Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Surveillance Camera Coordination Through Distributed Scheduling

Cash J. Costello and I-Jeng Wang
Applied Physics Laboratory
Johns Hopkins University
Laurel, MD 20723, USA
{cash.costello, i-jeng.wang}@jhuapl.edu

Abstract— **A challenge to scaling a video surveillance system is the amount of human supervision required for control of the cameras. In this paper, we consider the problem of coordinating a network of video cameras for the purpose of identifying people. We pose the problem as a machine scheduling problem where each person is a job that should be scheduled before a deadline. To ensure scalability, we propose a distributed algorithm that only depends on neighbor to neighbor communication. We compare the performance of this algorithm to a localized scheduling approach.**

I. INTRODUCTION

A network of video surveillance cameras can be used to address many security challenges such as facility monitoring and infrastructure protection. For these types of problems, the cameras provide the video needed for tracking people, observing activities and identifying individuals. To fulfill these missions, the network of cameras must provide video at varying resolutions and perspectives. Recognizing a person requires high resolution views. Tracking people or vehicles is best accomplished with cameras with a wide field-of-view. For many security environments, it is difficult, if not impossible, to collect the appropriate video with cameras that have fixed focal lengths and fixed perspectives. For example, in an area without choke points, it is particularly challenging collecting images of people's faces that can be used for identification. A non-practical solution is to litter the area with thousands of cameras to ensure a high resolution image is captured of every person.

A better solution is to use pan-tilt-zoom cameras that can change the perspective and resolution of the video they collect. This enables a much smaller number of cameras to provide the video needed for tasks such as activity analysis and person recognition. The question is how to control these cameras. A simple control strategy like panning the cameras back and forth is not satisfactory. Probably the most common solution in deployed surveillance systems is operator control. One or more human operators watch video from the network of cameras and then direct the cameras based on the observed activity. This is not a scalable solution due to its reliance on trained human operators. It also suffers from being reactive rather than proactive. Ideally, the surveillance system should be collecting appropriate video before a security event occurs to either prevent the event or allow for a forensic investigation.

In this paper, we are investigating coordinating video surveillance cameras to collect high resolution images of people that can be used for identification. In this problem, there are a set of cameras providing fixed, wide field-of-views of the areas of interest. Using computer vision techniques, this video is processed to detect moving objects [1] and track them through the monitored area [2], [3]. The moving objects can then be classified as to whether they are people, vehicles or some other type of object [4]. Figure 1 demonstrates an example of this processing on video from a single camera. A map of activity can be created from this data to show the locations of the moving objects in the monitored areas. Depending on how the data is shared among the cameras, this map can be either local or global in nature. The pan-tilt-zoom cameras (called active cameras in this paper) can now be tasked to capture images of people. To enable identification, the active cameras could collect high resolution pictures of a people's faces.

There will often be more than one person at a time in the area monitored by an active camera. There will also be times of congestion so that it is not possible for a camera to collect high resolution images of every person in the scene. Any person that moves through the region of coverage of a particular active camera will generally pass by other cameras. A tasking decision made for a camera will affect other cameras in the future. This is because a person only has been identified once since that information can be maintained through the tracking being performed by the other cameras in the surveillance system. That person does not have to be identified again by a different camera. The problem being considered in this paper is how to select which person to observe in a coordinated manner. We pose this tasking problem in a scheduling framework where each person is a job that must be processed by an active camera in the video surveillance system. The objective of any scheduling policy is to observe as many people as possible over the network of active cameras.

