CHAPTER

Virtual Vision for Camera Networks Research

21

Faisal Z. Qureshi[*] **and Demetri Terzopoulos**[†]

[*] *University of Ontario Institute of Technology, Canada*
[†] *University of California, Los Angeles, USA*

4.21.1 Introduction

Multi-camera systems are rapidly evolving from highly specialized wired networks of stationary passive and active cameras that provide visual coverage of the scene to *ad hoc* wireless networks of smart camera nodes, capable of near-autonomous operation in a variety of applications, such as urban and participatory sensing, disaster response, plant and animal habitat monitoring, etc. Whereas traditional multi-camera systems focus primarily on wide-area scene analysis, smart camera networks are also concerned with camera coordination and control, in-network processing and storage, and resource-aware visual analysis. Pre-recorded video, while useful, is inadequate in the study of camera control and coordination strategies. Rather, one needs online access to the entire network in order to control and study its behavior under different sensing regimes. This observation, together with the fact that most researchers who are motivated to study camera networks do not have access to physical camera networks of suitable complexity, led us to propose the "Virtual Vision" paradigm for camera networks research (see Figure 21.1).

4.21.1.1 Virtual vision

Virtual vision advocates employing visually and behaviorally realistic 3D virtual environments, populated with lifelike, self-animating objects (pedestrians, automobiles, etc.), to carry out camera networks research. Camera networks are simulated in these environments by deploying virtual cameras that mimic the characteristics of physical cameras. Virtual vision offers several advantages over the use of physical camera networks during the ideation, prototyping, and evaluation phases of camera networks research, among them:

- The virtual vision simulator runs on (high-end) commodity PCs, obviating the need to grapple with special-purpose hardware.
- The virtual cameras are very easily instantiated, relocated, and reconfigured in the virtual environment.
- The virtual world provides readily accessible ground-truth data for the purposes of algorithm/system validation.
- Experiments are perfectly repeatable in the virtual world, so we can easily modify algorithms and/or their parameters and immediately determine the effect.

Academic Press Library in Signal Processing. http://dx.doi.org/10.1016/B978-0-12-396501-1.00021-2

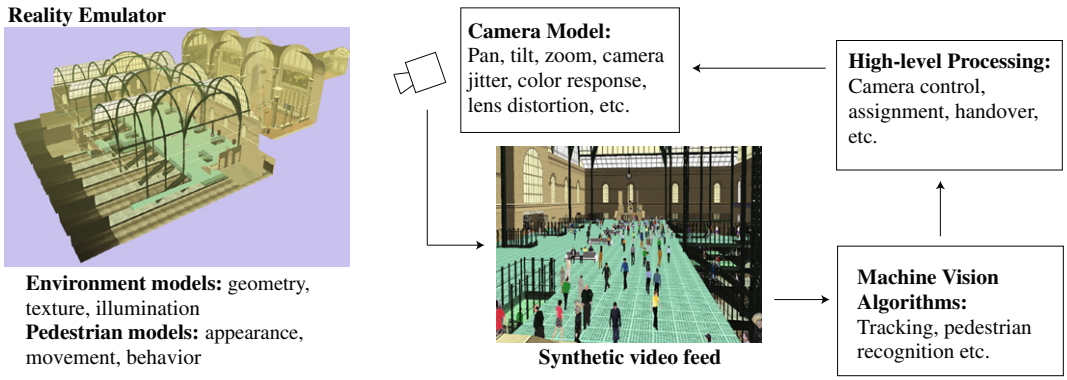


FIGURE 21.1

The *virtual vision* paradigm.

- Our simulated camera networks run in "real time" within the virtual world, with the virtual cameras actively controlled online by the vision algorithms. By 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.

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 view (FOV) 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 a virtual train station and a virtual office space. Our simulated networks exhibit performance characteristics similar to those of physical camera networks; e.g., video compression artifacts, latency, limited bandwidth, communication errors, camera node failures, etc.

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. [1], 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 useful 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.[1]

4.21.1.2 Overview

The remainder of this article is organized as follows: We review related work in the next section. Section 4.21.3 reviews the two virtual environments that we have employed in our work—a 3D reconstruction of a train station and a 3D model of a floor of an office tower. We then describe in Section 4.21.4 the camera networks that we have developed using these virtual vision simulators. Section 4.21.5 concludes this article with a summary.

4.21.2 Related work

Preceding virtual vision, a closely related software-based approach to facilitating active vision research was proposed, called *animat vision* [2], 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 [3], 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 [4]. 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 the powerful vision systems found in animals. Furthermore, it obviated the need to grapple with physical hardware—cameras, robots, and other paraphernalia—at least during the initial stages of research and development, thereby yielding substantial savings 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 [5], 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 [6]. 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 workreviewed

[1] 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 an existing large-scale physical camera network.

in this chapter realizes that concept within the reality emulator developed by Shao and Terzopoulos [7,8]—a virtual train station populated with autonomous pedestrians—and by Starzyk et al. [9]—a 3D office floor inhabited by scripted pedestrians.

