 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 [1, 15] as well as for packet routing in networks [12, 8].

3. Pedestrian Tracking

Our system employs appearance-based models to track pedestrians. Pedestrians are segmented in order to construct color-based pedestrian signatures (appearance models), which are then matched across subsequent frames. Zooming can drastically change the appearance of a pedes-

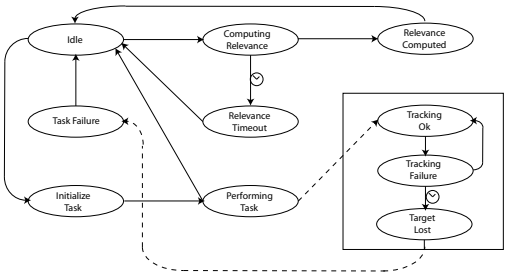

Figure 5. Camera behavioral controller.

trian, 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.

The tracking module mimics the abilities and 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 [citation to a published workshop paper].

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/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. 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 and follow 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 the network packet routing problem [6], where network packets are serviced by a router upon arrival. The packet routing problem is an online scheduling problem, as 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 case, 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.

[3] For the camera scheduling scheme, we assume that 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 packet routing problem, however, does not account for all aspects of the problem that 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 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. 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, and 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 amount of adjustments the camera needs to 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 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 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 coming in-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. The 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 looking at a pedestrian. The cameras use the 3D information stored in the scene model to choose an appropriate gaze direction in order to bring the pedestrian into view.

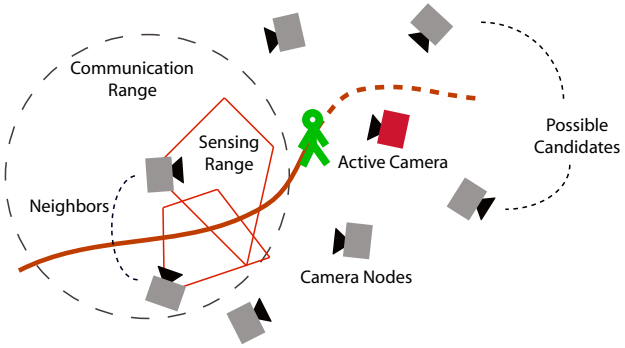

Figure 6. A camera network for video surveillance consists of camera nodes that can communicate with other nearby nodes. Collaborative tracking requires that cameras organize themselves to perform camera handover when the tracked subject moves out of the sensing range of one camera and into that of another.

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 model and a central scheduler that uses the scene model to assign PTZ cameras to different pedestrians, the collaborative tracking strategy altogether does away with any scene model, camera calibration, and the central controller. A

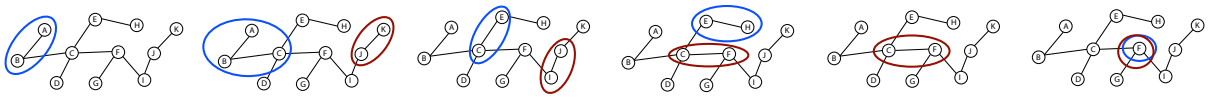

Figure 7. Grouping and conflict resolution. Left to right. (1) Group 1: A and B; possible candidate: C (2) Group 1: A, B, and C; possible candidates: E, F, and D. Group 2: J and K; possible candidate: I. (3) Group 1: E and C; possible candidates: H, F, D, and B. Group 2: J and I; possible candidate: F. (4) Group 1: E and H; possible candidate C. Group 2: C and F; possible candidates: B, D, G, I, and E. (5) Group 1 and 2 require the same resources, so Group 1 disbanded; task failure. (6) A situation where both groups successfully use the same nodes; e.g., imagine two groups tracking two pedestrians that started walking together.

camera node can communicate with nearby camera nodes (those that are within its communication range). Furthermore, we assume that each camera node can independently compute its *relevance* to a task [see supplemental PDF document for the details]. Inspired by the behavior-based autonomous agent design paradigm, we leverage the interaction between 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 with respect to the current task 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 group supervisor for groups consisting of multiple cameras, 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 cease 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 (Fig. 7(5-6)), are resolved within a Constraint Satisfaction Problem (CSP) framework [13]. 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*. The process generates multiple solutions, which are then ranked using the relevance values of the cameras (with respect to the involved groups), and the best solution is selected to find optimal assignment. The solution is then sent to every affected camera. A limitation of our approach to conflict resolution is that, currently, a camera can only be engaged 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, the supervisor failure resolution is more involved. When a member camera detects 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 a single group, establishing collaboration between these cameras. [For the technical details, we refer the reader to the PDF document in the supplementary materials.]

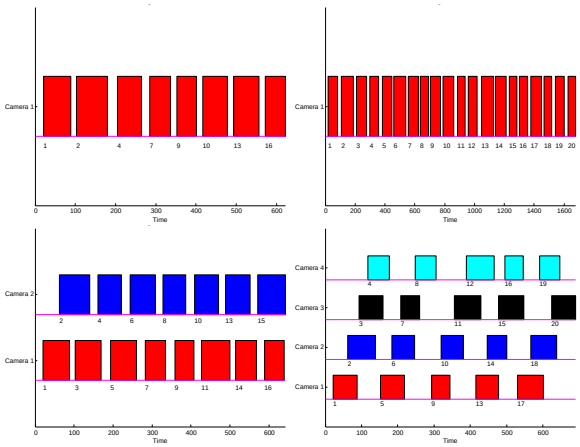

Figure 8. Left to right; top to bottom: (1) one camera, 20 pedestrians, (2) one camera, 20 pedestrian that tend to stick around, (3) two cameras, 20 pedestrians, and 4) four cameras, 20 pedestrians.

