

# Systems and Algorithms for Autonomously Simultaneous Observation of Multiple Objects Using Robotic PTZ Cameras Assisted by a Wide-Angle Camera

Yiliang Xu and Dezhen Song

**Abstract**— We report an autonomous observation system with multiple pan-tilt-zoom (PTZ) cameras assisted by a fixed wide-angle camera. The wide-angle camera provides large but low resolution coverage and detects and tracks all moving objects in the scene. Based on the output of the wide-angle camera, the system generates spatiotemporal observation requests for each moving object, which are candidates for close-up views using PTZ cameras. Due to the fact that there are usually much more objects than the number of PTZ cameras, the system first assigns a subset of the requests/objects to each PTZ camera. The PTZ cameras then select the parameter settings that best satisfy the assigned competing requests to provide high resolution views of the moving objects. We solve the request assignment and the camera parameter selection problems in real time. The effectiveness of the proposed system is validated in comparison with an existing work using simulation. The simulation results show that in heavy traffic scenarios, our algorithm increases the number of observed objects by over 200%.

## I. INTRODUCTION

Consider a wide-angle camera installed at an airport for human activity surveillance or in a forest for wildlife observation. The wide-angle camera can provide large, low resolution coverage of the scene. However, recognition and identification of humans and animals usually require close-up views at high resolution which need PTZ cameras. The resulting autonomous observation system consists of a fixed wide-angle camera with multiple PTZ cameras as illustrated in Figure 1. The wide-angle camera monitors the entire field to detect and track all moving objects. Each PTZ camera selectively covers a subset of the objects.

However there are usually more moving objects than the number of PTZ cameras. With these competing spatiotemporal observation requests, the major challenge is the control and scheduling of the PTZ cameras to maximize the “satisfaction” to the competing requests. The system design emphasizes the “satisfaction” to the requests which takes into account 1) the camera coverage over objects, 2) camera zoom level selection, and 3) camera traveling time. We approach the control and scheduling problem in two steps. In the first step, a subset of the requests/objects is assigned to each PTZ camera. In the second step, each PTZ camera selects its PTZ parameters to cover the assigned objects. We formulate the problems in both steps and solved them in real

time. We implemented the system and conducted numerical simulations. The experiment results show that our method outperforms an existing work by increasing the number of observed objects by over 200% in heavy traffic scenarios.

## II. RELATED WORK

The proposed autonomous observation system relates to the existing works on active video surveillance systems and the frame selection problem.

In the recent decade, multiple camera surveillance systems, especially those with both static and active cameras have attracted growing attention of research. Most of the works are master-slave camera configuration [1]. The master static camera(s) provide the general information about the wide-angle scene while the slave active cameras acquire the localized high-resolution imagery of the regions of interest. This is a relatively new research area with many directions to explore. A very recent live system in this category can be found in [2]. Our work belongs to this category.

Most works in this category schedule the active cameras based on simple heuristic rules. Zhou et al. [1] choose the object closest to the current camera setting as the next observation object. Hampapur et al. [3] adopt the simple round robin sampling. Bodor et al. [4] and Fiore et al. [5] propose a dual-camera system with one wide-angle static camera and a PTZ camera for pedestrian surveillance. Human activities (walking, running, etc.) are prioritized based on the preliminary recognition by the wide-angle camera. The PTZ camera focuses to the activity with the highest priority for further analysis. Costello et al. [6] are the first to formulate the single camera scheduling problem based on network packet scheduling literatures. The authors propose and compare several greedy scheduling policies. With different assumptions towards the observation scene and objects, various scheduling formulation and schemes are proposed. In Lim et al. [7], the scheduling problem is formulated as a graph matching problem. Bimbo and Pernici [8] truncate the continuous scheduling problem by a predefined observation deadline and each truncated camera scheduling problem is formulated as an online dynamic vehicle routing problem (DVRP). However these methods assign only one object to one active camera. Our system assigns multiple objects to individual cameras by selecting PTZ camera parameters such that the camera coverage-resolution tradeoff is achieved. This also enables group watching which is very meaningful in many applications.