The general problem of controlling active pan-tilt-zoom cameras to improve human recognition performance has received significant attention. Many such investigations involve one wide field-of-view camera cueing a high resolution, narrow field-of-view camera to collect imagery in constrained, indoor environments. Recent systems presented in the literature that track and image a single individual using multiple cameras include [5], [6], [7], [8], [9]. Systems for tracking and imaging more than one person are presented in [10], [11], [12]. Many of these efforts focus on the problem

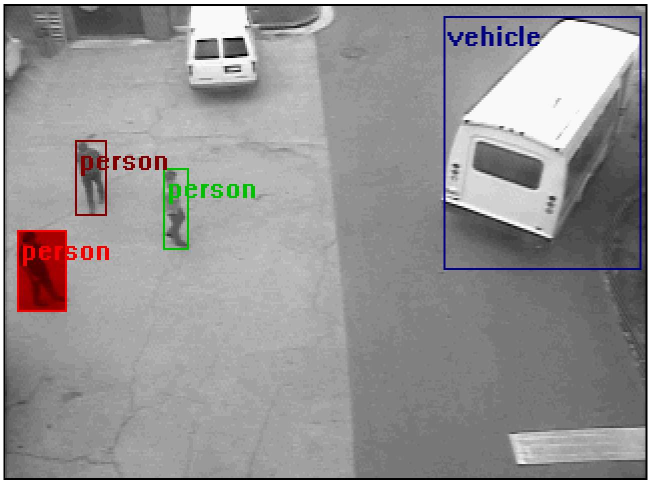

Fig. 1. Tracking and Classifying Moving Objects
This is a picture of the processed video from a camera in a research surveillance system. The person highlighted in red is currently being observed by an active camera for the purpose of identification.

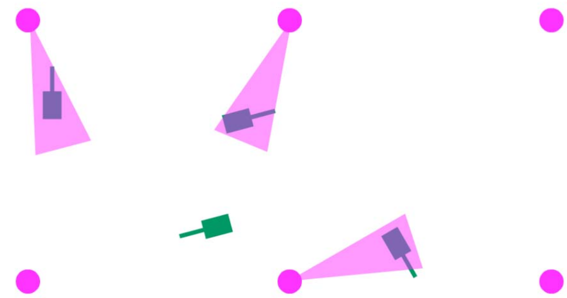

Fig. 2. Active cameras identifying tanks

of 3D tracking, pose estimation or actively tracking a person with the camera. Very little attention has been given to the problem of coordinating the tasking of multiple cameras to observe multiple people.

The organization of this paper is as follows: section II defines the camera tasking problem for identification of people and discusses the role of coordination. This is followed by formulating the problem as a machine scheduling problem in the next section. Section IV presents two scheduling policies for coordinating a network of active cameras for identification. The first policy is earliest deadline first scheduling. The second is a distributed policy based on load balancing that only uses neighbor to neighbor communication. Section V describes the experiments used to evaluate the policies. This is followed by a presentation and discussion of the empirical results. Section VI summarizes the paper.

II. Camera Tasking for Identification

In our problem, the objective of an active cameras is to make observations of unknown people in order to determine their unique global identities. People are continually entering and leaving the area under surveillance. When a person enters the area, the individual is detected and his/her movements are tracked by the other cameras in the system. As the person moves through the area, different active cameras have the opportunity to collect images that could be used for identification.

Each camera can focus on a single person at a time within its region of coverage. Its region of coverage is the total area that can be observed as the camera pans and tilts. The active camera only observes one person at a time because it zooms in until the person fills the field-of-view. We assume the amount of time it takes for the active camera to change its focus from one person to another is small compared to the length of time for an observation. This was a reasonable assumption given the pan and tilt speed of the cameras in our research surveillance system. Additionally, we assume that no two sensors have significant overlap between their regions of coverage. We will mention a modification to the camera tasking algorithms to handle overlap. For the purposes of this paper, a person's identity is established when a camera focuses on it. This only needs to occur once because of the continuity of identity due to tracking.

Figure 2 demonstrates how a set of active cameras in a simple configuration might look at an instant in time. In the figure, tanks are moving through a network of cameras. The camera in the bottom center of the figure has two tanks in its region of coverage and must choose which to view. More generally, in a high traffic environment any active camera may have multiple people passing through its coverage area at any one time. The camera tasking problem is dynamically deciding which person should be identified.

Coordinating the decision of which person is identified by a camera should improve the performance of the entire system. The individuals a camera is not able to identify before they leave the camera's region of coverage must be identified by other cameras in the future. A particular region in the surveillance network may become congested depending on traffic patterns. Congestion means there are more people in an area than the set of cameras covering that area can handle before the people leave. Ideally, active cameras would choose to identify people headed towards this congested area over people headed to an area of low activity. This increases the likelihood of identifying a higher percentage of the people.

