

# A Simulation Framework for Camera Sensor Networks Research

Faisal Qureshi<sup>1</sup> and Demetri Terzopoulos<sup>2,1</sup>

<sup>1</sup>Department of Computer Science, University of Toronto, Toronto, ON, Canada

<sup>2</sup>Computer Science Department, University of California, Los Angeles, CA, USA

**Keywords:** Computer vision, virtual vision, virtual reality, simulated camera sensor networks, multi-camera control and scheduling, persistent surveillance

## Abstract

We present our progress on the Virtual Vision paradigm, which prescribes visually and behaviorally realistic virtual environments, called “reality emulators”, as a simulation tool to facilitate research on large-scale camera sensor networks. We have successfully exploited a prototype reality emulator—a virtual train station populated with numerous autonomous pedestrians—to rapidly develop novel solutions to challenging problems, such as multi-camera control and scheduling for persistent human surveillance by next-generation networks of smart cameras deployed in extensive public spaces.

## 1. INTRODUCTION

Camera sensor networks are becoming increasingly important to next generation applications in surveillance, in environment and disaster monitoring, in city and traffic management, and in the military. In contrast to current video surveillance systems, camera sensor networks are characterized by smart cameras, large network sizes, and ad hoc deployment. These systems lie at the intersection of machine vision and sensor networks, raising issues in the two fields that must be addressed in unison. The effective visual coverage of extensive areas—urban environments, disaster zones, and battlefields—requires multiple cameras to collaborate towards common sensing goals. As the size of the camera 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 in performing advanced surveillance tasks. Therefore, it is desirable to develop camera sensor networks that are capable of performing visual surveillance tasks autonomously, or at least with minimal human intervention. This is a challenging long-term problem that will require a great deal of research.

Unfortunately, however, deploying a large-scale visual sensor network in the real world is a major undertaking whose cost can easily be prohibitive for most researchers interested in designing and experimenting with sensor networks. Moreover, privacy laws generally restrict the monitoring of people in public spaces for experimental purposes. These impediments hinder research and development.

This paper reports our progress on *Virtual Vision*—a simulation framework that facilitates research on large-scale camera networks situated in extensive, richly populated public spaces. A unique synergy of computer vision, computer graphics, and artificial life technologies, virtual vision prescribes the simulation of camera networks in visually and behaviorally realistic virtual environments, called “reality emulators”. Exploiting a prototype reality emulator, a virtual train station populated with numerous autonomous pedestrians, we have successfully developed multi-camera control and scheduling algorithms for next-generation networks of smart cameras (Fig. 1).

Legal and cost impediments aside, the use of realistic virtual environments in sensor network research offers wonderful rapid prototyping opportunities with significantly greater flexibility during the design and evaluation cycle, thus expediting the engineering process. Despite its sophistication, our simulator runs on high-end commodity PCs, thus 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 visual sensor network algorithm validation. In addition, by simply prolonging virtual-world time relative to real-world time, we can evaluate the competence of expensive, currently non-real-time algorithms and thereby gauge the potential payoff of accelerating them through more efficient software and/or special-purpose hardware implementations.

## 2. RELATED WORK

Over ten years ago, Terzopoulos and Rabie [19] introduced a purely software-based approach to designing active computer vision systems. Rather than the camera and mobile robot hardware that typically preoccupied active vision researchers at the time, their “animat vision” approach espoused artificial animals (animats) situated in physics-based virtual worlds as biomimetically embodied autonomous agents in which to implement and experiment with active vision systems. They developed such systems in self-animating virtual humans and lower animals [14]. Key animat vision algorithms were later adapted for use in a mobile vehicle tracking and traffic control system [13], which is a testa-

