Chapter 1

VIRTUAL VISION

Virtual Reality Subserving Computer Vision Research for Camera Sensor Networks

Demetri Terzopoulos
Computer Science Department, University of California, Los Angeles
dt@cs.ucla.edu

Faisal Z. Qureshi
Faculty of Science, University of Ontario Institute of Technology
faisal.qureshi@uoit.ca

Abstract Computer vision and sensor networks researchers are increasingly motivated to investigate complex multi-camera sensing and control issues that arise in the automatic visual surveillance of extensive, highly populated public spaces such as airports and train stations. However, they often encounter serious impediments to deploying and experimenting with large-scale physical camera networks in such real-world environments. We propose an alternative approach called "Virtual Vision" that facilitates this type of research through the virtual reality simulation of populated urban spaces, camera sensor networks, and computer vision on commodity computers. We demonstrate the usefulness of our approach by developing two highly automated surveillance systems comprising passive and active pan/tilt/zoom cameras that are deployed in a virtual train station environment populated by autonomous, lifelike virtual pedestrians. The easily reconfigurable virtual cameras distributed in this environment generate synthetic video feeds that emulate those acquired by real surveillance cameras monitoring public spaces. The novel multi-camera control strategies that we describe enable the cameras to collaborate in persistently observing pedestrians of interest and in acquiring close-up videos of pedestrians in designated areas.

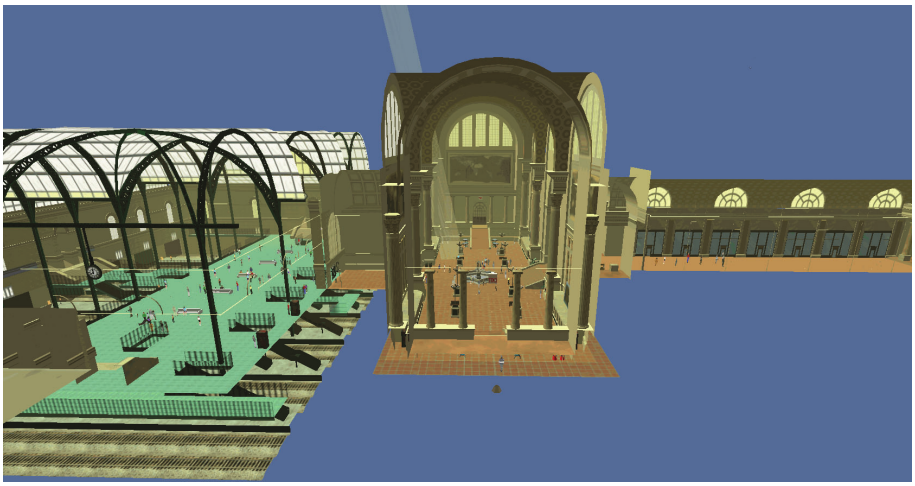
Keywords: Smart cameras, camera networks, sensor networks, computer vision, visual surveillance, persistent human observation, virtual reality

1. Introduction

Future visual sensor networks will rely on smart cameras, which are self-contained vision systems, complete with increasingly sophisticated image sensors, on-board processing and storage capabilities, power, and (wireless) communication interfaces. This opens up new opportunities to develop sensor networks capable of visually surveilling extensive public spaces, disaster zones, battlefields, and even entire ecosystems. These multi-camera systems lie at the intersection of Computer Vision and Sensor Networks and they pose challenging technical problems to researchers in both fields. In particular, as the size of a camera sensor network grows, 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 directly in order to maintain persistent surveillance. Therefore, it is desirable to design intelligent camera sensor networks that are capable of performing advanced visual surveillance tasks autonomously, or at least with minimal human intervention.

Unfortunately, deploying a large-scale physical surveillance system is a major undertaking whose cost can easily be prohibitive for most computer vision and sensor network researchers interested in experimenting with multi-camera systems. As a means of overcoming this barrier to entry, as well as to avoid privacy laws that restrict the monitoring of people in public spaces, we have introduced the *Virtual Vision* paradigm for fostering research in surveillance systems [Terzopoulos, 2003]. Thus far, we have pursued our unique approach using a dynamic virtual environment populated by autonomously self-animating, lifelike virtual pedestrians [Qureshi and Terzopoulos, 2006; Qureshi and Terzopoulos, 2008]. Cost and legal impediments aside, we have also found that virtual vision facilitates the scientific method by offering significant advantages during the surveillance system design and evaluation cycle.

Within the virtual vision framework, we review in this chapter surveillance systems comprising smart cameras that provide perceptive coverage of a large virtual public space—in our case, a reconstruction of New York City's original Pennsylvania Station that was demolished in 1963, populated by autonomously self-animating virtual pedestrians (Figure 1.1). Virtual passive and active cameras situated throughout the expansive chambers of the train station generate multiple synthetic video feeds that emulate those generated by real surveillance cameras monitoring public spaces (Figure 1.2). The advanced pedestrian animation system combines behavioral, perceptual, and cognitive human simulation algorithms [Shao and Terzopoulos, 2005a]. The simulator can efficiently synthesize well over 1000 self-animating pedestrians per-



(a)

(b) Waiting Room (c) Concourses (d) Arcade (e) Platforms

Figure 1.1: A large-scale virtual train station populated by self-animating virtual humans [Shao and Terzopoulos, 2005a].

forming 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 feel tired, purchase food/drinks from vending machines when they feel hungry/thirsty, etc., and proceed from the concourse area down the stairs to the train platforms if they wish to board a train. A graphics pipeline renders the busy urban scene with considerable geometric and photometric detail, as shown in Figure 1.1.