Coordination could be performed by a centralized controller. This controller would collect the tracking information from the fixed field-of-view cameras that are distributed throughout the surveillance network. Based on this data, the controller could predict which people will visit which cameras and when this would occur. Using this information, the tasking of the active cameras can be formulated as a constrained optimization problem. The optimization problem must be solved in real-time because the arrival times of people into the monitored area are not known a priori nor are the paths that they will take. This centralized approach has several drawbacks. First, the computational requirements become unmanageable as the size of the surveillance network increases. Second, its performance is highly dependent on predicting the complete paths that people will take through the monitored area. Third, there are significant communication demands with this approach. Wireless networking is already being used in some video surveillance systems so bandwidth limitations are an important consideration.

We have looked into distributed approaches since much less communication bandwidth will be consumed for coordination. Also, each camera will be solving a simpler optimization problem with a smaller state space which eases the computational burden on the surveillance network. The distributed approaches discussed in this paper also do not depend on predictions as much as the centralized approach presented above. Each active camera is provided with a list of all the people that have not been imaged in its region of coverage. The current positions and velocities of the people are also provided by the fixed cameras that are tracking objects.

III. Scheduling Problem Formulation

We formulate the camera coordination problem as the scheduling of m machines. In this formulation, each camera is a machine that can process a single job instance at a time. A person moving through the surveillance network corresponds to a job that should be processed by one of the machines. Processing time is a constant for all jobs and preemption is not allowed. The path that a person takes through the surveillance network determines which machines can process the job. The speed the person moves along this path determines the release times and deadlines for each machine for that job. The release time is the time when the person enters a camera's region of coverage and the deadline is the time the person leaves it. For this problem formulation, we do not allow people to pass through an active camera's region of coverage more than once so there is a single release time and deadline for a camera/person pair. A schedule matching jobs to machines at particular times must be developed without the knowledge of future job information since this is an online problem. It is assumed that the reward for processing each job is a constant. For any sequence of jobs, we seek to maximize the throughput of the set of machines. In other words, we seek to maximize the cardinality of the set of scheduled jobs.

Let the set of machines (active cameras) be denoted by $M = \{1, \ldots, m\}$. The set of jobs (people) is $J = \{1, \ldots, n\}$. Each job j is characterized by its release time and deadline vectors (R_j, D_j) where $R_j = \{r_{j,1}, \ldots, r_{j,m}\}$ and $D_j = \{d_{j,1}, \ldots, d_{j,m}\}$. There is restricted assignment of the jobs since each person will move through a subset of the cameras. Let $r_{j,i} = \infty$ and $d_{j,i} = \infty$ if job j will never be available for machine i. The global deadline for a job is defined as

$$d(j) = \max_{d_{j,i} \neq \infty} D_j. \tag{1}$$

This is the job's deadline for the last machine it can be processed on. If it is not processed by this time, it is lost to the entire surveillance network and that person leaves without being identified. Without loss of generality, we assume time is slotted as $t \in \{1, 2, \ldots\}$ and each job takes one time slot to be processed. At time t, the set of jobs available to machine i is represented by $J_t(i)$. This is all the people at time t that are in the camera's region of coverage that have not been identified. A schedule for machine i is a one-to-one mapping $t \rightarrow j \in J_t(i)$ if $J_t(i)$ is not an empty set.

This scheduling problem is different from traditional online scheduling problems because future release times and deadlines are not known. When a person enters the surveillance network, its trajectory is not generally known a priori. It can be predicted based on the person's heading, its path up to the current time, and past observed trajectories. A prediction can be as simple as assuming the person will continue to move in its current direction and at its current speed. The prediction could be more sophisticated and use a traffic model developed from the history of trajectories. Generally, the accuracy of the predictions decreases as the horizon of the prediction increases. To represent the uncertainty in the deadlines, we define $\tilde{d}_{j,i}(t)$ as the predicted deadline of person j at time t for camera i.

