Virtual Vision:
Simulating Camera Networks in Virtual Reality for Surveillance System Design and Evaluation

Demetri Terzopoulos
University of California, Los Angeles
Computer Science Department
http://www.cs.ucla.edu/~dt

Abstract

I review my research with Faisal Qureshi towards smart camera networks capable of carrying out advanced surveillance tasks with little or no human supervision. A unique centerpiece of our work is the combination of computer vision, computer graphics, and artificial life simulation technologies to develop such networks and experiment with them. Our prototype simulator has enabled us to readily develop and experiment with smart camera networks comprising static and active simulated video surveillance cameras that provide extensive coverage of a large virtual public space, a train station populated by autonomously self-animating virtual pedestrians. The simulated networks of smart cameras perform persistent visual surveillance of individual pedestrians with minimal intervention. Our "Virtual Vision" simulator has been a potent tool in our quest for innovative camera control strategies that naturally address camera aggregation and handoff, are robust against camera and communication failures, and require no camera calibration, detailed world model, or central controller.

1. Introduction

Future visual sensor networks will rely on *smart cameras* for sensing, computation, and communication. Smart cameras are self-contained vision systems, complete with increasingly sophisticated image sensors, power circuitry, (wireless) communication interfaces, and on-board processing and storage capabilities. They provide new opportunities to develop camera sensor networks capable of effective visual coverage of extensive areas—public spaces, disaster zones, battlefields, and even entire ecosystems. These multi-camera systems lie at the intersection of Computer Vision and Sensor Networks, raising research problems in the two fields that must be addressed simultaneously.

In particular, as the size of the 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 camera sensor networks that are capable of performing advanced visual surveillance tasks autonomously, or at least with minimal human intervention.

2. The Virtual Vision Paradigm

Unfortunately, research to this end is very difficult to carry out in the real world because of the expense in both time and money of deploying and experimenting with appropriately complex smart camera networks in large public spaces such as airports or train stations. Moreover, privacy laws generally restrict the monitoring of people in public spaces for experimental purposes. To bypass the legal and cost impediments, we have been advocating *Virtual Vision*, a unique synthesis of computer vision, computer graphics, and artificial life technologies (Fig. 1). Virtual vision is an advanced simulation framework for working with machine vision systems, including smart camera networks, that also offers wonderful rapid prototyping opportunities. Exploiting visually and behaviorally realistic environments, called *reality emulators*, virtual vision offers significantly greater flexibility and repeatability during the camera network design and evaluation cycle, thus expediting the scientific method and system engineering process.

In our work, we employ a virtual train station populated by autonomous, lifelike virtual pedestrians (Fig. 2), wherein we deploy virtual cameras (Fig. 3) that generate synthetic video feeds (Fig. 4) emulating those acquired by real surveillance cameras monitoring public spaces.

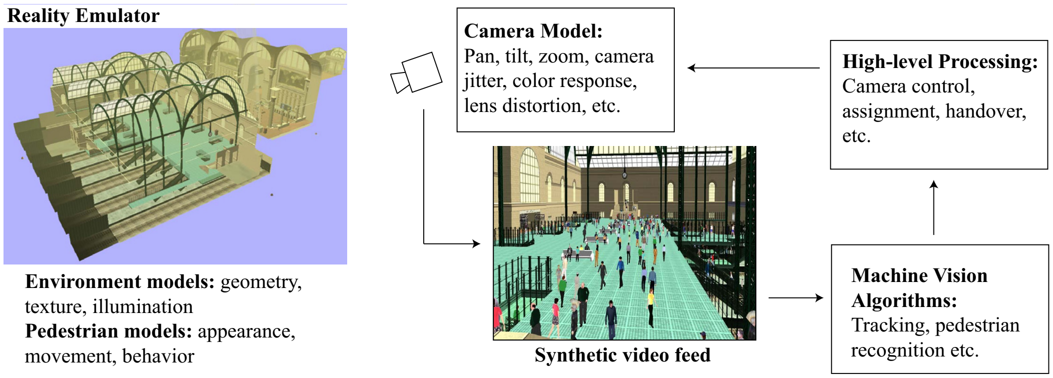

Reality Emulator
Environment models: geometry, texture, illumination
Pedestrian models: appearance, movement, behavior
Camera Model: Pan, tilt, zoom, camera jitter, color response, lens distortion, etc.
Synthetic video feed
High-level Processing: Camera control, assignment, handover, etc.
Machine Vision Algorithms: Tracking, pedestrian recognition etc.