In concordance with the virtual vision paradigm, Santuari et al. [10,11] 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 such as 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 problem that must be tackled in order to develop advanced surveillance systems [12–14].

In recent years there has been much interest in distributed algorithms for tracking in camera networks comprising passive and active cameras. Sankaranarayanan et al. tackle object detection, tracking and recognition in multi-camera systems [15]. They exploit real-world constraints, such as the presence of a three-dimensional scene model, consistency of color and texture, etc., for their purposes. Ding et al. develop a distributed optimization strategy to select the pan, tilt, and zoom settings of multiple active cameras in order to maximize various scene understanding performance criteria [16]. Song et al. [17] studied the problem of tracking and activity recognition in distributed camera networks. Refs. [16,17] are noteworthy; they evaluate the performance of the sensing strategy using a simulation environment, as well as on a small-scale physical camera network. The simulation environment used is much simpler then the virtual vision simulator that we have developed, as it does not support the full vision pipeline, from image acquisition to tracking.

Virtual vision simulators can play a key role in studying collaborative sensing strategies in PTZ camera networks. For example, a virtual vision simulator can be readily used to compare different sensing (control) strategies in PTZ camera networks. The only way to compare two sensing (control) strategies on a physical PTZ camera network is through scene re-enactment, since pre-recorded videos cannot be used for comparing sensing (control) algorithms. Scene re-enactments clearly have a very high human cost.

4.21.3 Virtual vision simulators

In this section we describe the 3D environments that we have used in our work. Virtual cameras situated in these environments capture synthetic video footage, which is then passed on to video analysis routines, such as background subtraction, blob detection, and pedestrian tracking. High-level camera control and coordination algorithms rely upon these routines when deciding how best to control a camera network in order to achieve one or more observation goals.

4.21.3.1 A train station

Our first virtual vision simulator was developed using an advanced pedestrian animation system that combines behavioral, perceptual, and cognitive human simulation algorithms [7]. The simulator

FIGURE 21.2

A large-scale virtual train station populated by self-animating virtual humans [7]. (a) Waiting Room. (b) Concourses. (c) Arcade. (d) Platforms.

reconstructs the original Penn Station in New York City, which was demolished in 1963 to make way for the current Penn Station and Madison Square Garden complex (Figure 21.2). The station simulation 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 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 21.2.

4.21.3.2 A floor of an office tower

More recently, we have developed a distributed virtual vision simulator that depicts a floor in a typical office tower in downtown Toronto [9]. This simulator can animate and render up to 100 pedestrians at 15 frames per second (fps). Here, pedestrians follow scripted paths as they move around in their workplace. The simulator features advanced lighting effects, such as sunlight filtering through large glass windows, static and dynamic objects casting shadows on one another, etc. More importantly, however, this simulator is highly scalable and can support much larger camera networks than is possible with the Penn Station virtual vision simulator. In some recent tests we have been able to simulate networks of 100 + cameras. Figure 21.3 shows images captured by virtual cameras installed on the simulated office floor.

The simulator is based upon the open-source game engine, Panda3D [18]. Panda3D is a 3D rendering framework, into which programmers insert 3D models. These 3D models can be inanimate, such as buildings, furniture, etc., or animated, such as people and vehicles. Many packages are available to create 3D models that can be imported into Panda3D. Additionally pre-made 3D models, both inanimate and animate, can be purchased from multiple vendors who specialize in creating digital assets for

FIGURE 21.3

A view of an upper floor of a virtual office building [9]. The city skyline is visible through floor-to-ceiling panoramic windows. Our scripted pedestrians use motion-capture data to simulate realistic motion and they cast dynamic shadows on their environment.

computer games and movies. Our virtual vision simulator contains the control, sensing, and communication routines needed to simulate active and passive camera networks. It also contains the algorithms needed to animate 3D pedestrian models to simulate human traffic in 3D environments. It is relatively easy to customize our virtual vision simulator to simulate other environments. What is needed is (1) an appropriate 3D model of the environment and (2) motion scripts for the pedestrians, so that the pedestrians do not collide with walls and each other.

4.21.3.3 Simulated cameras

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. It is an active sensor that is able to perform low-level visual processing and it 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 visual sensor network. Our imaging model can emulate various camera sensing artifacts, such as video interlacing, camera color response, camera jitter, compression artifacts, and sensor resolution.

