Scheduling an Active Camera to Observe People

Cash J. Costello, Christopher P. Diehl, Amit Banerjee, Hesky Fisher
Applied Physics Laboratory
Johns Hopkins University
11100 Johns Hopkins Road
Laurel, MD 20723
{cash.costello; chris.diehl; amit.banerjee; hesky.fisher}@jhuapl.edu

ABSTRACT

Remote identification of people is an important capability for security systems. Automatically controlling a pan-tilt-zoom camera is an effective way to collect high resolution video or images of people in an unconstrained environment. Often there will be more people in an area than cameras available. The cameras must then divide their time among the people in order to view everyone. In this paper, we discuss the challenges involved in scheduling an active camera to observe multiple people. We present some candidate scheduling policies to address these challenges and evaluate their performance. The evaluation was conducted with a simulation based on data collected with our cooperative active camera system.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*motion, video analysis*; I.2.10 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Scheduling*; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Design, Experimentation, Performance

Keywords

Video Surveillance, Scheduling, Active Imaging

1. INTRODUCTION

A network of video surveillance cameras is an important asset for solving many security challenges such as facility monitoring and infrastructure protection. The cameras provide the video needed for tracking people, observing activities and identifying individuals. To fulfill these missions, the network must provide video at varying resolutions and perspectives. Recognizing a person requires high resolution views of the person of interest. Detecting and tracking people requires continual coverage of the scene that is best provided by video cameras with a wide field of view. In an unconstrained environment lacking choke points, it is difficult to achieve sufficient high resolution coverage to capture appropriate imagery of each person moving through the scene. One possible solution is to litter the area of interest with hundreds or thousands of cameras. Though this would ensure high resolution imagery could be captured of each person, it is not a practical solution.

A better solution is to coordinate a network of pan-tilt-zoom (PTZ) cameras to collect the necessary data. In this type of network, a certain number of cameras are allocated to provide wide area views. Video from these cameras is used to build an estimate of where people are in the monitored area. The remaining cameras are assigned to collect the data needed for identifying people. This is accomplished by panning, tilting and zooming a camera so that it can capture high resolution video of a particular person. By coordinating these two activities, the network can provide similar effectiveness to much larger deployments of video surveillance cameras.

As an initial demonstration of coordinated surveillance, we have developed a two camera system where the cameras cooperate to collect high resolution video of people moving through an area. Figure 1 shows the design of the system. The tracking camera provides a fixed wide field of view perspective of the scene. It detects moving objects, tracks them through the monitored area, and classifies them as being a person, group of people, or vehicle. By doing this, the tracking camera maintains an estimate of the location and movements of all the people in the scene. This information is provided to the second camera of the system, the active camera. Once it knows the location of a person, the active camera can pan, tilt, and zoom to collect high resolution video of that person. This data could be used to identify an individual through a manual or automatic process. There will generally be more than one person at a time in the monitored area so the camera must split its time between the people in the scene. The order that the active camera observes each person is controlled though a scheduling policy. The objective of the scheduling policy is to maximize the number of people that can be identified from the high resolution images collected.

The key contributions of this paper are a discussion of the challenges involved with scheduling the active camera and an analysis of possible scheduling policies. Related scheduling problems in the literature include serving network packets

VSSN'04, October 15, 2004, New York, New York, USA.

[6, 10] and scheduling jobs on a machine [1, 12]. An additional challenge faced in scheduling the active camera is the amount of uncertainty. The amount of time a person spends in the scene can be difficult to accurately estimate. It is also difficult to know when the observation task has been completed. We will explore these issues further along with presenting an evaluation of several scheduling policies.

The general problem of actively controlling PTZ cameras to improve 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 [2, 3, 4, 7, 16]. Systems for tracking and imaging more than one person are presented in [14, 15]. Many of these efforts focus on the problem of 3D tracking, pose estimation or actively tracking a person with the PTZ camera. Very little attention is given to the problem of what to do when there are more people in the scene than active cameras available. In [16], a single person is tracked by the active camera. If multiple people are present in the scene, the person who is closest to the position of the previous tracked individual is chosen. [7] mentions the problem of trying to decide which camera should be assigned to which person and offers some general approaches. It should also be noted that there is no work on objectively evaluating the performance of multi-camera systems for acquiring high resolution imagery of people. Most results are presented in the form of video examples or a series of screen captures.