Figure 1. The *Virtual Vision* paradigm (image from [1]).

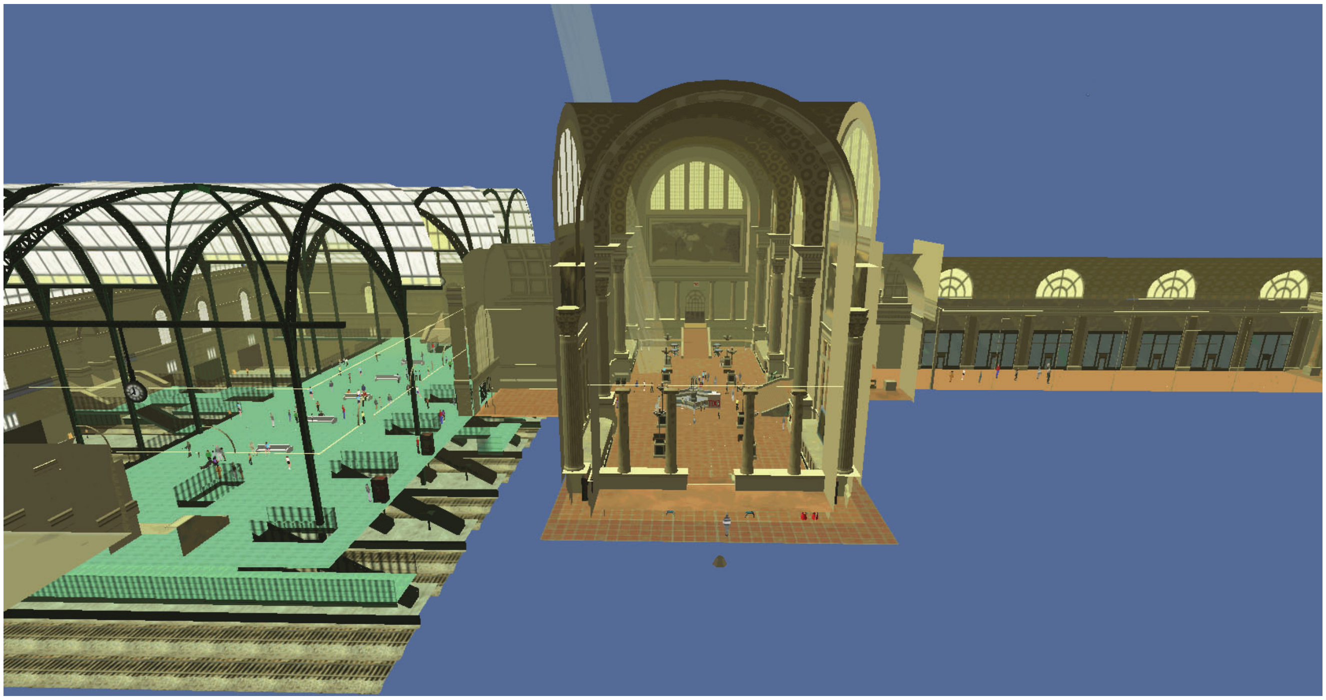


Figure 2. A cutaway side view of the train station model. The main waiting room is at the center, the arcade at the right, and the concourses on the left with the train tracks underneath them.

Despite its sophistication, our simulator runs on high-end commodity PCs, thereby obviating the need to grapple with special-purpose hardware and software. Unlike the real world,

1. the multiple virtual cameras are very easily reconfigurable in the virtual space,
2. we can readily determine the effect of algorithm and parameter modifications because experiments are perfectly repeatable in the virtual world, and
3. the virtual world provides readily accessible ground-truth data for the purposes of camera network algorithm validation.

It is important to realize that our simulated camera networks always 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 efforts to accelerate 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