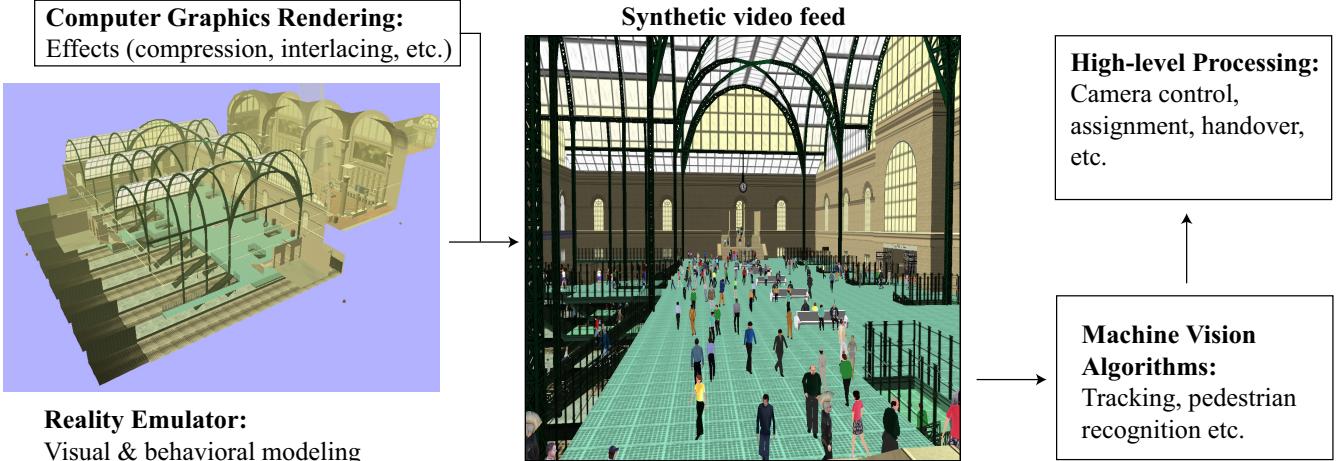


Figure 1: Virtual Vision: A simulation framework for camera networks research

ment to the value of the approach in computer vision research and vision system prototyping.

In 2003, Terzopoulos [18] proposed the idea of using visually and behaviorally realistic virtual environments, called reality emulators, in designing machine vision systems. He envisioned a computer simulated world that approaches the complexity and realism of the real world. Richly populated by virtual humans that look, behave, and move like real humans, such a virtual world, he proposed, could be used in revolutionary ways and its impact across multiple scientific disciplines would be profound. Over the past two years, we have been working toward realizing this idea [10, 11, 12]. We have been using a prototype reality emulator that was developed in 2005 by Shao and Terzopoulos [16], a virtual reconstruction of the original Pennsylvania Station in New York City, inhabited by virtual pedestrians, autonomous agents with functional bodies and brains.

In concordance with our virtual vision paradigm, Santucci et al. [15] advocated the development and evaluation of pedestrian segmentation and tracking algorithms using synthetic video generated within a virtual museum simulator occupied by several non-autonomous, scripted characters. They focus on low-level computer vision, whereas our work goes beyond this to include high-level computer vision issues, especially multi-camera control and scheduling in large-scale networks of smart cameras.

Previous work on multi-camera systems has dealt with object identification, classification, and tracking [2, 20, 6]. Many researchers have proposed camera network calibration strategies to achieve robust identification and tracking of moving objects from multiple viewpoints [9, 5]. However, little attention has been paid to the problem of controlling and/or scheduling active cameras when there are more objects to be monitored in the scene than there are cameras.

Master-slave systems have been proposed where a stationary wide-field-of-view camera controls one or more active pan/tilt/zoom (PTZ) cameras [3, 4]. Generally speaking the cameras are assumed to be calibrated and the total coverage of the cameras is restricted to the field of view of the stationary camera.

In the sensor networks community, optimal sensor node selection with respect to a given task has received much attention. In particular, many researchers have dealt with the problem of forming sensor groups based on task requirements and resource availability [21, 7]. Nodes comprising sensor networks are usually untethered sensing units with limited onboard power reserves. Hence, a crucial concern in sensor networks is the energy expenditure at each node, which determines the lifespan of a sensor network [1]. Node communications have large power requirements; therefore, sensor network control strategies attempt to minimize the internode communication [22]. Presently, we do not address this issue; however, the communication protocol proposed here limits the communication to the active nodes and their neighbors.

The node selection and grouping strategy that we describe later in this paper was inspired by the ContractNet distributed problem solving protocol [17] and it realizes group formation via internode negotiation. Group formation conflicts are represented as a Constraint Satisfaction Problem (CSP), which we solve using “centralized” backtracking. We avoid distributed constraint optimization techniques due to their explosive communication requirements [8].

### 3. THE REALITY EMULATOR

Our reality emulator, comprising the virtual train station model populated with self-animating pedestrians, is described in detail elsewhere [16]. Taking an advanced, artificial life