Espousing our virtual vision paradigm, we have developed novel camera control strategies that enable simulated camera nodes to collaborate both in tracking pedestrians of interest that move across the fields of

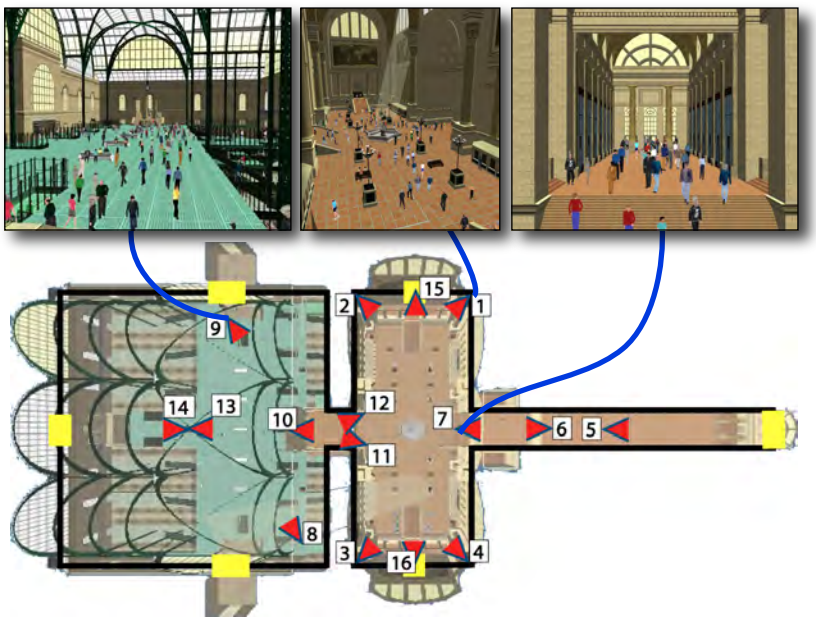


Figure 1.2: Plan view of the (roofless) virtual Penn Station environment, revealing the concourses and train tracks (left), the main waiting room (center), and the shopping arcade (right). (The yellow rectangles indicate pedestrian portals.) An example camera network is illustrated, comprising 16 simulated active (pan-tilt-zoom) video surveillance cameras. Synthetic images from cameras 1, 7, and 9 (from [Shao and Terzopoulos, 2005a]).

view (FOVs) of different cameras and in capturing close-up videos of pedestrians as they travel through a designated area. These virtual camera networks demonstrate the advantages of the virtual vision paradigm in designing, experimenting with, and evaluating prototype large-scale surveillance systems. Specifically, we have studied control and collaboration problems that arise in camera networks by deploying simulated networks within the virtual train station. These simulated networks have performance characteristics similar to those of physical camera networks; e.g., latency, limited bandwidth, communication errors, camera failures, compression artifacts, etc.

2. The Case for Virtual Vision

In virtual vision we combine computer vision and advanced graphics technologies to facilitate our development of camera network control

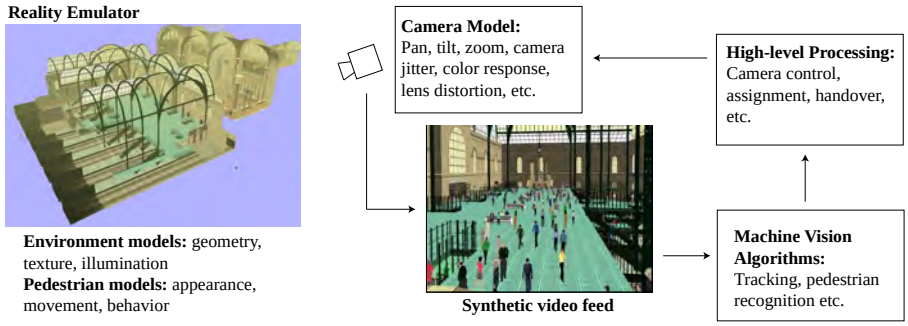


Figure 1.3: The *virtual vision* paradigm.

algorithms, through the deployment of virtual networks in simulated environments (Figure. 1.3). This enables us to investigate high-level control problems, such as camera assignment and handoff, that frequently arise in networks comprising smart cameras under realistic conditions. Virtual vision offers several advantages:

- The virtual vision simulator runs on (high-end) commodity PCs, obviating the need to grapple with special-purpose hardware.[1]
- The virtual cameras are very easily relocated and reconfigured in the virtual environment.
- The virtual world provides readily accessible ground-truth data for the purposes of surveillance algorithm/system validation.
- Experiments are perfectly repeatable in the virtual world, so we can easily modify algorithms and their various parameters and immediately determine the effect.
- Our simulated camera networks run on-line in "real time" within the virtual world, with the virtual cameras actively controlled by the vision algorithms. By suitably prolonging virtual-world time relative to real-world time, we can evaluate the competence of computationally expensive algorithms, thereby gauging the potential payoff of accelerating them through more efficient software and/or dedicated hardware implementations.