This work was supported in part by the National Science Foundation under IIS-0534848 and 0643298, and in part by Microsoft.

Y. Xu and D. Song are with Computer Science and Engineering Department, Texas A&M University, College Station, TX 77843. Email: ylxu@cse.tamu.edu and dzsong@cse.tamu.edu.

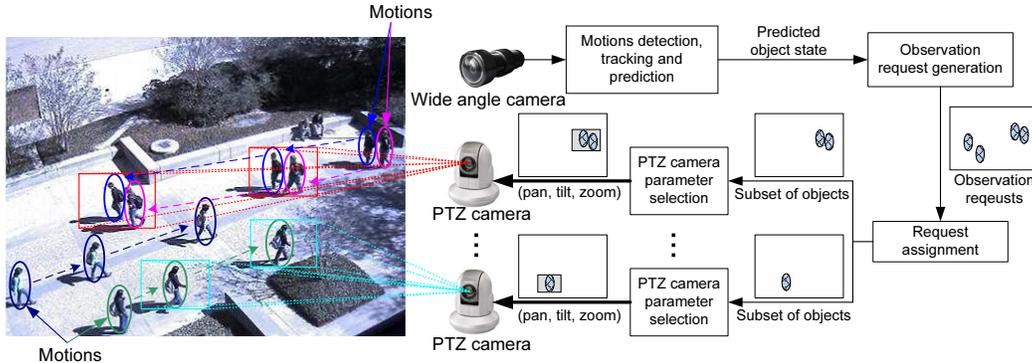


Fig. 1. System architecture.

Very few work considers the selection of the zoom level of active cameras and assigns multiple objects to individual cameras. Lim et al. [9] construct the observation task for each single object as a “task visibility interval” (TVI) based on its predicted states and corresponding camera settings. When TVIs have non-empty intersection, they are grouped to form a “multiple task visibility interval” (MTVI). Based on the order of the starting time of (M)TVIs, a directed acyclic graph (DAG) is constructed. The scheduling problem is formulated as a maximal flow problem. A greedy algorithm and a dynamic programming scheme are proposed to solve it. Zhang et al. [10] construct a semantic saliency map to indicate the observation requests. An exhaustive algorithm finds the optimal single frame that minimizes the information loss. Sommerlade and Reid [11] use an information-theoretic framework to study how to select a single active camera’s zoom level for tracking single object so as to balance the chances of losing the tracked object and that of losing trace of other objects. In contrast to these works, our scheduling does not require accurate motion prediction for the entire duration of objects in the FOV as in [9]. The assignment of multiple objects to individual PTZ cameras is carried out by selecting the camera parameters to achieve the tradeoff between coverage and resolution.

Our group focuses on developing intelligent vision systems and algorithms using robotic cameras for a variety of applications such as construction monitoring, distance learning, panorama construction and natural observation [12]. In the context of using PTZ camera for the collaborative observation, competing observation requests need to be covered by camera frame(s) to maximize the overall observation reward. This issue is formulated as the frame selection problem [13]. A series of algorithms for single frame selection (SFS) problem have been proposed [13], [14]. Song et al. [15] propose an autonomous observation system in which a single PTZ camera is used to fulfill competing spatiotemporal observation requests. In this work, multiple PTZ cameras are used to increase the observation coverage. Recently, an approximation algorithm for the multi-frame selection problem is proposed [16]. The algorithm coordinates  $p$  ( $p \geq 1$ ) camera frames to cover  $n$  ( $n \geq p$ ) competing obser-

vation requests in  $O(n/\epsilon^3 + p^2/\epsilon^6)$  time, where  $\epsilon$  is the approximation bound. This algorithm inspires the direction of simultaneous multi-object observation using multiple PTZ cameras as in this work.