We have chosen to investigate heuristic scheduling approaches to this problem because of its complexity due to reliance on predicted deadlines and the spatial nature of the problem. If the solution of the scheduling problem did not depend on these predictions and the spatial dependencies between machines, the problem would be very similar to the ones considered by Bar-Noy et al. in [13]. They present constant factor approximation algorithms for four different variants of the online problem of scheduling jobs on multiple machines. An optimal offline algorithm must be developed for our problem if competitive analysis is to be used for evaluation. How best to develop this offline algorithm when the set of machines to which a job can be assigned is unknown a priori is an open question. Instead, we present heuristic scheduling algorithms and evaluate them empirically. These algorithms only use a person's deadline for the camera that can currently view the person rather than considering the global deadline that may be very inaccurate.

IV. Scheduling Policies

In this section, we describe two scheduling policies for scheduling active cameras for person identification. The first is a local policy that schedules jobs for each machine based on the estimated deadline of the jobs released to it. The job with the earliest deadline is scheduled first. The second policy uses a small amount of neighbor to neighbor communication in order to improve performance. This policy seeks to balance the load on the machines to achieve a higher system throughput. It will generally outperform the local scheduling policy when there is a congested area present in the surveillance network.

As mentioned, we have assumed the cameras have little or no overlap between their regions of coverage. This is done to remove the problem of coordinating which camera should identify a person that is available to two or more cameras at the same time. Either of the presented algorithms could be extended to involve an additional step of coordination to handle this. This would increase the amount of communication, but this communication would only be between neighbors.

A. Local Scheduling Policy

Earliest Deadline First (EDF) scheduling is a well-studied computationally efficient policy that performs well under varying traffic conditions when preemption is not allowed. Jackson showed in [14] that EDF is optimal for deadline scheduling without preemption on a single machine. For this problem, each machine (camera) runs this policy on the jobs currently available to it in J_t. The deadline used is the estimated local deadline $\tilde{d}_{j,i}(t)$. This is a local policy because the decision of which job to schedule is made independent of the other machines. No communication is required between the cameras to perform the scheduling. Predicting the deadlines requires basic knowledge of the physical extent of the camera's region of coverage.

Local EDF Scheduling Algorithm

Input: set of jobs J_t.

Sort J_t by $\tilde{d}_{j,i}(t)$

Output: the first element of the sorted set, J_t

As the reader will note, the algorithm has time complexity $\mathrm{O}(n \log n)$ where n is the number of jobs in J_t. The EDF algorithm does not consider how the decision of which person to schedule affects other active cameras in the future.

B. Distributed Scheduling Policy

In the design of a distributed scheduling policy, we consider the question of what gain in performance can be realized over local scheduling with the addition of a small amount of communication between cameras. Communication between neighbor nodes is preferred to other types of communication due to its lower cost in most wireless networks. The distributed scheduling policy that we devised is based on load balancing. In a traditional online load balancing problem, there are m parallel machines with jobs arriving at arbitrary times. A centralized controller is responsible for immediately assigning the incoming jobs to machines. The typical objective is to minimize the maximum load over the machines (see [15] for a survey of online load balancing problems). The choice of a load balancing approach is motivated by the realization that congested areas will decrease the throughput of the entire surveillance system. Relieving the active cameras in the congested area of some of their load should positively affect performance.

In this distributed load balancing (DLB) policy, local decisions are made by each camera to decrease the load of its neighbor with the highest load. Load is a measure of how many people are available for a camera to view. Load could be measured as the current number of people in a camera's region of coverage that have not been identified or by a time average. The concept of reducing the load of neighbors is relevant for this problem because of the underlying spatial configuration of the video surveillance network. An active camera can identify people headed towards an overloaded camera creating a more uniform distribution of the total load over the network. Maximum throughput is achieved when all the machines are busy, not when a few machines are overloaded.

A multi-class framework is used to encode load information for scheduling. Each camera periodically queries its neighbors for load information. For each available person, the camera predicts which neighbor node the person is most likely to visit next. That neighbor's load is used to assign the person to a prioritized class. The higher the load, the higher the priority of the class is. If a person is headed out of the surveillance network, it is considered to be moving towards a camera with infinite load and so is placed in the highest priority class. The rest of the classes are distributed over the remaining load levels.