An important issue in camera network research is the comparison of camera control algorithms. Simple video capture suffices for gathering benchmark data from time-shared physical networks of passive, fixed cameras, but gathering benchmark data for networks that include any

smart, active PTZ cameras requires scene reenactment for every experimental run, which is almost always infeasible when many human subjects are involved. Costello *et al.* [Costello et al., 2004], who compared various schemes for scheduling an active camera to observe pedestrians, ran into this hurdle and resorted to Monte Carlo simulation to evaluate camera scheduling approaches. They concluded that evaluating scheduling policies on a physical testbed comprising even a single active camera is extremely problematic. By offering convenient and limitless repeatability, our virtual vision approach provides a vital alternative to physical active camera networks for experimental purposes.

Nevertheless, skeptics may argue that virtual vision relies on simulated data, which can lead to inaccurate results. Fretting that virtual video lacks all the subtleties of real video, some may cling to the dogma that it is impossible to develop a working machine vision system using simulated video. However, our high-level camera control routines do not directly process any raw video. Instead, these routines are realistically driven by data supplied by low-level recognition and tracking routines that mimic the performance of a state-of-the-art pedestrian localization and tracking system, including its limitations and failure modes. This enables us to develop and evaluate camera network control algorithms under realistic simulated conditions consistent with physical camera networks. We believe that the fidelity of our virtual vision emulator is such that algorithms developed through its use will readily port to the real world.[2]

3. Related Work

Preceding virtual vision, a closely related software-based approach to facilitating active vision research was proposed, called *animat vision* [Terzopoulos and Rabie, 1997], which prescribed eschewing the hardware robots that are typically used by computer vision researchers in favor of biomimetic artificial animals (animats) situated in physics-based virtual worlds. Salgian and Ballard describe another early use of virtual reality simulation, which employed synthetic video imagery as seen from the driver's position of a simulated car cruising the streets of a virtual town [Salgian and Ballard, 1998], in order to develop a suite of visual routines running in a real-time image processor to implement an autonomous driving system.

Rabie and Terzopoulos demonstrated their animat vision approach by implementing biomimetic active vision systems for artificial fishes and for virtual humans [Rabie and Terzopoulos, 2000]. Their active vision systems comprised algorithms that integrate motion, stereo, and color

analysis to support robust color object tracking, vision-guided navigation, visual perception, and obstacle recognition and avoidance abilities. Together, these algorithms enabled the artificial animal to sense, understand, and interact with its dynamic virtual environment. The animat vision approach appeared to be particularly useful for modeling and ultimately reverse-engineering the powerful vision systems found in higher-level animals. Furthermore, it obviated the need to grapple with real hardware—cameras, robots, and other paraphernalia—at least during the initial stages of research and development, thereby yielding substantial savings in terms of the cost in money and time to acquire and maintain the hardware. The algorithms developed within the animat vision approach were subsequently adapted for use in a mobile vehicle tracking and traffic control system [Rabie et al., 2002], which affirmed the usefulness of the animate vision approach in designing and evaluating complex computer vision systems.

The virtual vision paradigm for video surveillance systems research was proposed in [Terzopoulos, 2003]. Its central concept was to design and evaluate video surveillance systems using *Reality Emulators*, virtual environments of considerable complexity, inhabited by autonomous, lifelike agents. The work reviewed in this chapter realizes that concept within the reality emulator developed by Shao and Terzopoulos [Shao and Terzopoulos, 2005b; Shao and Terzopoulos, 2005a]—a virtual train station populated with lifelike, self-animating pedestrians.

In concordance with the virtual vision paradigm, Santuari *et al.* [Santuari et al., 2003; Bertamini et al., 2003] advocate the development and evaluation of pedestrian segmentation and tracking algorithms using synthetic video generated within a virtual museum simulator containing scripted characters. Synthetic video is generated via a sophisticated 3D rendering scheme, which supports global illumination, shadows, and visual artifacts like depth of field, motion blur, and interlacing. They have used their virtual museum environment to develop static background modeling, pedestrian segmentation, and pedestrian tracking algorithms. Their work focuses on low-level computer vision.

By contrast, our work has focused on high-level computer vision issues, especially multi-camera control in large-scale camera networks, which is a fundamental high-level problem that must be tackled in order to develop advanced surveillance systems [Qureshi and Terzopoulos, 2006; Qureshi and Terzopoulos, 2008; Qureshi and Terzopoulos, 2009].