In the following section, we discuss the problem of scheduling an active camera. Section 3 describes candidate scheduling approaches for this problem. The simulation framework created to evaluate scheduling performance is presented in Section 4. Section 5 contains a detailed discussion of the scheduling performance evaluation. We give a short description of the demonstration system in Section 6 followed by conclusions and possible future directions in Section 7.

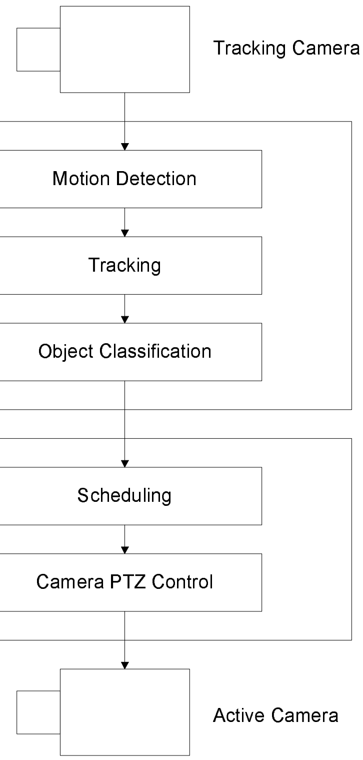

Figure 1: System Design

2. SCHEDULING PROBLEM

In our two camera surveillance system, the tracking camera provides updates on the state of the scene to the active camera. These state updates include the location and velocity of each person in the scene. This provides a picture of where people are located and where they could be going so that the active camera can collect high resolution images or video suitable for identification of people. The scheduling problem consists of deciding which person the active camera will focus its sensing resources on until the next state update. The arrival times of people entering the scene is not known a priori which makes this an online scheduling problem.

As mentioned in the introduction, a common, related problem in the literature is scheduling network packets to be served by a router. It is also an online problem since the arrival times of packets in the future are generally unknown. Each packet has a deadline that it must be served by or it is dropped. Also, each packet requires some amount of time to be served by the router. Both of these values are known when the packet arrives. In scheduling an active camera, a person corresponds to a packet. The camera must observe a person before the time he or she leaves the scene. That time serves as the deadline. The camera must also observe the person for some amount of time to capture the data required to perform identification.

An important difference between scheduling an active camera and serving packets is the amount of additional uncertainty in the problem. First, the deadline by which a person must be observed can only be estimated. It is not known upon a person's arrival in the scene due to the uncertainty of what path the person will take. Second, there is uncertainty over how long the observation task will take and when it is complete. For example, if the face is used for identification, a sudden turn of the head could prevent the camera from viewing the person from the needed perspective. Also, if there is not a real-time identification process, it is not possible to know when or if the observation was successful.

The objective of scheduling the active camera is to capture the data necessary for identifying as many individuals as possible. This objective can be broken into two competing sub-goals. One goal is to capture high resolution video of as many people as possible. The second goal is to view each person for as long or as many times as possible. To identify a person, the active camera at a minimum has to capture a single high resolution image of the person. The probability of identifying a person generally increases as the amount of data collected by the active camera of that person increases. The trade-off is that the more time that is spent viewing a particular person, the fewer people that can be observed overall. In one extreme, the camera would select an individual and follow that person along his or her entire path through the scene. This maximizes the odds of identifying that particular person, but all other people walking through the scene would go unobserved by the active camera. In the other extreme, the camera would observe a person for only a single video frame. If it has already looked at everyone in the scene, it would cycle back through to look at everyone again. This approach ensures that the active camera observes many people, but it has a serious drawback. There is a transition cost measured in time to be paid whenever the camera switches from person to person. This is because cameras cannot instantaneously change their pan and tilt angles or adjust their zoom lenses and focus. At this extreme, the camera spends almost all its time panning, tilting and zooming rather than collecting video of people.