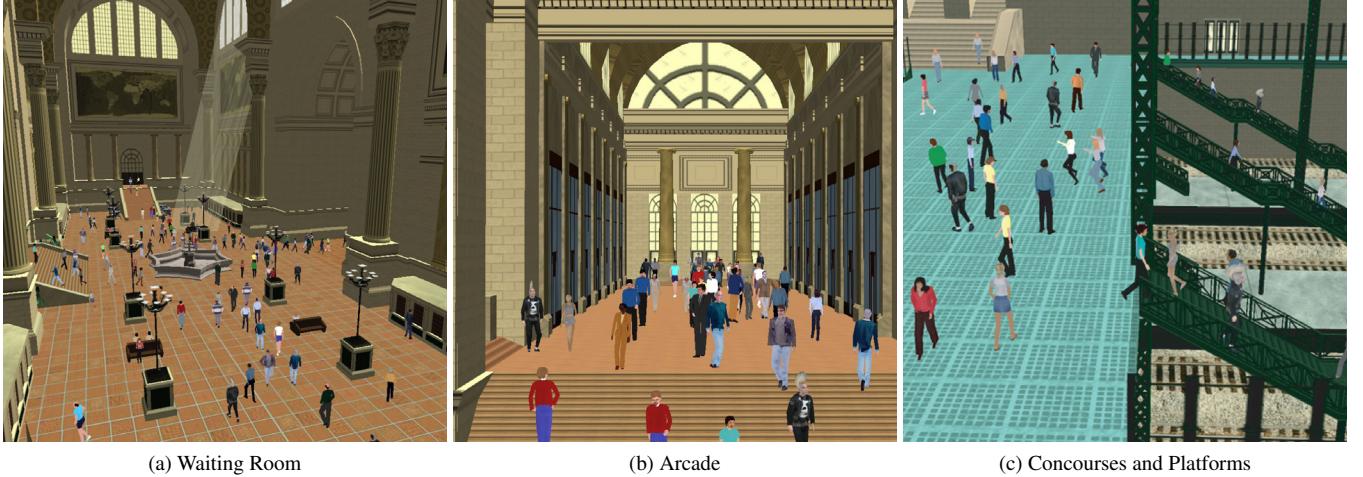


Figure 3: A large-scale virtual train station populated by self-animating virtual humans [16].

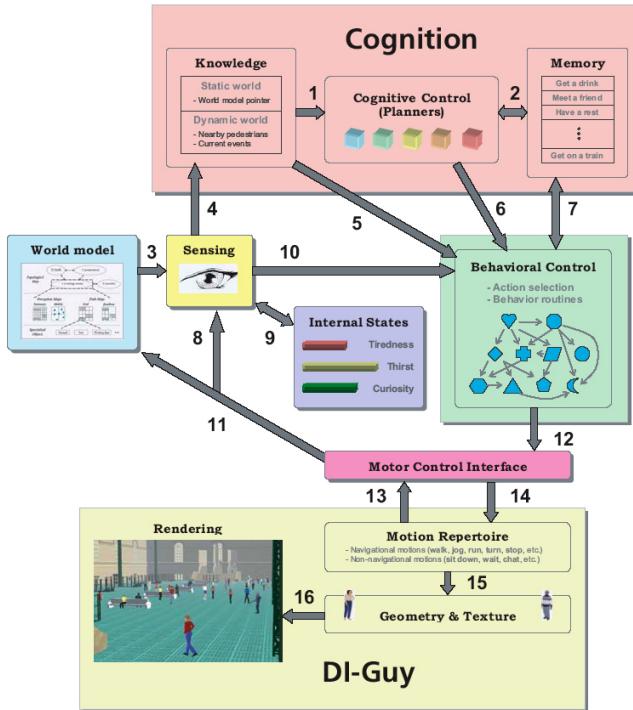


Figure 2: Autonomous pedestrian simulation model (from [16]).

modeling approach, the pedestrian animation system combines behavioral, perceptual, and cognitive human simulation algorithms (Fig. 2). Running on a high-end PC, the simulator can efficiently synthesize well over 1000 self-animating pedestrians performing a rich variety of activities in the large-

scale indoor urban environment. Like real humans, the synthetic pedestrians are fully autonomous. They perceive the virtual environment around them, analyze environmental situations, make decisions and behave naturally within the train station. They can enter the station, avoiding collisions when proceeding through portals and congested areas, queue in lines as necessary, purchase train tickets at the ticket booths in the main waiting room, sit on benches when they are tired, purchase food from vending machines when they are hungry, etc., and eventually proceed to the concourse area and down to the train tracks. The PC graphics pipeline renders the busy urban scene with considerable geometric and photometric detail, as shown in Fig. 3.

#### 4. VIRTUAL VISION

Little attention has been paid to the problem of controlling/scheduling active cameras to provide visual coverage of an extensive public space, such as an airport or train station. Our virtual vision approach has enabled us to expeditiously carry out research that addresses this problem. Specifically, we deploy synthetic visual sensor networks comprising static and active simulated video surveillance cameras that provide perceptive coverage of the virtual train station (Fig. 4). The multiple virtual cameras generate synthetic video feeds (Fig. 5) that emulate those generated by real surveillance cameras monitoring public spaces.