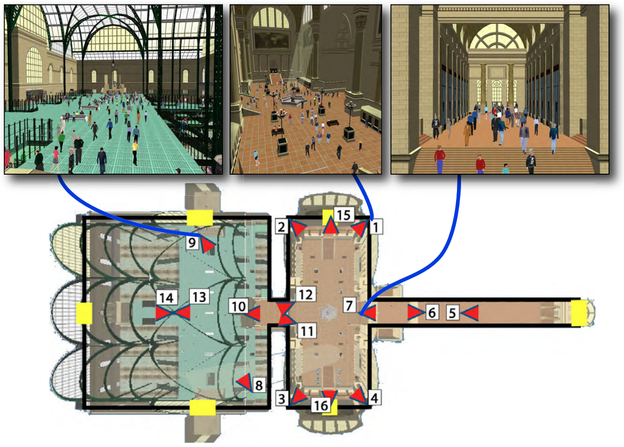


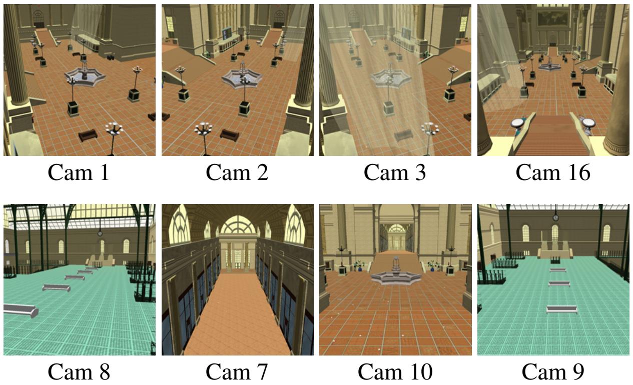
Figure 3. Plan view of the (roofless) virtual Penn Station environment, revealing the concourses and train tracks (left), the main waiting room (center), and the shopping arcade (right). (The yellow rectangles indicate pedestrian portals.) An example camera network is illustrated, comprising 16 simulated active (pan-tilt-zoom) video surveillance cameras. Synthetic images from cameras 1, 7, and 9 (from [1]).

smart, active PTZ cameras requires scene reenactment for every experimental run, which is almost always infeasible when many human subjects are involved. 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 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.

3. Background

In 1997, Tamer Rabie and I introduced a purely software-based approach to designing active vision systems, called *animat vision* [2]. Our approach prescribed the use of artificial animals (or animats) situated in physics-based virtual worlds to study and develop active vision systems, rather than struggling with physical hardware—the cameras and wheeled mobile robots typically used by computer vision researchers. We demonstrated the animat vision approach by implementing biomimetic active vision systems for virtual animals and humans [3].


Cam 1 Cam 2 Cam 3 Cam 16

Cam 8 Cam 7 Cam 10 Cam 9

Figure 4. Synthetic video feeds from multiple virtual surveillance cameras situated in the (empty) Penn Station environment. Camera locations are shown in Fig. 3.

Envisioning a large computer-simulated world inhabited by virtual humans that look and behave like real humans, I then proposed the idea of using such visually and behaviorally realistic environments, which I called "reality emulators", to design machine vision systems, particularly surveillance systems [4].

With support from DARPA, Wei Shao and I [1, 5] developed a prototype reality emulator, comprising a reconstructed model of the original Pennsylvania Station in New York City populated by virtual pedestrians, autonomous agents with functional bodies and brains (Fig. 2). The simulator incorporates a large-scale environmental model of the train station with a sophisticated pedestrian animation system including behavioral, perceptual, and cognitive human simulation algorithms. It 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 through congested areas and portals, queue in lines as necessary, purchase train tickets at the ticket booths in the main waiting room, sit on benches when tired, obtain food/drinks from vending machines when hungry/thirsty, etc., and eventually proceed to the concourses and descend stairs to the train platforms. Standard computer graphics techniques render the busy urban scene with considerable geometric and photometric detail (Fig. 3). The details of the train station simulator are presented in [6].

Faisal Qureshi and I incorporated virtual cameras into the simulator, eventually implementing a prototype virtual

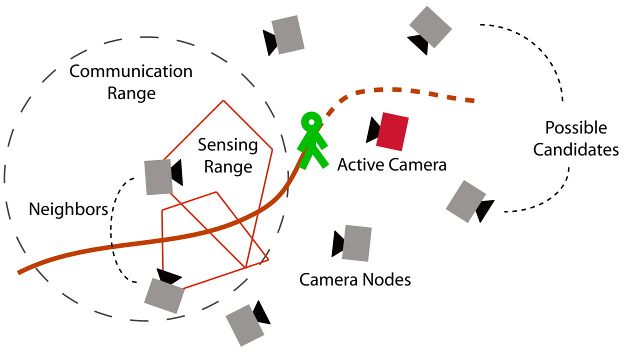


Figure 5. A camera network for video surveillance consists of camera nodes that can communicate with other nearby nodes. Collaborative, persistent surveillance requires that cameras organize themselves to perform camera handover when the observed subject moves out of the sensing range of one camera and into that of another.

vision system with the following camera node 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 sent 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) camera nodes can fail. These assumptions ensure that our virtual camera network faithfully mimics the operational characteristic of a real sensor network. We have presented our work in a series of conference papers [7, 8, 9, 10, 11, 12, 13] as well as in two archival journal articles [14, 15]. Additional details are available in Qureshi's PhD thesis [16].