In the design of any scheduling policy, a compromise has to be made between these two goals. The solution will depend on factors such as the physical limitations of the cameras and the expected amount of traffic in the scene. Generally, the solution will be much closer to the goal of viewing as many people as possible. For this surveillance system, the active camera spends a short period of time looking at a person before switching to the next one. This allows it the opportunity to observe each person several times if there is not much traffic while still being able to view many people if there is a surge in traffic. The time period length depends on the traffic currently in the monitored area. There is a minimum length of time the camera will spend watching an individual for each observation that was determined empirically. It is on the order of a few seconds. This enables the system to be very responsive to changes in pedestrian traffic levels.

3. SCHEDULING POLICIES

Because the active camera will typically observe each person multiple times, we have posed camera scheduling as a multi-class scheduling problem. Class assignments are made based on the number of times each person has been observed by the active camera. This multi-class structure could also be used to encode information about the importance of identifying a particular person. For example, a person who enters a restricted area or is behaving abnormally could be assigned to a high priority class to ensure he or she is identified. Also, the expected perspective of the person relative to the camera could be incorporated into the class assignments. This multi-class framework for scheduling provides much flexibility.

We have restricted this evaluation to only include greedy scheduling policies. This is for two reasons. First, as scheduling must be performed in real-time, computational complexity is an important constraint. Second, because the environment can be so unpredictable, trying to find a close to optimal schedule over a longer horizon yields little benefit and can actually decrease performance.

Two different types of greedy policies are considered: static priority and non-static priority. With static priority, every member of the highest priority class is scheduled before other classes are considered. For this problem, it means the active camera must view every person once before observing someone a second time. Sometimes this is referred to as round robin scheduling. With non-static priority policies, a person from a lower priority class will sometimes be selected first. For example, let there be two people in a monitored area. One person has not been viewed and so belongs to the highest priority class. That person is not expected to leave the area soon. The other person has been viewed but is close to exiting the scene. This type of policy might schedule the person just about to leave for a second look before observing the other person. Both approaches are greedy, but they have different criteria.

3.1 Static Priority Policies

A static priority policy is a multi-class scheduling policy that selects from the highest priority class. The different varieties presented differ in the criterion used to break ties when there is more than one person in a class. The + added to the policy abbreviations denotes that they are static priority versions of the more common policies.

Random

This is the simplest policy, as it does not use any tie breaking criteria, but instead selects an arbitrary person from the highest priority class.

First Come, First Serve (FCFS+)

This policy breaks ties by selecting the person who entered the scene first. The arrival times of people are maintained by the tracking camera and provided to the active camera. This approach to scheduling is one of the easiest to implement.

Earliest Deadline First (EDF+)

This policy breaks ties by selecting the person who is expected to leave the scene first. The deadline for each person can be estimated in several ways. A deadline could be calculated by projecting a person forward along his or her current trajectory. As long as the person does not significantly change his speed or direction, this provides a good estimate. Remember that this estimate is only used to make the next selection and is updated as the tracking camera provides new information. A more involved estimate of the deadline would use probabilistic modeling of the traffic in the scene. From this model, an expected value of the deadline could be calculated given the person's path up to this instant.

3.2 Non-Static Priority Policies

Many policies fit in this category. The more traditional versions of the first come, first serve and earliest deadline first are two examples. With the common earliest deadline first policy, people would be scheduled based solely on their expected deadlines. The policy could be modified to work in this multi-class formulation of the problem by removing a person from consideration after that person has been observed a certain number of times. We chose to include a single policy from this category.

Current Minloss Throughput Optimal (CMTO)

A greedy scheduling policy for serving packets from different classes is presented in [6]. Each class of packets has an associated weight or reward value. The objective of a scheduling policy in this context is to minimize the total weighted loss due to dropped packets. The authors prove that the CMTO policy dominates a multi-class extension to earliest deadline first scheduling under this measure. The earliest deadline first policy from this paper is not the same policy as considered here. The authors also prove that there is no optimal online policy for the multi-class scheduling problem under this error measure.