Distributed Load Balancing Scheduling Algorithm

Input: set of jobs J_t.

Request load from neighboring nodes

For each $j \in J_t$
  Assign j to a class based on load of next
  predicted machine

Set K_t = all jobs of J_t belonging to highest
priority class

Sort K_t by $\tilde{d}_{j,i}(t)$

Output: the first element of the sorted set, K_t

The step of determining which jobs of J_t belong to the highest priority class has complexity $\mathrm{O}(2n)$ where n is the size of J_t. Selecting the job from K_t to schedule has complexity $\mathrm{O}(|K_t| \log |K_t|)$. If a traffic pattern causes congestion, DLB will generally be more computationally efficient than EDF since $|K_t|$ would be smaller than $|J_t|$.

The only communication required to implement this policy is neighbor to neighbor communication. Each camera transmits its load to all of its neighbors at specified intervals. Little bandwidth is used for coordination since the size of load information is small and the number of neighbors should not be large.

If the load is distributed relatively evenly over the network, DLB reduces to the EDF policy. If an active camera's neighbors all have a similar load, the people in that camera's region of coverage will be assigned to the same class. In this case, scheduling reduces to sorting by the local deadline just as in EDF. This will depend on the number of classes and how they are distributed over the load levels. Having too many classes will result in distinctions being created between load levels that should be considered equivalent.

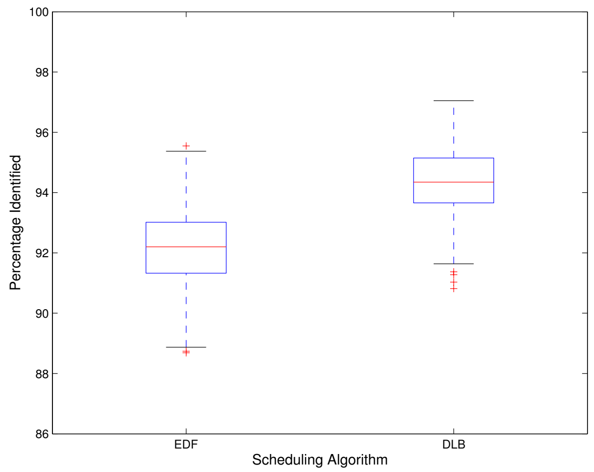

Fig. 3. Comparison of EDF and distributed load balancing for traffic with a congested area.

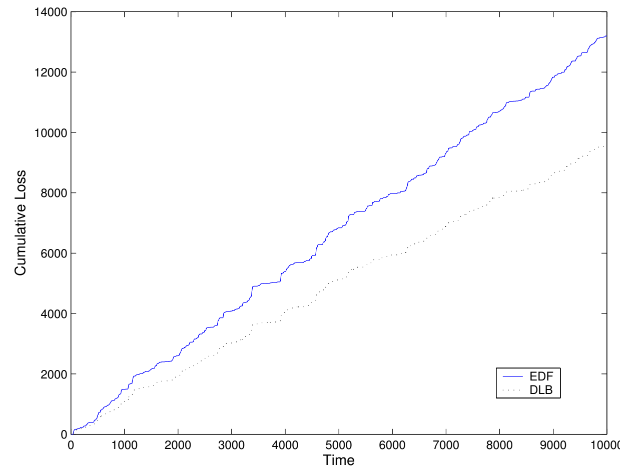

Fig. 4. Example cumulative loss plot for a single realization.

V. Empirical Results

In this section, we provide quantitative results of the performance of the EDF and DLB scheduling policies. Our results support the hypothesis that DLB achieves a higher throughput when there is congestion in an area of the surveillance network. This is the expected result because of the design of that algorithm. As noted by others [16], it can be difficult to randomly generate traffic for scheduling problems that is interesting. Often the traffic will be easily scheduled by almost any algorithm without loss of jobs or on the other extreme, cannot be scheduled without many jobs being lost. Meaningful analysis can occur when the traffic patterns are close to being schedulable without loss of jobs, but yet can cause scheduling algorithms to exhibit poor performance. We focus on high traffic experiments since there is little need for coordination in low traffic environments for this problem.