4. Persistent Surveillance

Consider how a network of smart cameras may be used in the context of video surveillance (Fig. 5). Any two camera nodes that are within communication range of each other are considered neighbors and the network can easily be modified through removal, addition, or replacement of camera nodes.

A human operator spots one or more mobile pedestrians of interest in a video feed and, for example, requests the network to "zoom in on this pedestrian," "observe this pedestrian," or "observe the entire group." The successful execution and completion of these tasks requires an intelligent allocation and coordination of the available cameras. In particular, the network must decide which cameras should track the pedestrian and for how long. The accuracy with which individual camera nodes are able to compute their relevance to the task at hand determines the overall performance of the network.

A detailed world model that includes the location of cameras, their fields of view, pedestrian motion prediction models, occlusion models, and pedestrian movement pathways may allow (in some sense) optimal allocation of camera resources; however, it is cumbersome and in most cases infeasible to acquire such a world model. Our approach eschews such detailed knowledge. We assume only that a pedestrian can be identified with reasonable accuracy by the camera nodes.

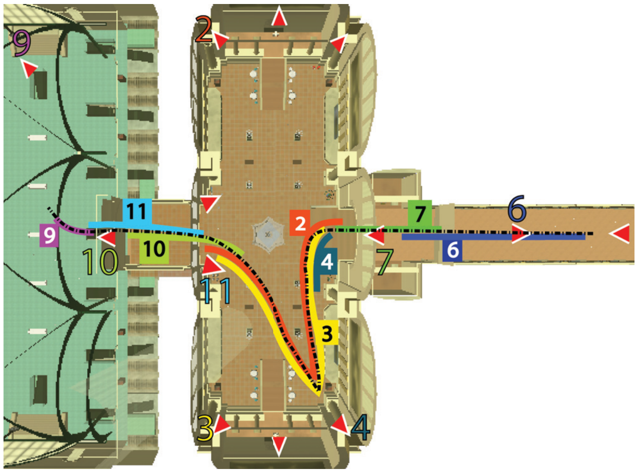
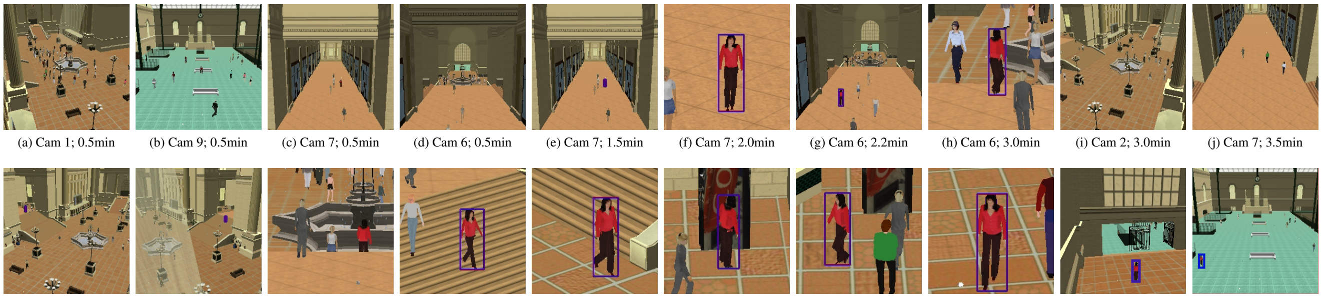


Figure 6. A pedestrian of interest walking through the train station for 15 minutes is automatically observed successively by Cameras 7, 6, 2, 3, 10, and 9 (refer to Fig. 3) as she makes her way from the arcade through the main waiting room and into the concourse. The dashed contour shows the pedestrian's path. The camera numbers are color coded and the portion of the path walked while the pedestrian is being observed by a particular camera is highlighted with the associated color.

We have developed a novel camera network control strategy that does not require camera calibration, or a detailed world model, or a central controller. The overall behavior of the network is the consequence of the local processing at each node and internode communication. The network is robust to node and communication failures. Moreover, it is scalable because of the lack of a central controller. Visual surveillance tasks are performed by groups of one or more camera nodes. These groups, which are created on the fly, define the information sharing parameters and the extent of collaboration between nodes. A group evolves—i.e., old nodes leave the group and new nodes join it—during the lifetime of the surveillance task. One node in each group acts as the group supervisor and is responsible for group-level decision making. We have developed a novel *constraint satisfaction problem* (CSP) formulation for resolving interactions between groups.