4.21.3.4 Visual analysis

We have developed tracking routines that are able to detect, identify and track pedestrians in the synthetic video feed generated by virtual cameras deployed in our 3D environments. The pedestrian tracker faithfully mimics the characteristics of a state-of-the-art pedestrian tracker that one might use to track people in video footage captured by physical cameras. Our tracker, for example, will momentarily fail due to occlusions, poor illumination, in crowded conditions, or when multiple individuals have a similar appearance. Our virtual vision simulator, however, 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. Thus far, we have employed appearance-based models to track pedestrians (Figure 21.4). Pedestrians

FIGURE 21.4

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).

are either segmented automatically (against a static background) or manually identified by the operator to construct an appearance model. Appearance models are then matched across the successive frames to track pedestrians. 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.

We have recently been developing a visual analysis pipeline that contains computer vision routines typically employed in surveillance systems; e.g., automatic camera calibration, pedestrian detection, tracking, and re-identification, face detection and identification, and head detection [9]. Figure 21.5a depicts the visual analysis pipeline responsible for pedestrian detection, segmentation and tracking in video feeds captured by passive wide FOV cameras. The passive visual analysis pipeline is able to track multiple pedestrians without any user input. The visual analysis pipeline for active PTZ cameras is depicted in Figure 21.5b. The PTZ visual analysis pipeline can also track multiple pedestrians; however, the appearance signatures for those pedestrians must be provided by another camera in the vicinity or manually by a human operator who can identify a person of interest by drawing a stroke over the camera image of that individual (Figure 21.7).

It is noteworthy that these visual analysis pipelines have been evaluated on real video footage recorded by physical cameras (Figure 21.6). This should assuage concerns that the synthetic imagery may lack the subtlety and richness of real footage, and should enable us to port the camera network software implemented within our virtual vision simulators to physical camera networks in the real world.

4.21.3.5 Image-driven PTZ zoom and fixation

A PTZ 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. The *fixate* routine brings the region of interest—e.g., the bounding box of a pedestrian—into

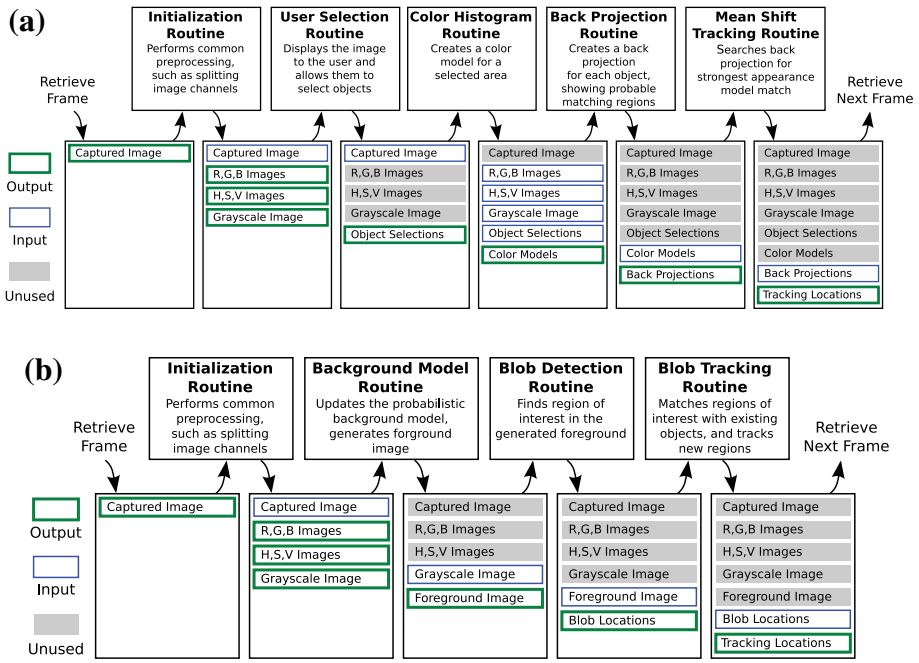


FIGURE 21.5

Visual analysis pipelines are realized as a collection of reusable vision routines. (a) Visual analysis pipeline for tracking pedestrians in PTZ cameras. (b) Visual analysis pipeline for tracking pedestrians in wide-FOV cameras.

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.

4.21.3.6 Behavior-based camera nodes

The camera controller determines the overall behavior of the camera node, taking into account the information gathered through visual analysis by the vision routines (bottom-up) and the current task (top-down). We model the camera controller as an augmented hierarchical finite state machine (Figure 21.8).

In its default state, *Idle*, the camera node is not involved in any task. It transitions into the *Computing-Relevance* 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 its visual analysis routines, the camera node can compute its *relevance* to the task [14]. 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

FIGURE 21.6

Our visual analysis pipeline is designed from the ground up to work with both synthetic (right) and real video (left) without any modification. Consequently, our vision pipeline faithfully mimics the performance of a vision pipeline implemented on physical cameras.