### III. SYSTEM ARCHITECTURE AND TIMELINE

Figure 1 shows the architecture of the system. The system consists of  $p$  ( $p \geq 1$ ) PTZ cameras and a wide-angle camera. All cameras are calibrated. The wide-angle camera detects and labels all moving objects in the scene. The states of the objects (e.g., size, position and velocity) in the 2D image space are tracked and predicted. Based on the prediction, the observation request generation module generates the competing spatiotemporal observation requests (shadowed ellipses) for all objects. Then the request assignment module assigns a subset of the objects/requests to each PTZ camera by computing the  $p$ -frame settings that best satisfy the requests. Each PTZ camera tracks the objects assigned to it by selecting the PTZ parameter settings that best satisfy these requests to capture high resolution images/videos of the objects.

Figure 2 shows the timeline of the system. An observation cycle starts at time  $t = t_0$ . The states of the objects at time  $t = t_0 + \delta_l$  are predicted, where  $\delta_l$  is termed as “lead time”. Based on the predicted states, the system generates the observation request at time  $t = t_0 + \delta_l$  for each object. A subset of these objects is then assigned to each PTZ camera. Then the system starts to adjust the PTZ cameras according to the request assignment. The camera traveling time is bounded by the “lead time”  $\delta_l$  so that the cameras intercept the objects at time  $t = t_0 + \delta_l$ . After that, each PTZ camera tracks its object subset for time  $\delta_r$  until the beginning of the next observation cycle.  $\delta_r$  is termed as “recording time” and is evenly divided into  $n_r$  intervals with each of length  $\tau$ . Based on the state prediction, the PTZ camera parameter selection module computes each camera’s setting at the end of each interval. Then each camera micro-adjusts its settings for up to  $\tau$  time and prepares for the next interval. By capturing images/videos for  $\delta_r$  time, the request assignment module re-initiates and the operations above repeat.  $T = \delta_l + \delta_r$  is called one observation cycle.

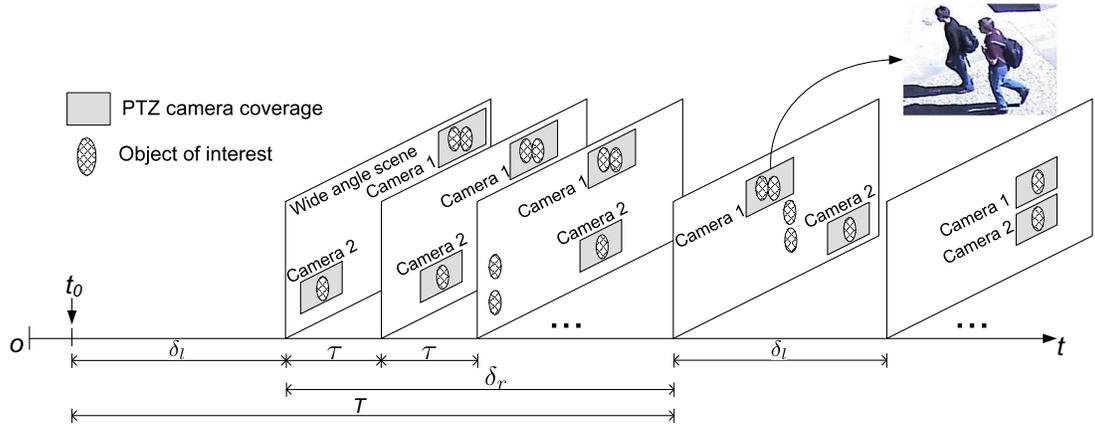


Fig. 2. System timeline. An observation cycle starts at  $t = t_0$ . Within each cycle time  $T = \delta_l + \delta_r$ , all PTZ cameras first take no more than  $\delta_l$  time to adjust the PTZ parameters based on the request assignment. Then each PTZ camera micro-adjusts its parameters within interval  $\tau$  to track the assigned subset of objects. This tracking lasts  $\delta_r$  time until a new observation cycle starts.