I refer the reader to [15, 16] for the technical details of our approach, as well as for the details regarding the following experimental results.

To date, we have simulated our smart camera network with up to 16 stationary and/or PTZ virtual cameras in the virtual train station populated with up to 100 autonomous


(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 7. 15-minute persistent observation of a pedestrian of interest as she makes her way through the train station (refer to Fig. 6). (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.

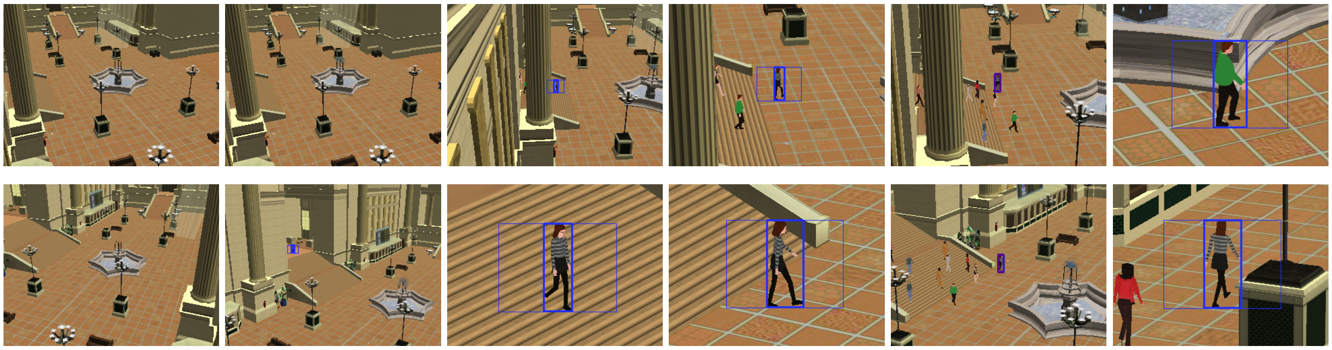

(a) (b) (c) (d) (e) (f)

Figure 8. Camera assignment and conflict resolution. (a) Camera 1 (top row) and Camera 2 (bottom row) observe the main waiting room. (b) Camera 2 starts observing a pedestrian as soon as she enters the scene. (c)–(d) Camera 1 recognizes the target pedestrian by using the pedestrian signature computed by Camera 2. Cameras 1 and 2 form a group to observe the first pedestrian. (e) The operator issues a second task for the camera network, to observe the pedestrian wearing green. The two cameras pan out to search for the latter. They decide between them which will carry out each of the two tasks. (f) Camera 1 is better suited to observing the pedestrian in the green top while Camera 2 continues observing the original pedestrian.

pedestrians.

For the 15-minute simulation illustrated in Fig. 6 and 7, with 16 active PTZ cameras in the train station as indicated in Fig. 3, an operator selects the female pedestrian with the red top visible in Camera 7 (Fig. 7(e)) and initiates an *observe* task. Camera 7 forms a task group and begins tracking the pedestrian. Subsequently, Camera 7 recruits Camera 6, which in turn recruits Cameras 2 and 3 to observe 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 persistently observe the pedestrian during her stay in the main waiting room, where she also visits a vending machine. When the pedestrian enters the portal connecting the main waiting room to the concourse, Cameras 10 and 11 are recruited and they take over the group from Cameras 2 and 3. Cameras 2 and 3 leave the group and return to their default states. Later, Camera 11, which is now acting as the group's supervisor, recruits Camera 9, which observes the pedestrian as she enters the concourse.

Fig. 8 illustrates camera assignment and conflict resolution. First, Cameras 1 and 2 situated in the main waiting room successfully form a group to observe the first pedestrian that enters the scene, and there is only one active task. When the user specifies a second task—follow the pedestrian wearing the green top—the cameras decide to dissolve the group and reassign themselves. They decide among themselves that Camera 1 is more suitable for observing the pedestrian in the green top. Camera 2 continues observing the first pedestrian that entered the scene. The cameras are able to handle the two observation tasks completely autonomously and the interaction between them is strictly local without involvement from any of the other 14 cameras.