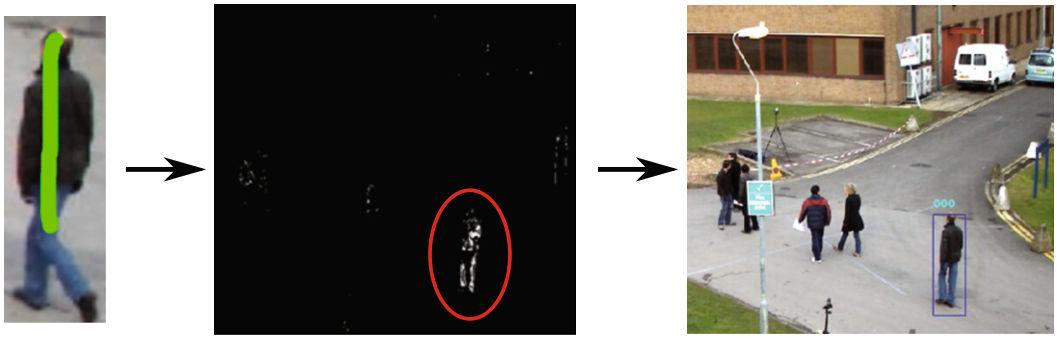


FIGURE 21.7

A stroke gesture is provided to select a pedestrian to be tracked in active PTZ cameras. Appearance signatures computed by passive wide-FOV cameras can also be used to track individuals in active PTZ cameras.

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

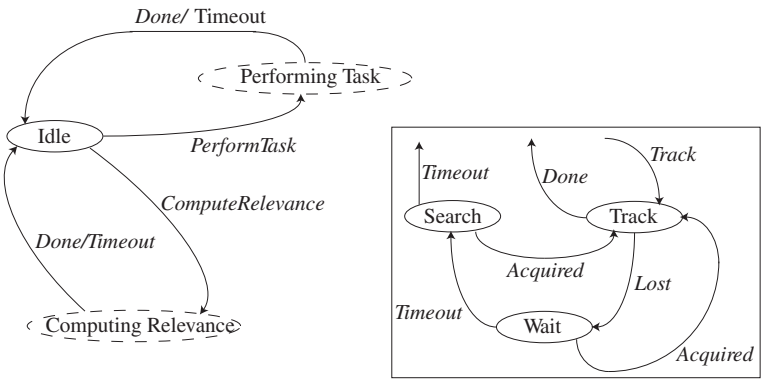


FIGURE 21.8

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.

timers (not shown in Figure 21.8) that allow the camera node to transition into the *Idle* state rather than wait indefinitely for a message from another node.

The child FSM (Figure 21.8 (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.

4.21.4 Prototype camera networks

We have used our virtual vision simulators to study the problems of active camera scheduling, collaborative sensing in *ad hoc* networks of smart active sensors, proactive PTZ camera control, and multi-tasking PTZ cameras. We have been able to rapidly develop novel camera control strategies to address these problems by deploying virtual camera networks in our simulated indoor urban environments.

4.21.4.1 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 [12]. Our approach assumes a non-clairvoyant model of the scene, supports multiple cameras, supports preemption, and allows multiple observations of the same pedestrian.