#### IV. CAMERA SCHEDULING ALGORITHM

For  $p$  PTZ cameras, there are usually much more objects. With the competing spatiotemporal requests, we need to control and schedule the PTZ cameras to capture sequences of images/videos that best satisfy the requests. Frame selection and camera scheduling module is developed for this purpose.

##### A. Observation request generation

The wide-angle camera detects all motions and tracks them continuously. Each object is represented by an iso-oriented elliptic region which is determined by a 4-parameter vector,

$$[u, v, a, b]^T, \quad (1)$$

where  $(u, v)$  indicates the center of the ellipse in the image space;  $a$  and  $b$  denote the two axes of the ellipse, respectively. Thus the state of the object at time  $t$  can be represented by

$$x(t) = [u(t), v(t), a(t), b(t), \dot{u}(t), \dot{v}(t)]^T, \quad (2)$$

where  $(\dot{u}(t), \dot{v}(t))$  indicates the velocity of the ellipse center in the image space at time  $t$ .

A non-parametric Gaussian background subtraction model [17] is used to detect and label any moving objects. A kernel-based mean-shift [18] algorithm is used to track the segmented objects. For predicting the object state, each labeled object is assigned a Kalman filter. A commonly used constant velocity model is adopted. Kalman filter is also able to handle short-term occlusion by predicting the object motion. It is worth mentioning that a lot of existing tracking algorithms [19] can be applied here.

Given the predicted state of  $i$ -th object at time  $t$  is

$$\hat{x}_i(t) = [\hat{u}_i(t), \hat{v}_i(t), \hat{a}_i(t), \hat{b}_i(t), \hat{\dot{u}}_i(t), \hat{\dot{v}}_i(t)]^T,$$

we define the spatiotemporal observation request as,

$$r_i(t) = [\hat{u}_i(t), \hat{v}_i(t), \hat{a}_i(t), \hat{b}_i(t), z_i, \omega_i(t)]^T, \quad (3)$$

where  $\hat{u}(t)$ ,  $\hat{v}(t)$ ,  $\hat{a}_i(t)$  and  $\hat{b}_i(t)$  define the desired rectangular requested region in the same way as  $u$ ,  $v$ ,  $a$  and  $b$  in (1);  $z_i$  indicates the desirable resolution and  $\omega_i(t)$  is the temporal

weight, which indicates the emergency/importance level of the  $i$ -th object at time  $t$ .  $\omega_i(t)$  plays an important role in balancing the observation service across all the objects and will be discussed in details later.

##### B. Request assignment

As shown in Figure 2, at the beginning of each recording time  $\delta_r$ , we need to coordinate  $p$  PTZ cameras so that each camera is assigned a subset of the objects. We choose the  $p$ -frame settings that best satisfy all the requests at that time. In our system, the PTZ camera setting is parameterized by a 3-vector,

$$c = [x, y, z]^T,$$

where  $(x, y)$  is the center point of the camera frame, which essentially indicates pan and tilt settings;  $z$  is the resolution of the frame. With a fixed aspect ratio (e.g., 4:3),  $z$  also determines the size of the frame.

The ‘‘satisfaction’’ to the observation request is quantified by a metric. We extend the Resolution Ratio with Non-partial Coverage (RRNPC) metric in [16] to cope with the spatiotemporal requests. Given a request  $r_i(t)$  and a frame  $c$ , we derive the definition of the satisfaction function as,

$$s(c, r_i(t)) = \omega_i(t) \cdot I(c, r_i(t)) \cdot \min\left(\frac{z_i}{z}, 1\right), \quad (4)$$

where  $I(c, r_i(t))$  is an indicator function,

$$I(c, r_i(t)) = \begin{cases} 1 & \text{if } r_i(t) \subseteq c, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