Continuing to use network terminology, the CMTO policy selects the packet that will result in minimizing the expected weighted loss given the current list of packets in the queue. It does this by looking forward in time to find the first point when it will have to drop a packet based on the deadlines of the current packets. This point in time serves as a scheduling horizon and is referred to as the cut. CMTO then creates a list of the highest weighted packets that have deadlines before or at this cut that could be served in the available time. A packet is selected from this list that will result in the most weight being served by the cut given the current packet queue. This is different from static priority policies that always seek to serve the packet that will result in the most weight served in the next time step. CMTO instead looks ahead in time to the cut. Sometimes this will result in a lower weight packet being served because it has an imminent deadline rather than a higher weight packet with a later deadline. Two different algorithm implementations of this policy are included in [6]. A major difference between the problem of serving packets and observing people is that after a person has been observed that person is still eligible to be scheduled as a member of a lower priority class.

4. SCHEDULING SIMULATION

Evaluating different scheduling policies using an actual video surveillance system is a very complicated task. First, it is extremely difficult to evaluate each algorithm on the same scenarios. Pre-recorded video can be used for the tracking camera, but not easily for the active camera. Scripting scenarios is not a good solution for this either. It is difficult to get actors to walk the same path and at the same speed for each iteration. Second, scripted scenarios do not provide enough diversity in the types of pedestrian traffic one can obtain. To truly evaluate a scheduling policy, a wide range of traffic levels and paths through the scene need to be used. Third, it is difficult to derive statistics about the performance of the different scheduling policy implementations by observing the system. For example, errors could be due to a mistake performed by the tracking camera rather than a weakness of the scheduling policy. A person could have been classified as a vehicle or the tracker might have failed to track a person through an occlusion. It is not easy to separate the performance of the scheduling policy from that of the overall system. Also, calculating statistics such as the percentage of people viewed by the active camera requires tremendous amounts of time and effort.

To address these difficulties, we have created a Monte Carlo simulation for evaluating scheduling policy implementations. This allows us to repeat the same experiments for each scheduling policy. Because it is a simulation, the locations and velocities of people provided to the active camera are ground truth. Any missed person is due solely to the scheduling approach. Additionally, doing evaluation through simulation provides control over parameters such as the amount of traffic or the location of cameras. Additional active cameras can also be added to simulate a network with multiple cameras.

The simulation consists of three main parts. There is the arrival process component. This component determines when people enter the simulated scene and is usually modeled as a Poisson process. The second piece is the mobility model that controls how people move about the scene. The mobility model can range from a simple random waypoint model to using tracks captured from a real surveillance system. The tracking information from this simulated scene is provided to the scheduler of an active camera, the third component. This is the same type of information that is provided to the active camera in our demonstration system.

Figure 2: Example Tracks
These six tracks are a small subset of the tracks captured by the system and stored in a database.

To help in producing realistic scenarios, we have collected tens of thousands of tracks observed by the tracking camera of our surveillance system. The area being monitored by the cameras is a paved area nearby buildings. There is both vehicular and pedestrian traffic. The pedestrian traffic is not constrained and there are several common paths people take through the scene to reach their destinations. For examples of the tracks captured by the tracking camera, see Figure 2.

5. POLICY EVALUATION

Using this simulation, the scheduling policies were evaluated in two ways related to the two competing sub-goals mentioned earlier. First, one metric of performance is how many people could possibly be identified from the data collected by the active camera. This corresponds to the percentage of people who passed through the scene and were observed at least once. The second evaluation criterion is the number of people viewed multiple times. The number of times a person is viewed on average is an alternative way of evaluating this second criterion. People who are observed multiple times by the active camera will generally be easier to identify. It allows the system to recover from problems such as a temporary occlusion. It can also result in different perspectives being taken of a person, which can improve the chances of identification.