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

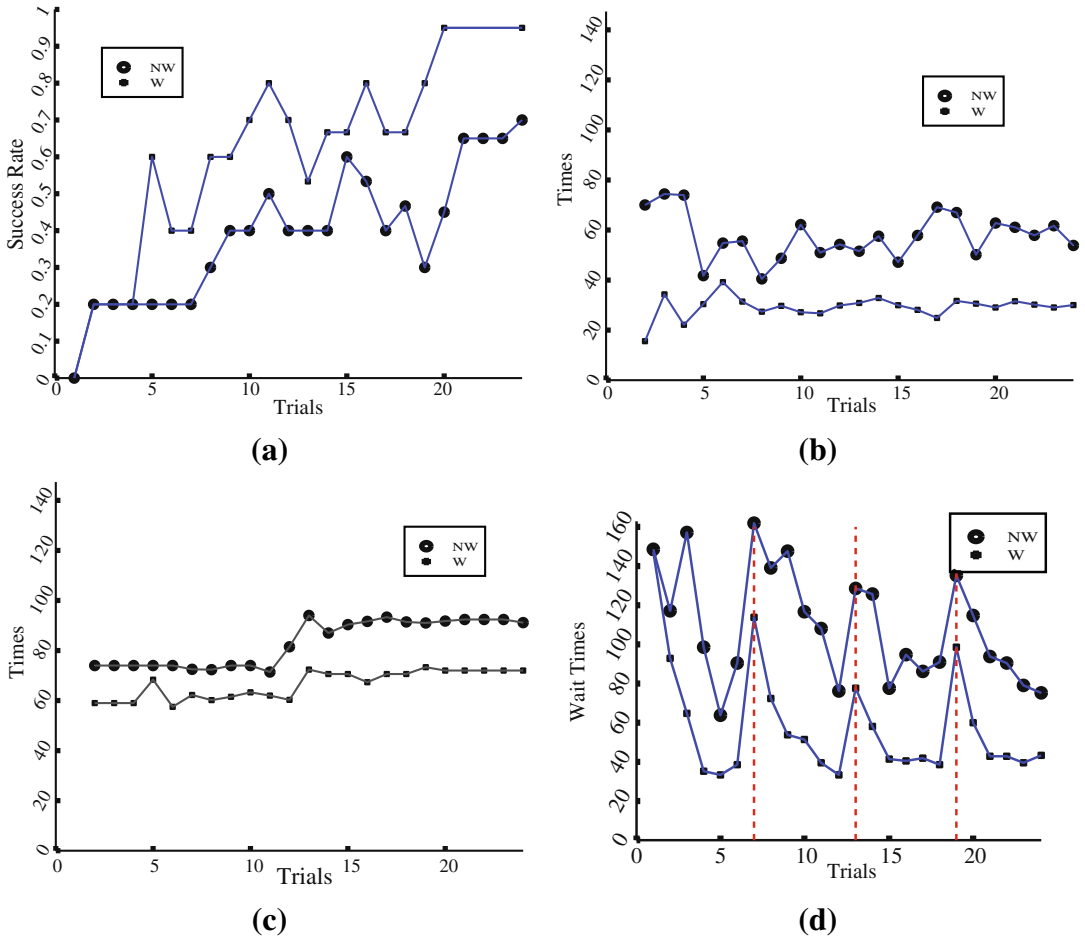


FIGURE 21.9

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.

affixed a fish-eye lens 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. Figure 21.9 compares weighted and non-weighted scheduling schemes for active PTZ camera assignment.

4.21.4.2 Collaborative persistent surveillance

In [13], 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 their presence 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. Figure 21.10 illustrates the 15-min 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 Figure 21.2.

4.21.4.3 Proactive PTZ control

PTZ camera networks that are able to anticipate future sensing requirements do a better job of managing sensing resources, avoiding assignments that might lead to sensing failures in the future. In [14] we developed a cognitive PTZ camera network that is able to plan ahead when determining camera assignments. The reasoning process—which considers both the immediate and the future consequences of different camera assignments when constructing a "plan" that is most likely to succeed (in a probabilistic sense) at the current observation task(s)—is capable of performing camera assignments and handoffs in order to provide persistent coverage of a region. In [19], we extended the reasoning engine with the ability to generalize and store the results of the reasoning process (Figure 21.11). Whenever the camera network encounters a previously unseen situation, it invokes the planning process to construct optimal camera assignment. The results of this process are stored as rules in a production system. Later, when the network again encounters a similar situation, it bypasses the reasoning process and uses the stored rules to perform camera assignments. Initially, the camera network relies mostly on the reasoning process; over time, however, camera assignments become instinctive.

4.21.4.4 Multi-tasking PTZ cameras

Reference [20] develops a behavior-based PTZ camera controller that automatically tunes the sensing parameters of PTZ cameras in response to the scene activity, choosing to capture close-up video when