The term  $\min(\frac{z_i}{z}, 1)$  indicates the resolution ratio. It reaches the maximum of 1 when the resolution level of the camera frame is better than that of the request. In (5) we abuse the set operator  $\subseteq$  in the way  $r_i(t) \subseteq c$  indicates that the requested region is fully contained in that of the frame. This means we do not accept partial coverage over the request. This is necessary for many purposes such as object recognition and identification. To maximize the overall coverage of the  $p$  frames, we also restrict that any two camera

frames do not fully contain a request region in common. This constraint also avoids multiple count for one request. Therefore, the overall satisfaction of a  $p$ -frame set  $C^p(t) = \{c_1(t), c_2(t), \dots, c_p(t)\}$  over  $n$  requests is the sum of the satisfaction to each individual request  $r_i(t), i = 1, 2, \dots, n$ ,

$$\begin{aligned} s(C^p(t)) &= \sum_{i=1}^n \sum_{u=1}^p s(c_u(t), r_i(t)) \\ &= \sum_{i=1}^n \sum_{u=1}^p \omega_i(t) \cdot I(c_u(t), r_i(t)) \cdot \min\left(\frac{z_i}{z_u(t)}, 1\right). \end{aligned} \quad (6)$$

Thus the request assignment problem is formulated as finding the optimal  $p$ -frame settings that maximizes the overall satisfaction,

$$C^{p*}(t) = \arg \max_{C^p(t)} s(C^p(t)). \quad (7)$$

This problem can be solved in [16] with running time  $O(n/\epsilon^3 + p^2/\epsilon^6)$ , where  $\epsilon$  is the approximation bound. After assigning the requests by finding the optimal  $p$  frame settings, we find the best camera-setting pairs that minimize the time for adjusting the PTZ cameras.

We summarize the request assignment scheme in Algorithm 1. We assume the states of the objects can be predicted trivially ahead of time. This is usually true for Kalman filter predictor.

---

**Algorithm 1:** Request Assignment (RA)

---

- Input:** Current time  $\sigma$ ; predicted object states at time  $\xi$ , ( $\xi \geq \sigma + \delta_i$ ),  $\hat{X}(\xi) = \{\hat{x}_1(\xi), \hat{x}_2(\xi), \dots, \hat{x}_n(\xi)\}$ .
- Output:**  $p$ -frame settings  $C^{p*}(\xi) = \{c_1^*(\xi), c_2^*(\xi), \dots, c_p^*(\xi)\}$ , with  $i$ -th camera being assigned an object subset.
- 1 Generate requests at time  $t$ ,  $R(\xi) = \{r_1(\xi), r_2(\xi), \dots, r_n(\xi)\}$  based on  $\hat{X}(\xi)$ ;  $O(n)$
  - 2 Compute  $C^{p*}(\xi)$  as in (7);  $O(n/\epsilon^3 + p^2/\epsilon^6)$
  - 3 Find pairs of camera and setting that minimize the camera traveling time;  $O(p^2 \log p)$
  - 4 Adjust  $p$  cameras based on  $C^{p*}(\xi)$  by  $t = \xi$ ;  $O(1)$
- 

*Theorem 1:* Algorithm RA runs in  $O(n/\epsilon^3 + p^2/\epsilon^6 + p^2 \log p)$  time.

### C. PTZ camera parameter selection

After each camera is assigned a subset of objects, the camera tries to track these objects for the recording time  $\delta_r$ . This requires to select the camera parameter setting such that the satisfaction is maximized for each recording interval. Given each recording interval is represented as  $[t - \tau, t)$  and the  $i$ -th camera is assigned a subset of objects with predicted states at time  $t$ ,  $\hat{X}_i(t) = \{\hat{x}_1(t), \hat{x}_2(t), \dots\}$ . The corresponding observation requests are generated  $R_i(t) = \{r_1(t), r_2(t), \dots\}$ . The camera setting at time  $t$ ,  $c^*(t)$ , is then determined by maximizing the satisfaction to  $R_i(t)$ ,

$$c^*(t) = \arg \max_c \sum_{r_i(t) \in R_i(t)} s(c, r_i(t)). \quad (8)$$

This problem can be solved in [14] with running time  $O(|\hat{X}_i|/\epsilon^3)$ , where  $|\hat{X}_i|$  is the cardinality of  $\hat{X}_i$  and  $\epsilon$  is