The performance of the camera network is ultimately tied to the capabilities of the low-level machine vision routines responsible for gathering the sensory data. Consequently, when working with camera networks, it is important to make accurate assumptions about the capabilities and performance of the low-level visual sensing processes. We ensure the validity of our assumptions about the low-level visual sensing by im-

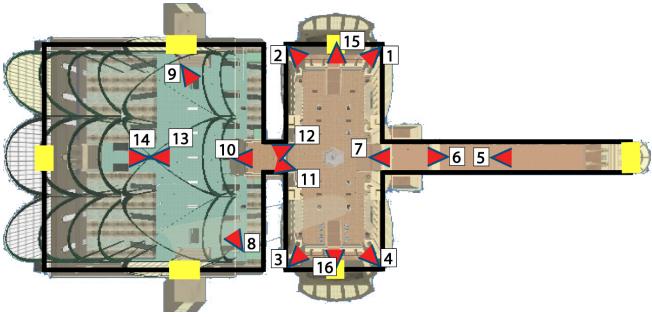


Figure 4: Plan view of the virtual Penn Station environment with the roof not rendered, revealing the concourses and train tracks (left), the main waiting room (center), and the long shopping arcade (right). (The yellow rectangles indicate station pedestrian portals.) An example visual sensor network comprising 16 simulated active (pan/tilt/zoom) video surveillance cameras is shown.

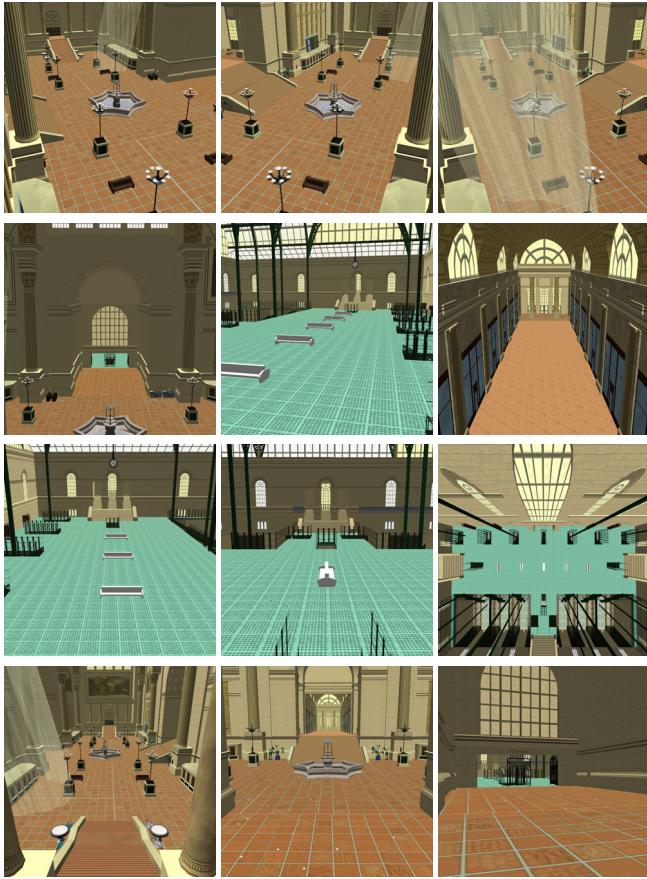
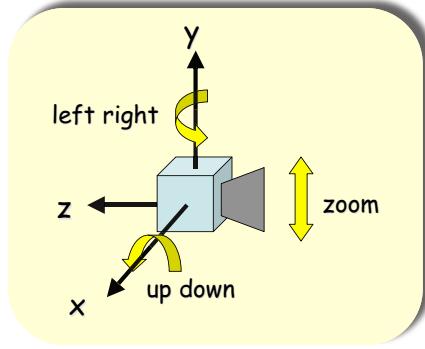
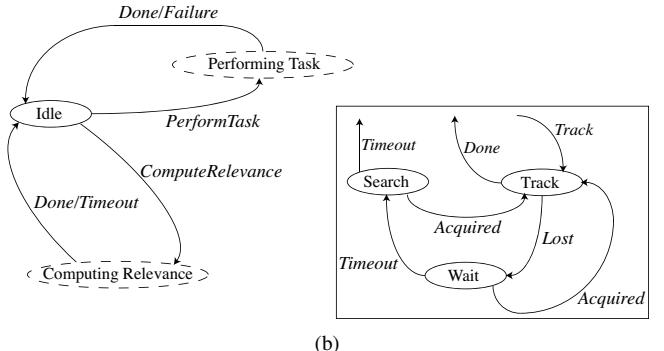


Figure 5: Synthetic video feeds from multiple virtual surveillance cameras situated in the (empty) Penn Station environment.



(a)



(b)

Figure 6: (a) Active PTZ camera. (b) Camera behavioral controller. Dashed states contain the child finite state machine shown in the inset (lower right).

plementing a pedestrian tracking system that operates solely upon the synthetic video captured through the virtual cameras.

Each virtual camera node in the sensor network is an active sensor with a repertoire of camera behaviors (Fig. 6). The camera is also able to perform low-level visual processing. In particular, it has a suite of visual analysis routines for pedestrian recognition and tracking, which we dub “Local Vision Routines” (LVRs). LVRs are computer vision algorithms that operate upon the video generated by virtual cameras to identify, locate, and track the pedestrians present in the scene. LVRs faithfully mimic the performance of a state-of-the-art pedestrian recognition and tracking system and exhibit errors usually associated with a pedestrian tracking system operating upon real footage, including the loss of track due to occlusions, sudden change in lighting, and bad segmentation. Tracking sometimes locks onto the wrong pedestrian, especially if the scene contains multiple pedestrians with similar visual appearance, i.e., wearing similar clothes.