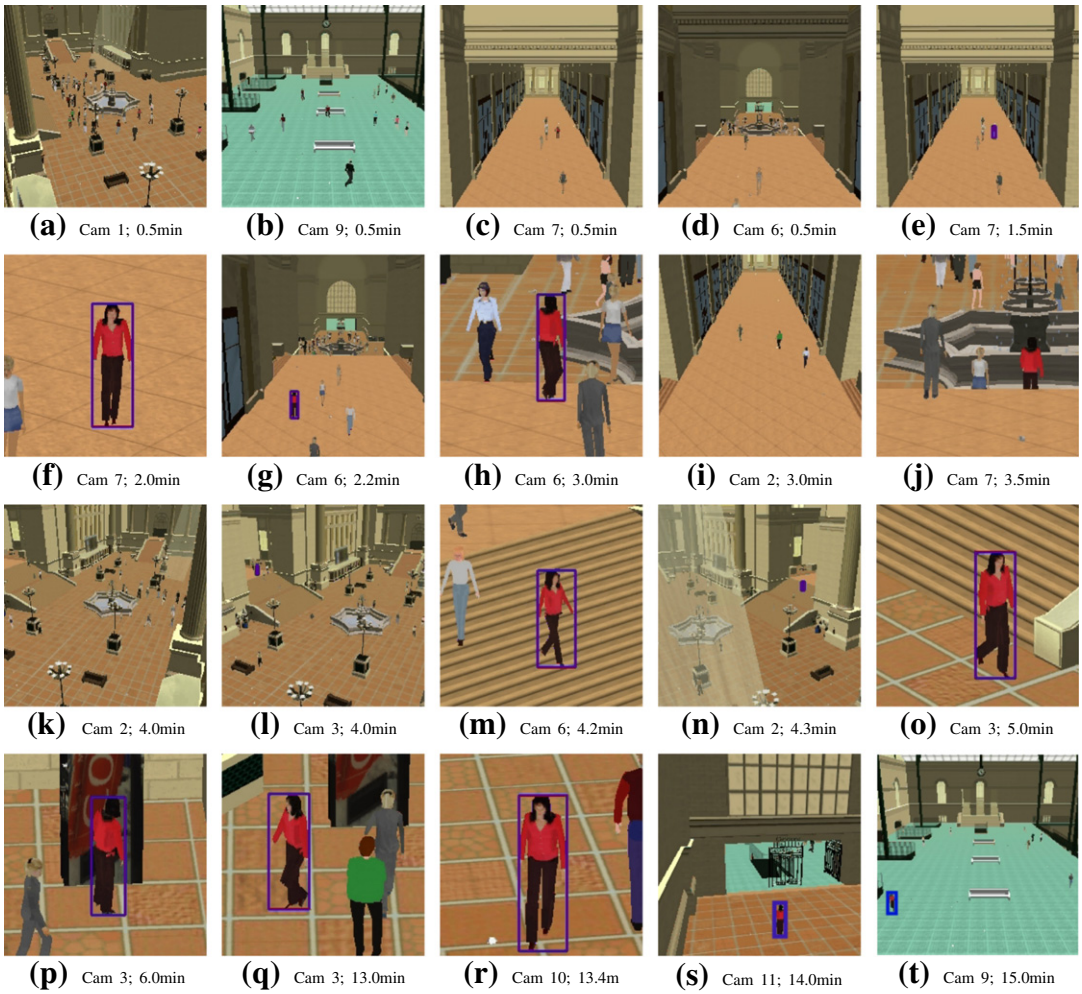


FIGURE 21.10

Fifteen-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.

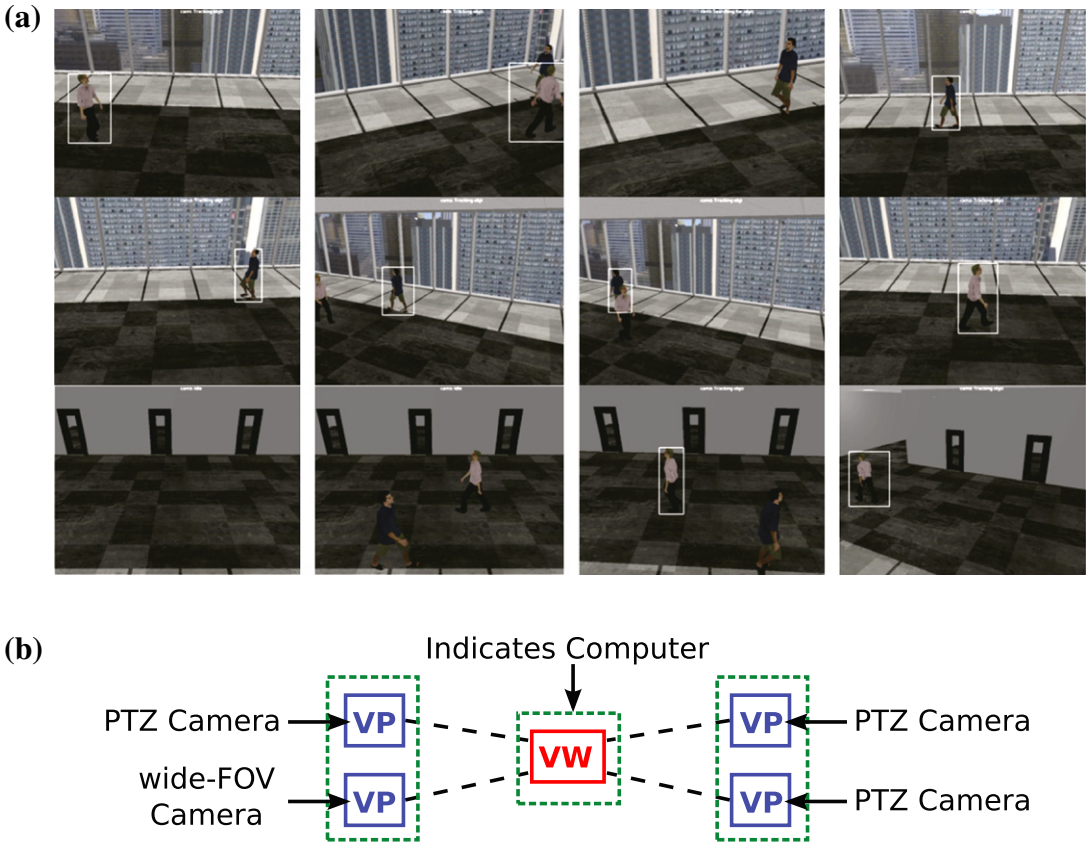
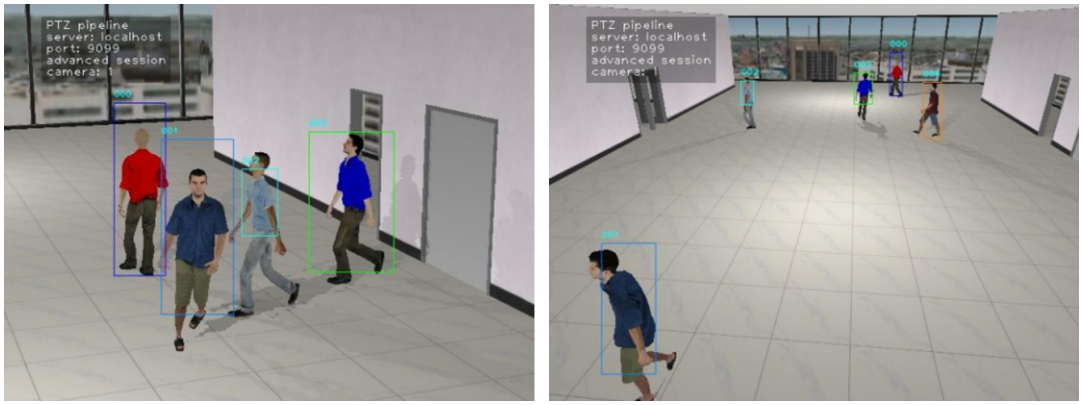