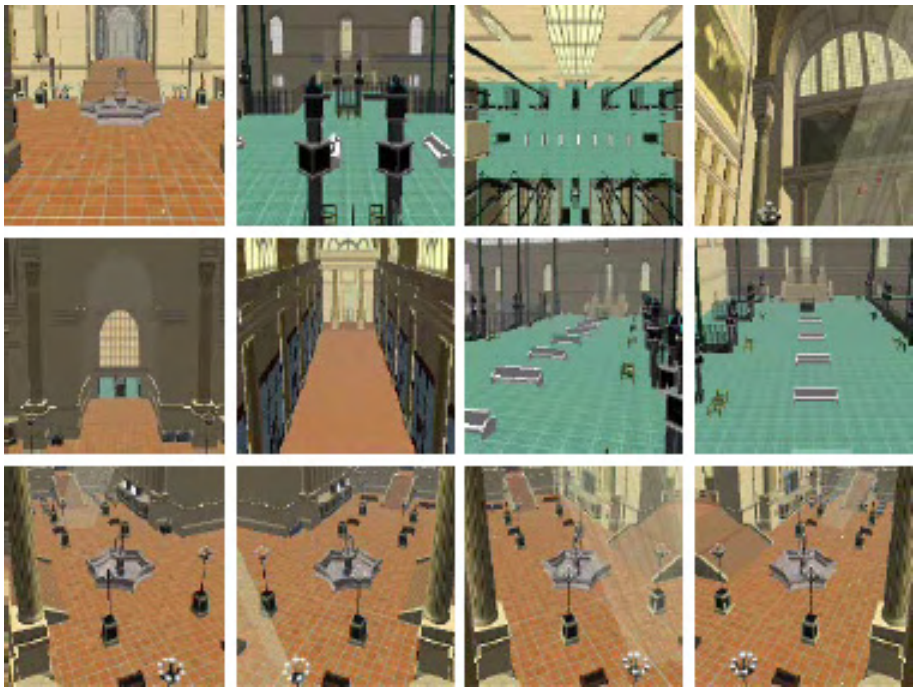


Figure 1.4: Synthetic video feeds from multiple virtual surveillance cameras situated in the (empty) Penn Station environment.

4. Smart Camera Nodes

Each virtual camera node in the sensor network is able to render the scene from its own vantage point in order to generate synthetic video suitable for visual surveillance (Figure 1.4). It is an active sensor that is able to perform low-level visual processing and has a repertoire of autonomous camera behaviors. Furthermore, it is capable of communicating (wirelessly) with nearby nodes in the network. We assume the following communication model: 1) nodes can communicate with their neighbors, 2) messages from one node can be delivered to another node if there is a path between the two nodes, and 3) messages can be broadcast from one node to all the other nodes. Furthermore, we assume the following network model: 1) messages can be delayed, 2) messages can be lost, and 3) nodes can fail. These assumptions ensure that our virtual camera network faithfully mimics the important operational characteristics of a real sensor network.

The following sections describe the capabilities of a camera node in greater detail.

Synthetic Video

Virtual cameras use the OpenGL library and standard graphics pipeline to render the synthetic video feeds. Our imaging model emulates imperfect camera color response, detector and data drop-out noise, compression artifacts, and video interlacing (Figure 1.5). Noise is introduced during a post-rendering phase, and the amount of noise present determines the quality of the input to the visual analysis routines, which affects the performance of the pedestrian segmentation and tracking module.

We model the variation in color response across cameras by manipulating the HSV channels of the rendered image. Similarly, we can adjust the tints, tones, and shades of an image by adding the desired amounts of blacks, whites, and grays, respectively [Birren, 1976]. Our visual analysis routines rely on color-based appearance models to track pedestrians; hence, camera handovers are sensitive to variations in the color responses of the different cameras.

Bandwidth is usually at a premium in sensor networks, especially so in camera networks. To keep bandwidth requirements within acceptable limits, camera nodes typically compress the captured video frames before transmitting them to the monitoring station or to other nodes for the purposes of camera coordination, camera handover, and multi-camera sensing operations. Compression artifacts and the limited resolution of the captured video pose a challenge to visual analysis routines and are therefore relevant to camera network research. To enhance realism, we introduce compression artifacts into the synthetic video by subjecting it to JPEG compression and decompression before providing it to the pedestrian recognition and tracking module.

Visual Processing

The sensing capabilities of a camera node are determined by low-level visual routines (LVR). The LVRs, which implement basic functionalities such as pedestrian detection, tracking, and identification, are computer vision algorithms that directly operate upon the synthetic video generated by the virtual cameras and mimic the performance and limitations of a state-of-the-art surveillance video analysis module. Our virtual vision simulator affords us the benefit of fine tuning the performance of this module by taking into consideration the ground truth data readily available in the virtual world.