We model the variation in color response across cameras by manipulating the Hue, Saturation, and Value channels of the rendered image. Our visual analysis routines rely on

color-based appearance models to track pedestrians; therefore, camera handovers are sensitive to the variations in the color response of different cameras. Bandwidth is at a premium in sensor networks, in general, and in camera networks, in particular. In many instances, images captured by camera nodes are transmitted to a central location for analysis, storage, and monitoring purposes. Routinely camera nodes exchange information among themselves during camera handover, camera coordination, and multi-camera sensing operations. The typical data flowing in a camera network is the image/video data, which places much higher demands on a network infrastructure than, say, alpha-numeric or voice data. Consequently, in order to keep the bandwidth requirements within acceptable limits, camera nodes compress the captured images and video before transmitting them to other camera nodes or to the monitoring station. Compression artifacts together with the low-resolution of the captured images/video pose a challenge for visual analysis routines. Compression artifacts, therefore, are relevant to camera network research. We introduce compression effects into the synthetic video by passing it through a JPEG compressor/decompressor stage before passing it onto the pedestrian recognition and tracking module.

Nevertheless, the synthetic video captured by our virtual cameras lacks some of the subtleties of real video, and the evaluation of any machine vision system will ultimately require the latter. However, we believe that our simulator enables us to develop and test camera network control algorithms under realistic assumptions derived from physical camera networks, and that these algorithms should readily port to the real world. Our high-level camera control routines do not need to process any raw video directly. They are dependent on the lower-level recognition and tracking routines, which mimic the performance (including failure modes) of a state-of-the-art pedestrian localization and tracking system and generate realistic input data for the high-level routines.

In particular, pedestrian tracking can fail due to occlusions, poor segmentation, bad lighting, or crowding. Tracking sometimes locks on the wrong pedestrian, especially if the scene contains multiple pedestrians with similar visual appearance; i.e., wearing similar clothes. Our imaging model emulates artifacts that are of interest to camera network researchers, such as video compression and interlacing. It also models camera jitter and imperfect color response.

## 5. SENSOR NETWORK RESEARCH

In the context of smart camera networks and visual surveillance, we have developed a novel camera network control strategy that does not require camera calibration, a detailed world model, or a central controller. Our scheme is robust to camera node failures and communication errors. It enables a network of collaborating smart cameras to provide perceptive

scene coverage and perform persistent surveillance with minimal intervention from a human operator.

Each camera node is an autonomous agent capable of communicating with nearby nodes. The LVRs determine the sensing capabilities of a camera node, whose overall behavior is determined by the LVR (bottom-up) and the current task (top-down). Each camera can *fixate* and *zoom* in on an object of interest (Fig. 6(a)). The fixation and zooming routines are image driven and do not require any 3D information, such as camera calibration or a global frame of reference. The *fixate* routine brings the region of interest—e.g., the bounding box of a pedestrian—into the center of the image by tilting the camera about its local  $x$  and  $y$  axes. The *zoom* routine controls the field of view of the camera such that the region of interest occupies the desired percentage of the image. The camera behavioral controller enables the camera to achieve its high-level sensing goals as determined by the current task. Typical sensing goals might be, “look at the pedestrian  $i$  at location  $(x, y, z)$  for  $t$  seconds,” or “track the pedestrian whose appearance signature is  $h$ .”

The camera controller is an augmented hierarchical finite state machine (Fig. 6(b)). In its default state, *Idle*, the camera node is not involved in any task. A camera node 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, as we will explain in the next section. For example, a camera can use visual search to find a pedestrian that matches the appearance-based signature passed by the querying node. The relevance encodes the expectation of how successful a camera node will be at a particular sensing task. The camera returns to the *Idle* state when it fails to compute the relevance because it cannot find a pedestrian that matches the description. When the camera successfully finds the desired pedestrian, however, it returns the relevance value to the querying node. The querying node passes the relevance value to the leader (leader node) of the group, which decides whether or not to include the camera node in the group. The camera goes into the *Performing-Task* state upon joining a group, where the embedded child finite state machine (Fig. 6(b) inset) hides the sensing details from the top-level controller and enables the node to handle short-duration sensing (tracking) failures. Built-in timers allow the camera node to transition into the default state instead of hanging in some state waiting for a message from another node that might never arrive due to a communication error or node failure. The camera node returns to its default state after finishing a task using the *reset* routine, which is a PD controller that attempts to minimize the error between the current pan/tilt/zoom settings and the default pan/tilt/zoom settings.