In our experiments, people are introduced into a simulated surveillance network according to a stochastic process. We use either a Poisson process model or a Markov-modulated Poisson process (MMPP). The MMPP provides more realistic traffic arrival patterns since it could model the burstiness one might expect in certain environments. The people move in a 2D plane among the cameras. The trajectories are determined by a mobility model. Mobility models are frequently used in ad-hoc networking research. [17] provides an overview of the different types of mobility models commonly used for simulation. For our experiments, a waypoint model is used for mobility. Initially, random waypoints were used, but this did not create interesting traffic patterns for analyzing these algorithms. The traffic was uniformly distributed over the surveillance network so the two algorithms exhibited the same performance as was explained in the previous section. Instead, we arbitrarily define paths through the surveillance network that could result in an area of congestion. The congestion resembled something similar to what would be expected if two or three sidewalks or roads crossed each other.

The active cameras are distributed according to a grid structure. Each camera has at most eight neighbors because of this structure. A person that completes its path through the surveillance network without being identified is considered lost. The cameras collect their neighbors' load after each time step of the simulation. This load information is used to partition the people in a camera's region of coverage into 4 classes for the DLB algorithm. This algorithm is not very sensitive to the number of classes as long as it is not on either extreme.

We tested the two scheduling algorithms against a series of arrival sequences. As a measure of performance, we used the percentage of people that the cameras observed. Figure 3 demonstrates the performance of EDF and DLB for 200 traffic sequences that exhibited a single area of congestion. While DLB provides better average performance, it should be noted that DLB does not dominate EDF. There exists arrival sequences such that EDF will have superior performance though these were rare in our experiments. In this experiment, the total amount of traffic is slightly below what the maximum load of the entire network would be if it were uniformly distributed. It is the congestion that causes less than 100% of the people to be identified.

The performance of the DLB algorithm partially depends on the accuracy of the prediction of the next camera a person will be passing by. As a person gets closer to leaving a camera's region of coverage, the prediction of which camera is next will generally become more reliable. If these predictions are consistently and significantly incorrect, it is possible for EDF to outperform DLB.

A single realization of the above experiment was used in creating Figure 4. This figures shows the number of jobs dropped over time. These dropped jobs correspond to people that have made their way through the surveillance without being identified. The overall shape of the two curves mirror each other. This is because the changes in slope are due to the arrival process switching from low to high traffic rates as determined by the Markov chain. For this particular realization, the DLB algorithm identified over 3000 more people than EDF did.

One limitation of our distributed approach is that each active camera only considers the load level of its neighbors when selecting a person to observe. Only the cameras that have neighbors in the congested area will make decisions to reduce this overload. The existence of a congested area is not communicated to the rest of the cameras. We believe that the approach presented in this paper can be extended to address this deficiency. Instead of a camera sharing its load with just its neighbors, the camera could maintain the average load of each neighbor. By computing a weighted average of its neighbors and itself and then sharing that as its load to it neighbors, areas of high congestion will affect more than just its neighbors. In doing this, a type of gradient field can be calculated that encodes where areas of congestion are in the surveillance network. Instead of predicting which neighbor a person is going to visit next, a camera would use the heading of the person to determine whether the person could contribute to an overloaded area in the future by using the load gradient in that direction. Deadlines could also be implicitly encoded in this gradient field since cameras on the edge of the surveillance network would have a large load under this scheme. Combining a traffic model with this gradient approach could have some benefits. A person's contribution to a congested area could be estimated by integrating over the different directions he or she may be headed weighted by the probability of heading in that direction derived from the traffic model. We have not had the opportunity to test this extension with an experiment.

One factor not consider in this paper is the orientation of the person to the camera. If the active cameras are seeking to capture high resolution images of people's faces, then the person needs to be facing the camera. For vehicles, the system might try to capture pictures of a vehicle's license plate. It is possible to encode the probability of capturing images that can be used for identification with the multi-class framework of the distributed load balancing algorithm. The classes would be determined by both the load of the predicted neighbor and the probability that an exploitable image is obtained. This is an interesting problem that needs further investigation.