7. Results

Note to referee: *Please download and unzip the file 1400-supplemental.zip and browse* **index.html** *in the resulting directory for annotated links to video demos and other supplemental material.*

To conduct camera scheduling experiments, we populated the virtual train station with up to twenty autonomous pedestrians, who enter, wander, and leave 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

CVPR #1400

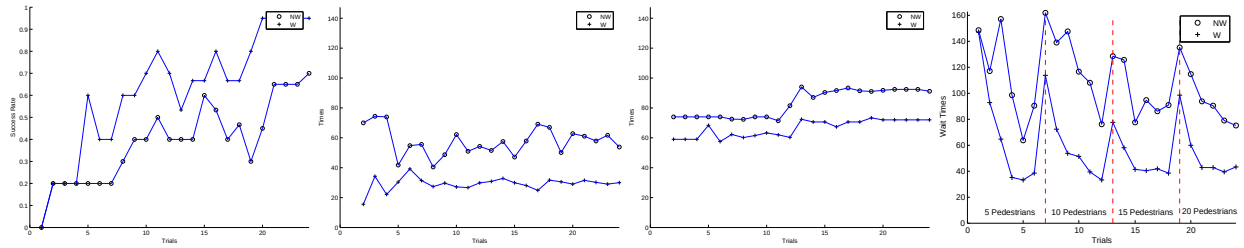

Figure 9. Comparisons of Weighted (circled curve) and Non-weighted scheduling schemes. The weighted scheduling strategy, which takes into account the suitability of a camera for recording a particular pedestrian, outperforms its non-weighted counterpart as evident from its (1) higher success rates and (2) shorter lead, (3) processing, and (4) 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.

location of the pedestrians. As expected, the chances of a given set of cameras to view the pedestrians present in the scene increase when there are fewer pedestrians or when pedestrians tend to linger longer in the area (Figure 8).

In Figure 9, we compare the weighted and non-weighted scheduling schemes (averaged over multiple runs). 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 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.

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 right pedestrian, and poor pedestrian tracking adversely affects the performance of the camera network.

For the example shown in Fig. 10, 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. 10(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, sets Camera 3 as the 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.

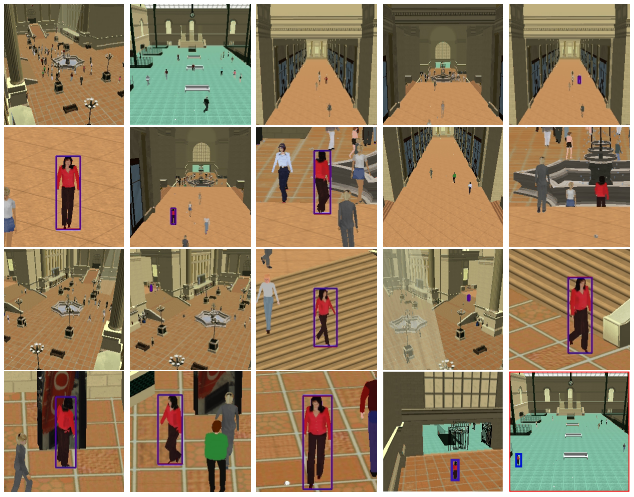

Figure 10. Left to right; top to bottom. A pedestrian is successively tracked by cameras 7, 6, 2, 3, 10, and 9 (see Fig. 1) for 14 minutes as she makes her way through the station to the concourse. (1-4) Cameras 1, 9, 7, and 8 observing the station (elapsed time: 30 sec). (5) Operator selects a pedestrian 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. The work presented in this paper is a step toward the realization of these new sensor networks.

We have presented a scheduling strategy for intelligently managing multiple PTZ cameras in order to track 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. We assume that the PTZ cameras are uncalibrated, but in accomplishing the latter task, we also assume that the stationary cameras are calibrated. 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 has facilitated 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.

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 -name- of DARPA for his generous support and encouragement. We also thank -name- and -name- for their invaluable contributions to the implementation of the Penn Station simulator.

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. 3

[2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Number 0198538642. Oxford University Press, November 1995. 4

[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, September 2002. 3

[4] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89(10):1456–1477, October 2001. 3

[5] D. Comaniciu, F. Berton, and V. Ramesh. Adaptive resolution system for distributed surveillance. *Real Time Imaging*, 8(5):427–437, 2002. 3

[6] 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. 3, 4

[7] 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. 3

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

[9] 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. 3

[10] 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. 3

[11] 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, October 2003. 3

[12] T. Ling and N. Shroff. Scheduling real-time traffic in ATM networks. In *Proc. IEEE Infocom*, pages 198–205, 1996. 3

[13] 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. 6

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

[15] 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. 3

[16] W. Shao and D. Terzopoulos. Autonomous pedestrians. In *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 19–28, Los Angeles, CA, July 2005. 2

[17] S. Stillman, R. Tanawongsuwan, and I. Essa. A system for tracking and recognizing multiple people with multiple cameras. Technical Report GIT-GVU-98-25, Georgia Institute of Technology, Graphics, Visualization, and Usability Center, 1998. 3

[18] M. Trivedi, K. Huang, and I. Mikic. Intelligent environments and active camera networks. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 804–809, October 2000. 3

[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. 3, 4