We model the virtual cameras as nodes in a communication

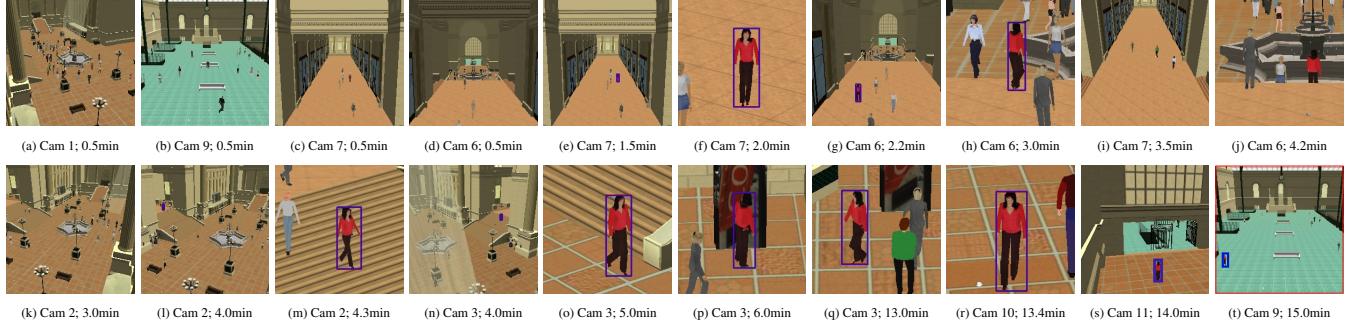


Figure 7: A pedestrian is successively tracked by Cameras 7, 6, 2, 3, 10, and 9 (see Fig. 4) as she makes her way through the station to the concourse. (a-d) Cameras observing the station. (e) Operator selects a pedestrian in feed 7. (f) Camera 7 has zoomed in on the pedestrian, (g) Camera 6, which is recruited by Camera 7, acquires the pedestrian. (h) Camera 6 zooms in on the pedestrian. (i) Camera 7 reverts to its default mode after losing track of the pedestrian—it is now ready for another task (j) Camera 6 has lost track of the pedestrian. (k) Camera 2. (l) Camera 2, which is recruited by camera 6, acquires the pedestrian. (m) Camera 2 tracking the pedestrian. (n) Camera 3 is recruited by the Camera 6; camera 3 has acquired 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 tracking the pedestrian. (s) Camera 11 is recruited by Camera 10. (t) Camera 9 is recruited by Camera 10.

network that emulates those found in physical camera sensor networks: 1) nodes can communicate directly with their neighbors, 2) if necessary, a node can communicate with another node in the network through multi-hop routing, and 3) imperfect communication. We currently do not model message corruption and interference.

Task-specific camera grouping and handoff are addressed within our camera network protocol as follows: An operator issues a particular sensing request to one of the cameras, which publishes this task to its neighboring cameras. The neighboring cameras respond with their bids on the task. A bid quantifies the suitability of a camera for a particular task, and it is computed through local, onboard processing. Neighbors with the highest bids are selected to form a group with the aim of fulfilling the sensing task. The *group*, which formalizes a collaboration between member cameras, is a dynamic arrangement that evolves throughout the lifetime of the task. At any given time, multiple groups might be active, each performing its respective task.

Additionally, we have developed a new formulation for resolving camera assignment conflicts when multiple observation tasks are active, in the form of a constraint satisfaction problem. Our sensor management scheme is well suited to the challenges of designing camera networks for surveillance applications that are potentially capable of fully automatic operation.

The full details of our virtual vision work are given in [12].

## 5.1. Results

We have tested our algorithms by deploying virtual camera networks comprising up to 16 uncalibrated active

pan/tilt/zoom and passive cameras within our reality emulator. The train station simulation environment is populated with up to 100 self-animating pedestrians exhibiting realistic commuter behaviors. Fig. 7 demonstrates our distributed surveillance system following a pedestrian as she makes her way through the train station. The cameras automatically form groups and perform hand-offs to keep the pedestrian persistently in view despite occasional tracking failures. We have also demonstrated the ability of our system to successfully resolve camera assignment conflicts when multiple observation tasks are active.

For the example shown in Fig. 7, we placed 16 active PTZ cameras in the train station, as shown in Fig. 4. An operator selects the pedestrian with the red shirt in Camera 7 (Fig. 7(e)) and initiates the “follow” task. Camera 7 forms the task group and begins tracking the pedestrian. Subsequently, Camera 7 recruits Camera 6, which in turn recruits Cameras 2 and 3 to track the pedestrian. Camera 6 becomes the leader 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 leader, and leaves the group. Cameras 2 and 3 track the pedestrian during her stay in the main waiting room, where she also visits a vending machine. When the pedestrian starts walking towards the concourse, Cameras 10 and 11 take over the group from Cameras 2 and 3. Cameras 2 and 3 leave the group and return to their default modes. Later Camera 11, which is now acting as the group’s leader, recruits Camera 9, which tracks the pedestrian as she enters the concourse.