FIGURE 21.11

(a) The three rows show three cameras observing two pedestrians as they cross each other on their way to opposite sides of the lobby. The three cameras are able to perform handoff while keeping both pedestrians in view. This is achieved through a reasoning mechanism that considers both the short-term and long-term consequences of camera assignments. (b) The virtual vision simulator comprised one VW and four VP modules spread over three computers.

the number of pedestrians present in the scene is low and electing to capture lower-resolution video as the number of pedestrians increases, thus continually keeping every pedestrian in view. These cameras enable the video surveillance system to intelligently respond to scene complexity, automatically capturing close-up imagery of the pedestrians present in the scene when possible, and behaving as wide-FOV cameras when the number of pedestrians increases. Figure 21.12 shows a multi-tasking PTZ camera: The PTZ camera is able to capture higher resolution video of the pedestrians in the scene when there are



(a) (b)

FIGURE 21.12

PTZ cameras automatically decide how best to observe a scene. (a) When possible, the PTZ camera selects a higher zoom to capture higher resolution images of the individuals present in the scene. (b) As the individuals spread out and move away from the camera, the PTZ camera selects a lower zoom setting to keep all of them in view, albeit at a lower resolution.

only a few pedestrians present; however, it begins to behave like a wide-FOV camera as the pedestrians present in the scene spread out and move away from the camera.

4.21.5 Conclusion

Virtual vision is a unique synthesis of virtual reality, artificial life, computer graphics, computer vision, and sensor network technologies, with the objective of facilitating computer vision research applied to human surveillance using 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, without having to deal with special-purpose surveillance hardware. We have demonstrated our prototype surveillance systems in two simulated virtual environments populated by self-animating pedestrians, which have facilitated our ability to design visual sensor networks and experiment with them on commodity personal computers.

The future of advanced simulation-based approaches for the purposes of low-cost prototyping and facile experimentation appears promising and our virtual vision approach will continue to benefit from long-term efforts to increase the complexity of virtual worlds. Imagine an entire city, including indoor and outdoor environments, subway stations, automobiles, shops and market places, homes and public spaces, all richly inhabited by autonomous virtual humans. Such city-scale virtual worlds will one day provide unprecedented opportunities to develop and assess large-scale camera sensor networks in ways not yet possible with our current simulators.

Glossary

Smart camera	a self-contained vision system, which includes an image sensor, on-board processing and storage capabilities, power, and (often wireless) communication interfaces
Camera network	a network of camera nodes; unlike traditional multi-camera systems, smart camera networks typically comprise smart camera nodes
Virtual vision	the use of visually and behaviorally realistic three-dimensional (3D) environments to carry out camera networks research
Reality emulator	a virtual world richly populated with lifelike, self-animating agents approaching the realism and the complexity of the physical world
Autonomous virtual humans	self-animating synthetic human agents whose motor, perception, behavior, and cognition routines enable them to function autonomously in their simulated environment
Virtual vision simulator	a reality emulator with autonomous virtual humans (pedestrians) that includes the computer vision and camera network machinery necessary to simulate active vision systems. A typical virtual vision simulator uses computer graphics rendering to simulate video capture by passive wide field-of-view (FOV) and active pan/tilt/zoom (PTZ) cameras, and it incorporates computer vision routines that operate upon the synthetic video streams, providing video analysis capabilities similar to those found in physical camera networks; e.g., object detection and tracking

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. We also thank Adam Domurad for his work on visual analysis pipelines and Wiktor Starzyk for developing and implementing the office floor simulator.