For the first simulation, the length of each scenario is one thousand time steps. The policies are run against one thousand scenarios. The simulated people move through the area according to tracks captured from the demonstration system. Each scenario averages around eight hundred people walking through the scene. People enter the scene according to a Poisson process. Figure 3 is a box plot of the percentage of people observed by the camera for the different policy implementations. The static priority algorithms outperform the CMTO policy under this measure of performance. Of the three static priority algorithms, EDF+ is the best performer.

There are two primary reasons that the static priority approaches outperformed CMTO in the first simulation. First of all, as previously mentioned, the problem as posed is not a true multi-class scheduling problem. Every person arrives in

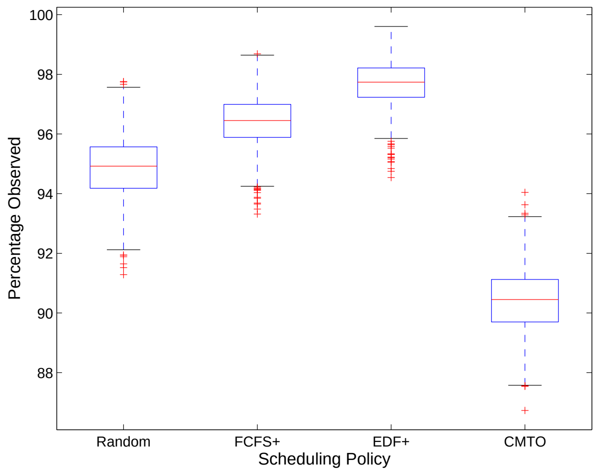

Figure 3: Policy Performance for Captured Traffic

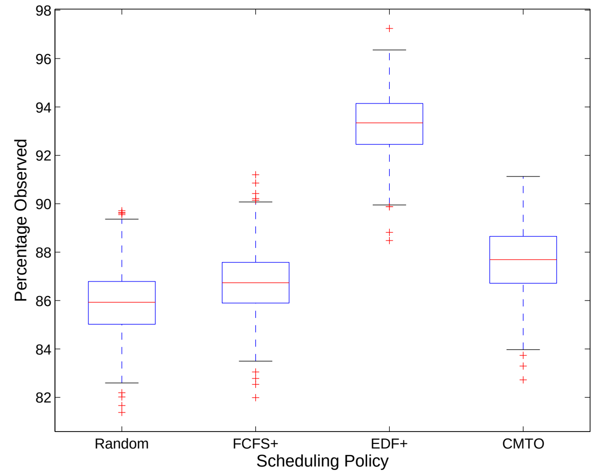

Figure 4: Policy Performance for Predictable Traffic

the scene as a member of the highest priority class. When a person is observed, instead of removing the person from the queue, the person remains to be scheduled as a member of a lower priority class. This violates an implicit assumption of the CMTO policy. The result is that the cut or scheduling horizon does not have the same value as with a traditional multi-class scheduling problem. Secondly, CMTO is very dependent on deadline accuracy. The estimated deadlines are used to calculate the cut and in selecting which person to observe. In this simulation, the deadlines were estimated by determining when the person would depart the scene if he or she continued moving in the same direction and at the same speed. These deadline estimates can be very error prone given the realistic nature of the movements of people in the simulation. A person can suddenly change directions or stop to talk to someone or get into a vehicle. These reasons and many others can make the deadlines unreliable. This demonstrates that policies that greatly depend on predictions of the future will suffer significant performance loss with poor predictions.

Of the static priority policies, EDF+ performed the best. In fact, it can be shown that under certain conditions this policy is the optimal online policy under this performance measure. The conditions are that the deadlines are accurate and each observation takes a single unit of time. See [8] for the proof of this policy's optimality. In scheduling terminology, EDF+ is referred to as throughput optimal. This means it will observe as many people as possible at least once. It outperforms the other two approaches even though the deadlines are estimates that are often inaccurate. Ordering by these estimates still provides more value than using arrival times. As is expected, scheduling based on the arrival times of people is more productive than randomly selecting a person to observe. The longer a person has been in the scene, the more likely it is that the person will leave soon. If every person took a constant amount of time to transit the scene, FCFS+ and EDF+ would be equivalent.