Fig. 8 illustrates camera assignment and conflict resolution. Cameras 1 and 2 successfully form a group to observe

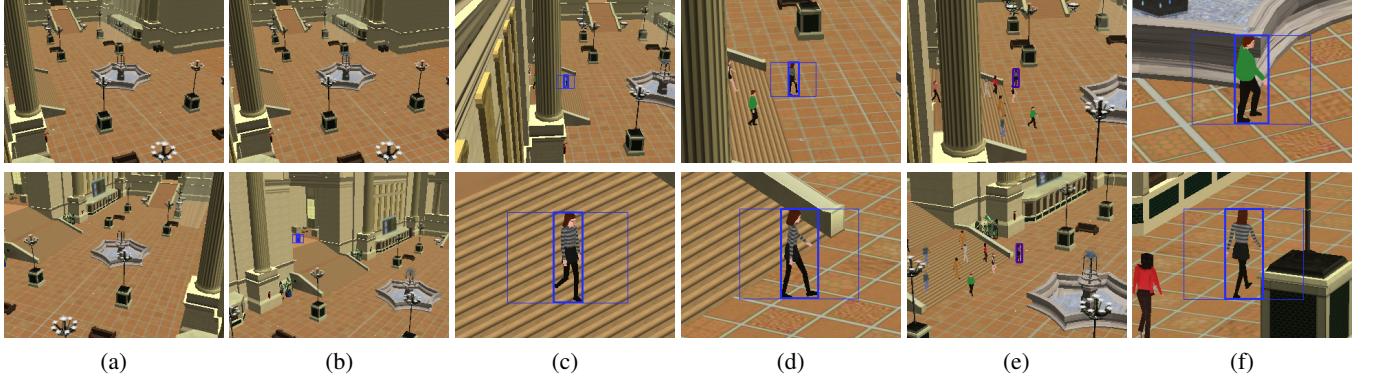


Figure 8: Camera assignment and conflict resolution. Cameras 1 (upper row) and 2 (lower row) are set to observe the first pedestrian who enters the main waiting room (a). Camera 2 starts observing the pedestrian as soon as she enters the scene (b). (c)-(d) Camera 1 recognizes the target pedestrian by using the pedestrian signature computed by Camera 2. Cameras 1 and 2 successfully setup a group to observe the first pedestrian. (e) The operator sets up another goal for the camera network, which is to observe the pedestrian wearing green. The two cameras then pan out to visually search for the green pedestrian and decide between them to each carry out a different task. (f) Camera 1 is deemed more suitable for observing the green pedestrian while camera 2 continues observing the first pedestrian who entered the scene.

the first pedestrian who enters the scene, as there is only one active task. On the other hand, when the user specifies a second task—follow the pedestrian wearing the green sweater—the cameras decide to break the group and reassign themselves. Among themselves, the cameras decide that camera 1 is better suited for observing the pedestrian wearing the green sweater. Camera 2 continues observing the first pedestrian who entered the scene. It bears repeating that the cameras are able to handle the two observation tasks completely autonomously. Additionally, the interaction between the two cameras is strictly local—the other cameras present in the camera network (Fig. 4) are not involved.

## 6. CONCLUSION

Future surveillance systems will comprise networks of static and active cameras capable of providing perceptive coverage of extensive environments with minimal reliance on human operators. Such systems will require not only robust, low-level vision routines, but also novel sensor network methodologies. Our work has taken strides toward their realization.

A unique feature of our efforts is that we have developed prototype surveillance systems in a virtual train station environment populated by autonomous, lifelike pedestrians. This simulator facilitates the design of and experimentation with large-scale visual sensor networks in virtual reality on commodity personal computers. The future of such advanced simulation-based approaches appears promising for the purposes of low-cost sensor network research and multi-camera surveillance system prototyping.

## ACKNOWLEDGEMENTS

The research reported herein was supported in part by a grant from the Defense Advanced Research Projects Agency (DARPA) of the Department of Defense. We thank Tom Strat formerly of DARPA, for his generous support and encouragement. We also thank Wei Shao and Mauricio Plaza-Villegas for implementing the Penn Station simulator and a prototype virtual vision infrastructure within it.