5. Conclusions

Future video surveillance systems will be networks of stationary and active cameras (and other sensors) capable of maintaining extensive urban environments under persistent surveillance with minimal reliance on human operators. Such systems will require not only robust, low-level vision routines, but also new camera network methodologies. Our work is a step toward the realization of such smart camera networks and our initial results appear promising.

A unique and important aspect of our work is the Virtual Vision framework; we design, experiment with, validate, and refine our prototype video surveillance systems in virtual reality. To date, this has taken the form of a realistic train station environment populated by lifelike, autonomously self-animating virtual pedestrians. Our sophisticated camera network simulator should continue to facilitate our ability to design such large-scale networks and conveniently experiment with them on commodity personal computers.

The Virtual Vision approach can be extended in several important directions. Our current simulator is restricted to cameras mounted in fixed locations. In particular, we would like to build a simulator of cameras mounted on fleets of airborne platforms. This would require the faithful simulation of unmanned aerial vehicles (UAVs) and their flight capabilities and limitations. In theory, we can readily generalize our sensor model to simulate current and future UAV vision sensors (active PTZ, EPTZs, and gigapixel wide-FOV such as ARGUS, IR, etc.). We would also need to construct a suitably extensive, geometrically and photometrically accurate terrestrial site model populated by autonomous virtual humans, which can be realistically imaged by the UAV-mounted camera sensors. Finally, we would need to simulate the appropriate wireless network communication capabilities among the UAVs in flight, as well as between them and a ground station. Given such a simulator, we will be able to work on online sensornet control and active visual surveillance algorithms to implement a "smart visual sensor network in the sky" capable of persistent surveillance and activity recognition on the ground. These ideas can be extended to combinations of air and ground mobile multi-camera platforms.

Acknowledgments

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

References

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

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

[3] T. Rabie and D. Terzopoulos. Active perception in virtual humans. In *Proc. Vision Interface (VI'00)*, pages 16–22, Montreal, Canada, May 2000.

[4] D. Terzopoulos. Perceptive agents and systems in virtual reality. In *Proc. ACM Symp. on Virtual Reality Software and Technology (VRST'03)*, pages 1–3, Osaka, Japan, Oct 2003.

[5] W. Shao and D. Terzopoulos. Environmental modeling for autonomous virtual pedestrians. In *Proc. SAE Digital Human Modeling Symposium*, Iowa City, Iowa, Jun 2005.

[6] W. Shao and D. Terzopoulos. Autonomous pedestrians. *Graphical Models*, **69**(5–6):246–274, Sep/Nov 2007.

[7] F. Qureshi and D. Terzopoulos. 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-PETS'05)*, pages 177–184, Beijing, China, Oct 2005.

[8] F. Qureshi and D. Terzopoulos. Surveillance camera scheduling: A virtual vision approach. In *Proc. ACM Int. Workshop on Video Surveillance and Sensor Networks (VSSN'05)*, pages 131–139, Singapore, Nov 2005.

[9] F. Qureshi and D. Terzopoulos. Distributed coalition formation in visual sensor networks: A virtual vision approach. In *Proc. IEEE Int. Conf. on Distributed Computing in Sensor Systems (DCOSS'07)*, *LNCS* **4549**, pages 1–20, Santa Fe, NM, Jun 2007.

[10] F. Qureshi and D. Terzopoulos. Surveillance in virtual reality: System design and multicamera control. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'07)*, pages 1–8, Minneapolis, MN, Jun 2007.

[11] F. Qureshi and D. Terzopoulos. Virtual vision and smart cameras. In *Proc. First ACM/IEEE Int. Conf. on Distributed Smart Cameras*, pages 87–94, Vienna, Austria, Sep 2007.

[12] F. Qureshi and D. Terzopoulos. Virtual vision: Visual sensor networks in virtual reality. In *Proc. ACM Symposium on Virtual Reality Software and Technology (VRST'07)*, pages 247–248, Newport Beach, CA, Nov 2007.

[13] F. Qureshi and D. Terzopoulos. Multi-Camera Control Through Constraint Satisfaction for Persistent Surveillance. In *Proc. 5th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS'08)*, pages 1–8, Santa Fe, NM, Sep 2008.

[14] F. Qureshi and D. Terzopoulos. Surveillance camera scheduling: A virtual vision approach. *Multimedia Systems*, **12**:269–283, Dec 2006.

[15] F. Qureshi and D. Terzopoulos. Smart camera networks in virtual reality. *Proceedings of the IEEE*, **96**(10):1640–1656, Oct 2008.

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