VI. Conclusions

In this work, we have described a new distributed scheduling algorithm for the coordinated identification of people by a network of active cameras. This algorithm scales well as the size of the surveillance network increases. It uses only neighbor to neighbor communication and can be easily implemented to provide real-time performance. As our empirical results demonstrate, DLB will generally outperform EDF when congestion is present in an area in the surveillance network. There is some dependence on the reliability of the next predicted camera. The DLB algorithm is useful for situations when high traffic is expected in the surveillance network. It would be valuable to evaluate these algorithms on traffic sequences derived from real world observed traffic that exhibits congestion. A future extension under consideration is integrating the expected perspective of the person into the scheduling algorithm since this affects identification performance.

VII. Acknowledgments

This work was supported through IR&D funding provided by JHU/APL.

References

[1] Chris Stauffer and W. Eric L. Grimson, "Learning patterns of activity using real time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–767, 2000.

[2] Omar Javed and Mubrark Shah, "Tracking and object classification for automated surveillance," in *The seventh European Conference on Computer Vision*, Copenhagen, 2002.

[3] A. Senior, A. Hampapur, Y-L. Tian, L. Brown, S. Pankanti, and R. Bolle, "Appearance models for occlusion handling," in *Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.

[4] Chris P. Diehl and John B. Hampshire II, "Real-time object classification and novelty detection for collaborative video surveillance," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, 2002, vol. 3, pp. 2620–2625.

[5] J. Batista, P. Peixoto, and H. Araujo, "Real-time active visual surveillance by integrating peripheral motion detection with foveated tracking," in *Proceedings of the IEEE Workshop on Visual Surveillance*, 1998, pp. 18–25.

[6] D. Comaniciu, F. Berton, and V. Ramesh, "Adaptive resolution system for distributed surveillance," *Real Time Imaging*, vol. 8, no. 5, pp. 427–437, 2002.

[7] T. Darrell, B. Moghaddam, and A.P. Pentland, "Active face tracking and pose estimation in an interactive room," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1996, pp. 67–72.

[8] A. Hampapur, S. Pankanti, A.W. Senior, Y-L. Tian, L. Brown, and R. Bolle, "Face cataloger: Multi-scale imaging for relating identity to location," in *IEEE Conference on Advanced Video and Signal Based Surveillance*, Miami, FL, 2003.

[9] X. Zhou, R.T. Collins, T. Kanade, and P. Metes, "A master-slave system to acquire biometric imagery of humans at a distance," in *ACM SIGMM 2003 Workshop on Video Surveillance*, Berkeley, CA, 2003.

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

[11] M. Trivedi, K. Huang, and I. Mikic, "Intelligent environments and active camera networks," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2000, vol. 2, pp. 804–809.

[12] C.J. Costello, C.P. Diehl, A. Banerjee, and H. Fisher, "Scheduling an active camera to observe people," in *2nd ACM International Workshop on Video Surveillance and Sensor Networks*, New York City, 2004.

[13] Amotz Bar-Noy, Sudipto Guha, Joseph (Seffi) Naor, and Baruch Schieber, "Approximating the throughput of multiple machines in real-time scheduling," *SIAM Journal on Computing*, vol. 31, no. 2, pp. 331–352, 2001.

[14] J. R. Jackson, "Scheduling a production line to minimize maximum tardiness," Research Report 43, Management Science, University of California, Los Angeles, CA, 1955.

[15] Yossi Azar, Andrei Z. Broder, and Anna R. Karlin, "On-line load balancing," *Theoretical Computer Science*, vol. 130, no. 1, pp. 73–84, 1994.

[16] Hyeong Soo Chang, Robert Givan, and Edwin K. P. Chong, "On-line scheduling via sampling," in *5th International Conference on Artificial Intelligence Planning and Scheduling*, 2000, pp. 62–71.

[17] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing: Special issue on Mobile Ad Hoc Networking:Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.