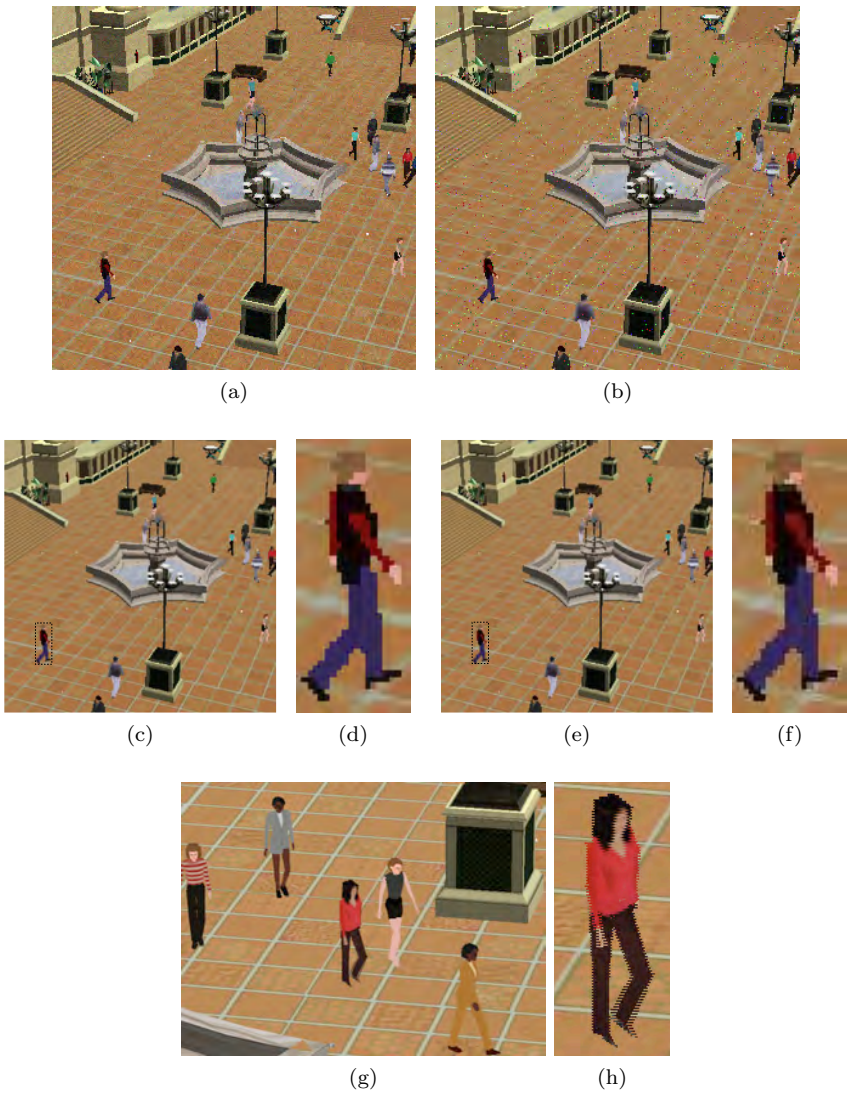


Figure 1.5: Simulating noise in synthetic video: (a) Detector noise. (b) Data drop-out noise. Compression artifacts in synthetic video: (c) Uncompressed image. (d) Enlarged region of the rectangular box in (c). (e) JPEG-compressed image. (f) Enlarged region of the rectangular box in (e). Video interlacing effects: (g) Video frame obtained by interlacing two consecutive fields. (h) Close-up view of a pedestrian in (g).

(a) (b) (c)

Figure 1.6: 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).

We have employed appearance-based models to track pedestrians. Pedestrians are segmented to compute robust color-based signatures, which are then matched across subsequent frames. Color-based signatures have found widespread use in tracking applications [Comaniciu et al., 2000], but they are sensitive to illumination changes. This shortcoming can be mitigated, however, by operating in HSV rather than RGB color space. Furthermore, zooming can drastically alter the appearance of a pedestrian, thereby confounding conventional appearance-based schemes. We employ a modified color-indexing scheme [Swain and Ballard, 1991] to tackle this problem. Thus, a distinctive characteristic of our pedestrian tracking routine is its ability to operate over a range of camera zoom settings. Note that we do not assume that the active cameras are calibrated.

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 poor illumination (Fig. 1.6). 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.

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

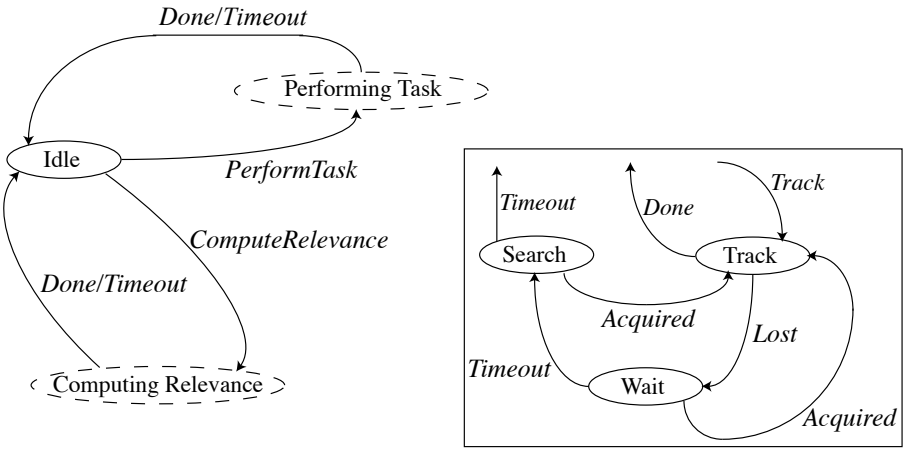


Figure 1.7: The top-level camera controller consists of a hierarchical finite state machine (FSM). The inset (right) represents the child FSM embedded within the *PerformingTask* and *ComputingRelevance* states in the top-level FSM.