An interesting question is how the accuracy of deadline estimates affects policy performance. Both EDF+ and CMTO depend on these deadline predictions. EDF+ performed well on the real tracks used in the first simulation. CMTO did not perform well on these tracks in comparison with the static priority policies. In Figure 4, results are presented for scenarios that contain very simple predictable tracks. In these scenarios, the paths taken by people are all straight lines through the scene and people move at constant speeds. Therefore, the predicted deadlines are correct for this simulation. As expected, EDF+ and CMTO improve their performance relative to the other policies. CMTO now outperforms the Random policy and FCFS+. EDF+ improves from 1.3% better than FCFS+ to 7.6% better. From this, we conclude that scenes with traffic that is predictable are good matches for scheduling approaches like EDF+. This also points to the possibility of traffic modeling for improving scheduling performance. It is likely that if a traffic model is used in the first simulation, the performance of EDF+ and CMTO will improve.

It is worth noting that results cannot be compared across simulations with different mobility models. For example, even though the traffic is more predictable in the second simulation, the overall performance of all the scheduling approaches is lower. Many of the random straight paths through the scene were very short so this decreased the performance of all the policies. This demonstrates a general rule about scheduling policies: performance is highly dependent on the type of traffic in the scene.

The rate that people enter the scene also affects the expected performance of the scheduling policy implementations. To do an analysis of this, the Poisson process rate was varied in several simulations like the first one. The results are in Figure 5. The ordering of the different scheduling policies does not change. EDF+ is consistently the best policy. As traffic increases, the differences between the policies decrease. Eventually, as the amount of traffic increases it does not matter what policy is implemented. In fact, at very high traffic rates, it is more effective to use an active camera to scan a crowd collecting images than schedule it to observe individual people. This type of analysis could be useful for determining the number of cameras needed to monitor an area in multi-camera systems.

The second criterion used for evaluation is the number of people observed multiple times. See Figure 6 for a comparison of the approaches using this measure. This comparison used data from the first simulation. EDF+, which performs

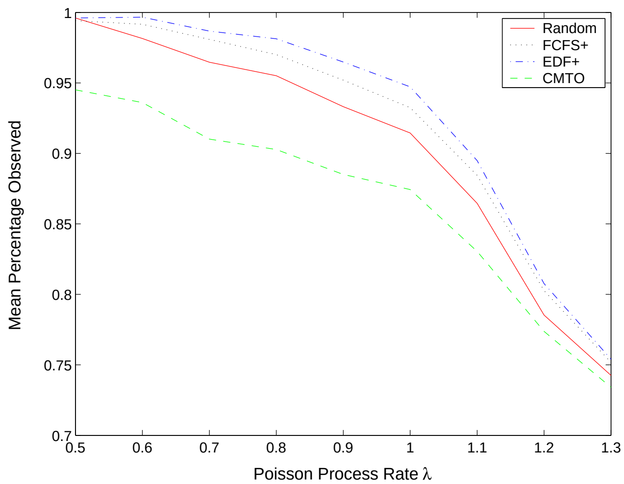

Figure 5: Policy Performance Versus Arrival Rate

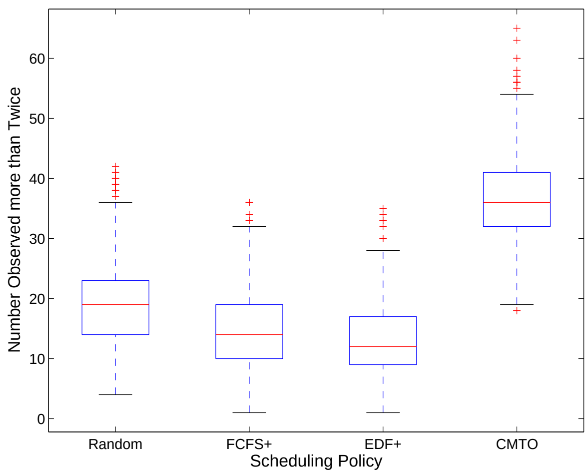

Figure 6: People Observed Multiple Times

best under the first criterion, is the worst under this one. CMTO performed twice as well as the second best policy. This fits with the expectations of this policy. Though not always the case, the policies performed in the opposite order under this measure compared to the first one. In general, approaches that perform well under one measure will not perform as well under the other. The challenge is to determine how to balance these two competing goals to increase the overall system performance.