References

[1] C.J. Costello, C.P. Diehl, A. Banerjee, H. Fisher, Scheduling an active camera to observe people, in: Proceedings of the ACM International Workshop on Video Surveillance and Sensor Networks, ACM Press, New York, NY, 2004, pp. 39–45.

[2] D. Terzopoulos, T. Rabie, Animat vision: Active vision in artificial animals, Videre: J. Comput. Vis. Res. 1 (1) (1997) 2–19.

[3] G. Salgian, D.H. Ballard, Visual routines for autonomous driving, in: Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, January 1998, pp. 876–882.

[4] T. Rabie, D. Terzopoulos, Active perception in virtual humans, in: Vision Interface (VI 2000), Montreal, Canada, May 2000, pp. 16–22.

[5] T. Rabie, A. Shalaby, B. Abdulhai, A. El-Rabbany, Mobile vision-based vehicle tracking and traffic control, in: Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC 2002), Singapore, September 2002, pp. 13–18.
[6] D. Terzopoulos, Perceptive agents and systems in virtual reality, in: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, Osaka, Japan, October 2003, pp. 1–3.
[7] W. Shao, D. Terzopoulos, Autonomous Pedestrians, Graphical Models 69 (5–6) (2007) 246–274.
[8] W. Shao, D. Terzopoulos, Environmental modeling for autonomous virtual pedestrians, in: Proceedings of the SAE Digital Human Modeling Symposium, Iowa City, Iowa, June 2005.
[9] W. Starzyk, A. Domurad, F.Z. Qureshi, A virtual vision simulator for camera networks research, in: Proceedings of the Nineth Conference on Computer and Robot Vision, Toronto, Canada, May 2012, pp. 1–8, in press.
[10] F. Bertamini, R. Brunelli, O. Lanz, A. Roat, A. Santuari, F. Tobia, Q. Xu, Olympus: An ambient intelligence architecture on the verge of reality, in: Proceedings of the International Conference on Image Analysis and Processing, Mantova, Italy, September 2003, pp. 139–145.
[11] A. Santuari, O. Lanz, R. Brunelli, Synthetic movies for computer vision applications, in: Proceedings of the IASTED International Conference: Visualization, Imaging, and Image Processing (VIIP 2003), number 1, Spain, September 2003, pp. 1–6.
[12] F.Z. Qureshi, D. Terzopoulos, Surveillance camera scheduling: A virtual vision approach, ACM Multimedia Syst. J. 12 (3) (2006) 269–283.
[13] F.Z. Qureshi, D. Terzopoulos, Smart camera networks in virtual reality, Proceedings of the IEEE 96 (10) (2008) 1640–1656 (Special Issue on Smart Cameras).
[14] F.Z. Qureshi, Demetri Terzopoulos, Planning ahead for PTZ camera assignment and control, in: Proceedings of the Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 09), Como, Italy, August 2009, pp. 1–8.
[15] A.C. Sankaranarayanan, A. Veeraraghavan, R. Chellappa, Object detection, tracking and recognition for multiple smart cameras, Proceedings of the IEEE 96 (10) (2008) 1606–1624 (Special Issue on Smart Cameras).
[16] C. Ding, B. Song, A. Morye, J.A. Farrell, A.K. Roy-Chowdhury, Collaborative sensing in distributed PTZ camera network, IEEE Trans. Image Process. 21 (7) (2012) 3282–3295.
[17] B. Song, A.T. Kamal, C. Soto, C. Ding, J.A. Farrell, A.K. Roy-Chowdhury, Tracking and activity recognition through consensus in distributed camera networks, IEEE Trans. Image Process. 19 (10) (2010) 2564–2579.
[18] Panda3D Game Engine Manual. Retrieved on May 9, 2013, from <http://panda3d.org>.
[19] F.Z. Qureshi, Wiktor Starzyk, Learning proactive control strategies for PTZ cameras, in: Proceedings of the Fifth ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2011), Ghent, Belgium, August 2011, pp. 1–6.
[20] W. Starzyk, F.Z. Qureshi, Multi-tasking smart cameras for intelligent video surveillance systems, in: Proceedings of the Eighth IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS 11), Klagenfurt, August 2011, p. 6.