The *fixate* routine brings the region of interest—e.g., the bounding box of a pedestrian—into the center of the image by rotating the camera about its local x and y axes. The *zoom* routine controls the FOV of the camera such that the region of interest occupies the desired percentage of the image.

The implementation details of the various LVRs are presented elsewhere [Qureshi, 2007].

Camera Node Behavioral Controller

The camera controller determines the overall behavior of the camera node, taking into account the information gathered through visual analysis by the LVRs (bottom-up) and the current task (top-down). We model the camera controller as an augmented hierarchical finite state machine (Fig. 1.7).

In its default state, *Idle*, the camera node is not involved in any task. It transitions into the *ComputingRelevance* state upon receiving a *queryrelevance* message from a nearby node. Using the description of the task that is contained within the *queryrelevance* message, and by employing the LVRs, the camera node can compute its *relevance* to the task [Qureshi and Terzopoulos, 2009]. For example, it can use visual search to find a pedestrian that matches the appearance-based signature forwarded by the querying node. The relevance encodes the expectation

of how successful a camera node will be at a particular sensing task. The camera node returns to the *Idle* state if it fails to compute its relevance because it cannot find a pedestrian matching the description. Otherwise, when the camera successfully finds the desired pedestrian, it returns its relevance value to the querying node. The querying node passes the relevance value to the supervisor node of the group, which decides whether or not to include the camera node in the group. The camera goes into the *PerformingTask* state upon joining a group, where the embedded child finite state machine hides the sensing details from the top-level controller and enables the node to handle transient sensing (tracking) failures. All states other than the *PerformingTask* state have built-in timers (not shown in Fig. 1.7) that allow the camera node to transition into the *Idle* state rather than wait indefinitely for a message from another node.

The child FSM (Fig. 1.7 (inset)) starts in *Track* state, where video frames are processed to track a target without panning and zooming a camera. *Wait* is entered when track is lost. Here camera zoom is gradually reduced in order to reacquire track. If a target is not reacquired during *Wait*, the camera transitions to the *Search* state, where it performs search sweeps in PTZ space to reacquire the target.

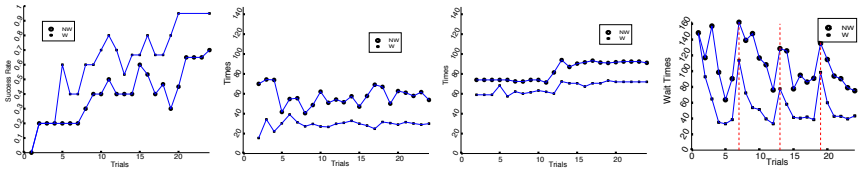
A camera node returns to its default state after finishing a task, using the *reset* routine, which is a proportional-derivative (PD) controller that attempts to minimize the difference between the current zoom/tilt settings and the default zoom/tilt settings.

5. Surveillance Systems

To date, we have studied the problems of active camera scheduling and collaborative, persistent observation within smart camera networks. We have been able to rapidly develop novel camera control strategies to address these problems by deploying virtual camera networks in the virtual Penn Station's large-scale simulated indoor urban environment.

Active Camera Scheduling

In 2005, we introduced a camera scheduling strategy for intelligently managing multiple, uncalibrated active PTZ cameras, supported by several static, calibrated cameras in order to satisfy the challenging task of automatically recording close-up biometric videos of pedestrians present in a scene. Our approach assumes a non-clairvoyant model of the scene, supports multiple cameras, supports preemption, and allows multiple observations of the same pedestrian [Qureshi and Terzopoulos, 2006].



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

Figure 1.8: Comparisons of Weighted (W) 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.

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.

We formulated the multi-camera control strategy as an online scheduling problem and proposed a solution that combines the information gathered by the wide-FOV cameras with weighted round-robin scheduling to guide the available PTZ cameras, such that each pedestrian is observed by at least one PTZ camera while in the designated area. Fig. 1.8 compares weighted and non-weighted scheduling schemes for active PTZ cameras assignment.

Collaborative Persistent Surveillance

In [Qureshi and Terzopoulos, 2008], we developed a distributed coalition formation strategy for collaborative sensing tasks in camera sensor networks. The proposed model supports task-dependent node selection and aggregation through an announcement/bidding/selection strategy combined with a constraint satisfaction problem (CSP) based conflict