6. DEMONSTRATION SYSTEM

As shown in Figure 1, the demonstration video surveillance system consists of two cameras: the tracking camera and the active camera. The tracking camera detects moving objects using a multi-modal background subtraction method based on [13]. A multi-hypothesis blob tracker tracks the moving objects through the scene. It is based on the work of [9] and [11]. A cascade of support vector machines is used to classify the moving objects as either person, group of people, or vehicle. The classifiers were trained on data collected from the system. See [5] for a description of designing real-time object classifiers via machine learning. The active camera receives the locations and velocities of people in the scene from the tracking camera. The correspondence between points in the tracking camera's static field of view and the active camera's field of regard was established manually with a calibration target. From the list of people, the active camera selects a person to observe using one of the greedy scheduling policies presented earlier. Images of the person of interest are segmented out of the video stream using optical flow. A screen shot of the system in operation is in Figure 7.

Figure 7: Screen Shot of System
This is a screen shot of the system in operation. The person highlighted in red is the person scheduled to be imaged next.

7. CONCLUSIONS AND FUTURE WORK

Automated high resolution imaging of individuals using PTZ cameras is an important capability for many security challenges. Many times the cameras must divide their time among multiple people to view everyone in an area. We have shown that for a single active camera, scheduling the order people are observed adds value. It is important to be aware that selecting a scheduling approach depends on the expected traffic in the monitored area. Policies that depend on complicated predictions about the future can quickly degrade in performance if these predictions are poor.

In the future, we intend to present results related to scheduling multiple active cameras. There are additional challenges with multiple cameras such as having to schedule not only when to view a particular person but also determine which camera should be assigned to which person. Scalability becomes an issue if the number of cameras is large. To address this, we will be looking into both centralized and distributed approaches to the problem. Also, traffic modeling appears to be a good avenue for improving scheduling performance.

8. ACKNOWLEDGMENTS

This work was supported through IR&D funding from JHU/APL. The authors would also like to acknowledge the efforts of Salvatore Guarnieri in the creation of the demonstration system.

9. REFERENCES

[1] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber. Approximating the throughput of multiple machines in real-time scheduling. *SIAM Journal on Computing*, 31(2):331–352, 2002.

[2] 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*, pages 18–25, 1998.

[3] D. Comaniciu, F. Berton, and V. Ramesh. Adaptive resolution system for distributed surveillance. *Real Time Imaging*, 8(5):427–437, 2002.

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

[5] C. P. Diehl and J. 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*, volume 3, pages 2620–2625, 2002.

[6] R. L. Givan, E. K. P. Chong, and H. S. Chang. Scheduling multiclass packet streams to minimize weighted loss. *Queueing Syst. Theory Appl.*, 41(3):241–270, 2002.

[7] A. Hampapur, S. Pankanti, A. 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.

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

[9] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *The seventh European Conference on Computer Vision*, Copenhagen, 2002.

[10] T. Ling and N. Shroff. Scheduling real-time traffic in atm networks. In *Proc. of IEEE INFOCOM*, pages 198–205, 1996.

[11] 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.

[12] J. Sgall. On-line scheduling – a survey. In A. Fiat and G. Woeginger, editors, *On-Line Algorithms: The State of the Art*, Lecture Notes in Computer Science, pages 196–231. Springer-Verlag, 1998.

[13] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–767, 2000.

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

[15] 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*, volume 2, pages 804–809, 2000.

[16] X. Zhou, R. 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.