## REFERENCES

- [1] M. Bhardwaj, A. Chandrakasan, and T. Garnett. Upper bounds on the lifetime of sensor networks. In *Proc. IEEE International Conference on Communications*, number 26, pages 785–790, 2001.
- [2] R. Collins, O. Amidi, and T. Kanade. An active camera system for acquiring multi-view video. In *Proc. International Conference on Image Processing*, pages 517–520, Rochester, NY, September 2002.
- [3] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89(10):1456–1477, October 2001.
- [4] C. J. Costello, C. P. Diehl, A. Banerjee, and H. Fisher. Scheduling an active camera to observe people. In *Proc. ACM International Workshop on Video Surveillance and Sensor Networks*, pages 39–45, New York, NY, 2004. ACM Press.
- [5] D. Devarajan, R. J. Radke, and H. Chung. Distributed metric calibration of ad hoc camera networks. *ACM Transactions on Sensor Networks*, 2(3):380–403, 2006.
- [6] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1355–1360, October 2003.

- [7] J. Mallett. *The Role of Groups in Smart Camera Networks*. PhD thesis, Program of Media Arts and Sciences, School of Architecture, Massachusetts Institute of Technology, Feb. 2006.
- [8] P. J. Modi, W.-S. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1–2):149–180, Mar 2006.
- [9] F. Pedersini, A. Sarti, and S. Tubaro. Accurate and simple geometric calibration of multi-camera systems. *Signal Processing*, 77(3):309–334, 1999.
- [10] F. Qureshi and D. Terzopoulos. Surveillance camera scheduling: A virtual vision approach. *ACM Multimedia Systems Journal*, 12:269–283, Dec 2006.
- [11] F. Qureshi and D. Terzopoulos. Distributed coalition formation in visual sensor networks: A virtual vision approach. In J. Aspnes, C. Scheideler, A. Arora, and S. Madden, editors, *Proc. IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2007)*, number 4549 in Lecture Notes in Computer Science, pages 1–20, Santa Fe, NM, June 2007. Springer-Verlag.
- [12] F. Z. Qureshi. *Intelligent Perception in Virtual Camera Networks and Space Robotics*. PhD thesis, Department of Computer Science, University of Toronto, January 2007.
- [13] T. Rabie, A. Shalaby, B. Abdulhai, and A. El-Rabbany. Mobile vision-based vehicle tracking and traffic control. In *Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC 2002)*, pages 13–18, Singapore, Sep 2002.
- [14] T. Rabie and D. Terzopoulos. Active perception in virtual humans. In *Vision Interface (VI 2000)*, pages 16–22, Montreal, Canada, May 2000.
- [15] A. Santuari, O. Lanz, and R. Brunelli. Synthetic movies for computer vision applications. In *Proc. IASTED International Conference: Visualization, Imaging, and Image Processing (VIIP 2003)*, number 1, pages 1–6, Spain, September 2003.
- [16] W. Shao and D. Terzopoulos. Autonomous pedestrians. *Graphical Models*, 69(5-6):246–274, September/November 2007.
- [17] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, Dec 1980.
- [18] D. Terzopoulos. Perceptive agents and systems in virtual reality. In *Proc. ACM Symposium on Virtual Reality Software and Technology*, pages 1–3, Osaka, Japan, October 2003.
- [19] D. Terzopoulos and T. Rabie. Animat vision: Active vision in artificial animals. *Videre: Journal of Computer Vision Research*, 1(1):2–19, September 1997.
- [20] M. Trivedi, K. Huang, and I. Mikic. Intelligent environments and active camera networks. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 804–809, October 2000.
- [21] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich. Collaborative signal and information processing: An information directed approach. *Proceedings of the IEEE*, 91(8):1199–1209, 2003.
- [22] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration for tracking applications. In *IEEE Signal Processing Magazine*, volume 19, pages 61–72. March 2002.

## Biographies

**Faisal Qureshi** is a software developer at Autodesk, Inc., in Toronto. He obtained his PhD in Computer Science from the University of Toronto in 2007. He also holds an MSc in Computer Science from the University of Toronto, and an MSc in Electronics from Quaid-e-Azam University, Pakistan. His research interests include sensor networks, computer vision, and computer graphics. He has also published papers in space robotics. He has interned at ATR Labs (Kyoto, Japan), AT&T Research Labs (Red Bank, NJ, USA), and MDA Space Missions (Brampton, ON, Canada). He is a member of the IEEE and the ACM.

**Demetri Terzopoulos** is the Chancellor’s Professor of Computer Science at the University of California, Los Angeles, and a (status-only) Professor of Computer Science and Electrical & Computer Engineering at the University of Toronto. He graduated from McGill University and obtained his PhD degree ('84) from MIT. He is a Fellow of the ACM, a Fellow of the IEEE, a Fellow of the Royal Society of Canada, and a member of the European Academy of Sciences. His many awards include an Academy Award for Technical Achievement from the Academy of Motion Picture Arts and Sciences for his pioneering work on physics-based computer animation, and the inaugural IEEE Computer Vision Significant Researcher Award for his pioneering and sustained research on deformable models and their applications. He is one of the most highly-cited engineers and computer scientists in the world, with more than 300 published research papers and several volumes, primarily in computer graphics, computer vision, medical imaging, computer-aided design, and artificial intelligence/life.