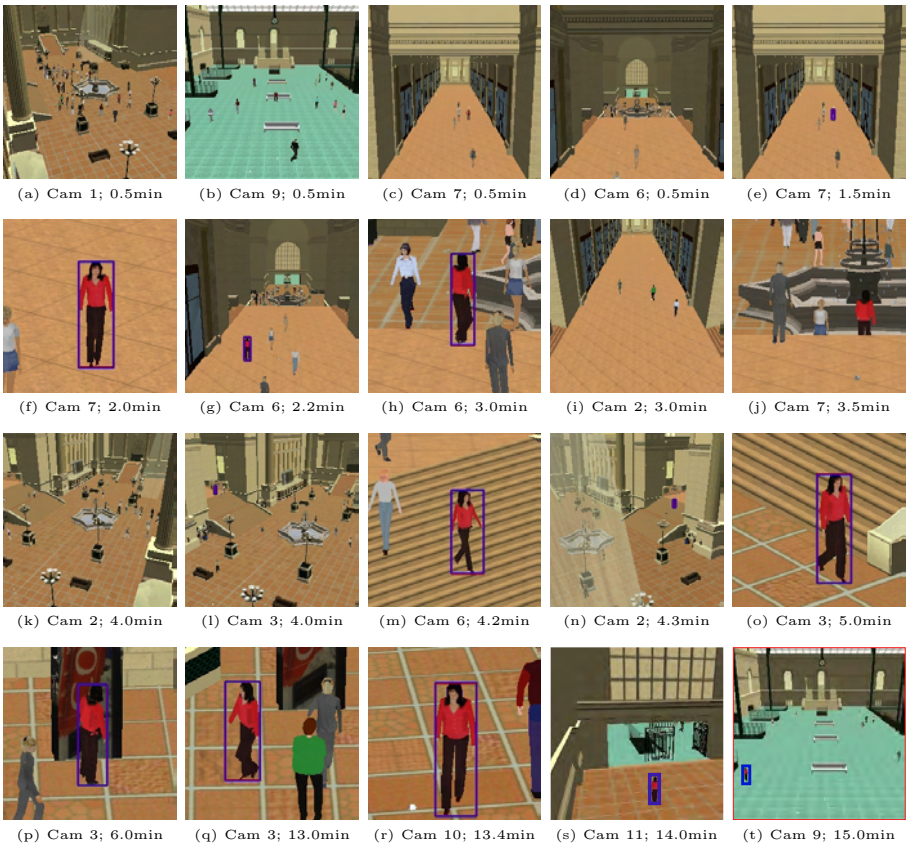
resolution mechanism. Our technique is scalable as it lacks any central controller, and it is robust to node failures and imperfect communication. In response to a sensing task, such as, "observe pedestrian i during his stay in the region of interest," wide-FOV passive and PTZ active cameras organize themselves into groups with the objective of fulfilling the task. These groups evolve as the pedestrian enters and exits the fields of view of different cameras, ensuring that the pedestrian remains persistently under surveillance by at least one camera. Fig. 1.9 illustrates the 15-minute persistent observation of a pedestrian of interest as she makes her way through the train station. For this example, we placed 16 active PTZ cameras in the train station, as shown in Fig. 1.2.

6. Conclusions

Virtual Vision is a unique synthesis of virtual reality, artificial life, computer graphics, computer vision, and sensor network technologies, with the purpose of facilitating computer vision research for camera sensor networks. Through the faithful emulation of physical vision systems, any researcher can investigate, develop, and evaluate camera sensor network algorithms and systems in virtual worlds simulated on high-end commodity personal computers. Without having to deal with special-purpose surveillance hardware, we have demonstrated our prototype surveillance systems in a virtual train station environment populated by lifelike, autonomous pedestrians. This simulator has facilitated our ability to design visual sensor networks and experiment with them on commodity personal computers.

In this chapter we described two prototype multi-camera 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 the multiple active cameras in order to capture close-up videos of pedestrians as they move through designated areas. The second surveillance system managed multiple uncalibrated passive and active cameras intelligently in order to persistently observe pedestrians of interest that enter and exit the FOVs of different cameras as they travel across the train station. Our companion chapter in this volume reviews our most recent work on proactive planning for PTZ camera assignment and handoff [Qureshi and Terzopoulos, 2009].

The future of advanced simulation-based approaches for the purposes of low-cost prototyping and facile experimentation appears promising. Imagine an entire city, including indoor and outdoor environments, subway stations, automobiles, shops and market places, homes and public



(a) Cam 1; 0.5min (b) Cam 9; 0.5min (c) Cam 7; 0.5min (d) Cam 6; 0.5min (e) Cam 7; 1.5min

(f) Cam 7; 2.0min (g) Cam 6; 2.2min (h) Cam 6; 3.0min (i) Cam 2; 3.0min (j) Cam 7; 3.5min

(k) Cam 2; 4.0min (l) Cam 3; 4.0min (m) Cam 6; 4.2min (n) Cam 2; 4.3min (o) Cam 3; 5.0min

(p) Cam 3; 6.0min (q) Cam 3; 13.0min (r) Cam 10; 13.4min (s) Cam 11; 14.0min (t) Cam 9; 15.0min

Figure 1.9: 15-minute persistent observation of a pedestrian of interest as she makes her way through the train station. (a-d) Cameras 1, 9, 7, and 8 monitoring the station. (e) The operator selects a pedestrian of interest in the video feed from Camera 7. (f) Camera 7 has zoomed in on the pedestrian, (g) Camera 6, which is recruited by Camera 7, acquires the pedestrian. (h) Camera 6 zooms in on the pedestrian. (i) Camera 2. (j) Camera 7 reverts to its default mode after losing track of the pedestrian and is now ready for another task. (k) Camera 2, which is recruited by Camera 6, acquires the pedestrian. (l) Camera 3 is recruited by Camera 6; Camera 3 has acquired the pedestrian. (m) Camera 6 has lost track of the pedestrian. (n) Camera 2 observing the pedestrian. (o) Camera 3 zooming in on the pedestrian. (p) Pedestrian is at the vending machine. (q) Pedestrian is walking towards the concourse. (r) Camera 10 is recruited by Camera 3; Camera 10 is observing the pedestrian. (s) Camera 11 is recruited by Camera 10. (t) Camera 9 is recruited by Camera 10.

spaces, all richly inhabited by autonomous virtual humans. Such large-scale virtual worlds will one day provide unprecedented opportunities for studying large-scale camera sensor networks in ways not currently possible in our train station simulator. Future work on virtual vision research will therefore benefit from long-term efforts to increase the complexity of virtual worlds.

Acknowledgments

We thank Wei Shao for developing and implementing the train station simulator and Mauricio Plaza-Villegas for his valuable contributions. We thank Tom Strat, formerly of DARPA, for his generous support and encouragement.

Notes

1. With regard to software, a virtual vision simulator consists of an environmental model, character models, an animation engine, and a rendering engine. Most commercial modeling/animation systems enable users to create 3D virtual scenes, including virtual buildings populated by virtual characters, and they incorporate rendering subsystems to illuminate and visualize the scenes. The animation subsystem can animate the virtual characters, but autonomous pedestrian animation is an area of active research in the computer animation community and there are as yet no adequate commercial solutions.

2. We are currently validating our virtual vision paradigm in a collaborative project with the University of California, Riverside, through the development of a virtual vision simulator that emulates a large-scale physical camera network that they have deployed.

References

Bertamini, F., Brunelli, R., Lanz, O., Roat, A., Santuari, A., Tobia, F., and Xu, Q. (2003). Olympus: An ambient intelligence architecture on the verge of reality. In *Proc. International Conference on Image Analysis and Processing*, pages 139–145, Mantova, Italy.

Birren, F. (1976). *Color Perception in Art.* Van Nostrand Reinhold, New York.

Comaniciu, D., Ramesh, V., and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, volume 2, pages 142–151, Hilton Head Island, SC.

Costello, C. J., Diehl, C. P., Banerjee, A., and Fisher, H. (2004). "Scheduling an active camera to observe people," in *Proc. ACM Int. Workshop on Video Surveillance and Sensor Networks*, pages. 39–45, New York.

Qureshi, F. Z. (2007). *Intelligent Perception in Virtual Camera Networks and Space Robotics.* PhD thesis, Department of Computer Science, University of Toronto, Canada.

Qureshi, F. Z. and Terzopoulos, D. (2005). Towards intelligent camera networks: A virtual vision approach. In *Proc. Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS05)*, pages 177–184, Beijing, China.

Qureshi, F. Z. and Terzopoulos, D. (2006). Surveillance camera scheduling: A virtual vision approach. *ACM Multimedia Systems Journal*, 12:269–283.

Qureshi, F. Z. and Terzopoulos, D. (2008). Smart camera networks in virtual reality. *Proceedings of the IEEE (Special Issue on Smart Cameras)*, 96(10):1640–1656.

Qureshi, Faisal Z. and Terzopoulos, Demetri (2009). Planning ahead for PTZ camera assignment and control. In *Proc. Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 09)*, pages 1–8, Como, Italy.

Rabie, T., Shalaby, A., Abdulhai, B., and El-Rabbany, A. (2002). Mobile vision-based vehicle tracking and traffic control. In *Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC 2002)*, pages 13–18, Singapore.

Rabie, T. and Terzopoulos, D. (2000). Active perception in virtual humans. In *Vision Interface (VI 2000)*, pages 16–22, Montreal, Canada.

Salgian, G. and Ballard, D. H. (1991). Visual routines for autonomous driving. *Sixth International Conference on Computer Vision*, pages 876–882, Bombay, India.

Santuari, A., Lanz, O., and Brunelli, R. (2003). Synthetic movies for computer vision applications. In *Proc. IASTED International Conference: Visualization, Imaging, and Image Processing*, pages 1–6, Spain.

Shao, W. and Terzopoulos, D. (2005a). Autonomous pedestrians. In *Proc. ACM SIGGRAPH/EG Symposium on Computer Animation*, pages 19–28, Los Angeles, CA.

Shao, W. and Terzopoulos, D. (2005b). Environmental modeling for autonomous virtual pedestrians. In *Proc. SAE Digital Human Modeling Symposium*, Iowa City, IA.

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7(1):11–32.

Terzopoulos, D. (2003). Perceptive agents and systems in virtual reality. In *Proc. ACM Symposium on Virtual Reality Software and Technology*, pages 1–3, Osaka, Japan.

Terzopoulos, D. and Rabie, T. (1997). Animat vision: Active vision in artificial animals. *Videre: Journal of Computer Vision Research*, 1(1):2–19.

Index