the approximation bound. However, (8) does not consider the fact that within time  $\tau$ , the PTZ camera can only micro-adjust within a limited setting range. We assume the pan, tilt and zoom motion of the camera are independent. The reachable ranges for pan, tilt and zoom settings within time  $\tau$  are  $\alpha$ ,  $\beta$  and  $\gamma$ , respectively. Then we rewrite (8) as,

$$c^*(t) = \arg \max_{c \in \alpha \times \beta \times \gamma} \sum_{r_i(t) \in R_i(t)} s(c, r_i(t)). \quad (9)$$

It is worth mention that most PTZ cameras' pan and tilt motion is fast enough to keep tracking most objects in the scene. For example, the empirically estimated transition speed of the Panasonic HCM 280 camera is  $300^\circ/sec$ . for pan,  $200^\circ/sec$ . for tilt and  $5 \text{ levels}/sec$ . for zoom. Considering the camera has  $21 \times$  zoom levels and only less than  $50^\circ$  FOV, the time for changing pan and tilt settings is much less than the time for changing the camera zoom. Changing the zoom level when the camera is moving also creates significant motion blurring and requires re-focusing. Therefore, in practice, we only search for the pan and tilt settings in  $\alpha \times \beta$  while remain the zoom level.

We summarize the PTZ camera parameter selection scheme in Algorithm 2. Note  $\sum_i |\hat{X}_i| \leq n$ .

---

**Algorithm 2:** PTZ Camera Parameter Selection (PTZ-CPS)

---

- Input:** Current time  $\xi$ ;  $i$ -th camera current setting  $c_i^*(\xi)$ ; predicted object subset states  $\hat{X}_i(\xi + \tau) = \{\hat{x}_1(\xi + \tau), \hat{x}_2(\xi + \tau), \dots\}$ .
- Output:**  $i$ -th camera setting at time  $\xi + \tau$ ,  $c_i^*(\xi + \tau)$ .
- 1 Generate requests  $R_i(\xi + \tau) = \{r_1(\xi + \tau), r_2(\xi + \tau), \dots\}$  based on  $\hat{X}_i(\xi + \tau)$ ;  $O(|\hat{X}_i|)$
  - 2 Compute  $\alpha, \beta, \gamma$  based on  $c_i^*(\xi)$ ;  $O(1)$
  - 3 Compute  $c^*(\xi + \tau)$  as in (9);  $O(|\hat{X}_i|/\epsilon^3)$
  - 4 Micro-adjust  $i$ -th camera based on  $c_i^*(\xi + \tau)$  by  $t = \xi + \tau$ ;  $O(1)$
- 

*Theorem 2:* Algorithm PTZ-CPS runs in  $O(|\hat{X}_i|/\epsilon^3)$  time, where  $|\hat{X}_i|$  is the cardinality of  $\hat{X}_i$ . Computing parameters for all  $p$  cameras takes  $O(n/\epsilon^3)$  time.

### D. Dynamic weighting

If we keep the request weight in (3) unchanged, the system will create a ‘‘biased frame selection’’ model that always prefers certain objects instead of balancing the camera resource for all objects. We address this issue by carefully designing the temporal weight  $\omega_i(t)$  based on two intuitions: 1) object exiting FOV sooner is of more importance and 2) object less satisfied in history is of more importance. The first intuition is derived from the earliest deadline first (EDF) policy [6]. The policy addresses the emergency of the requests. The second intuition addresses sharing the camera resource for all objects to achieve balanced observation over time. We define,

$$\omega_i(t) = \mu_i(t) \cdot \nu_i(t),$$

where  $\mu_i(t)$  and  $\nu_i(t)$  address the first and second intuitions, respectively. One candidate form of  $\mu_i(t)$  is,

$$\mu_i(t) = \min(\rho^{\hat{d}_i - t}, 1), \quad (10)$$

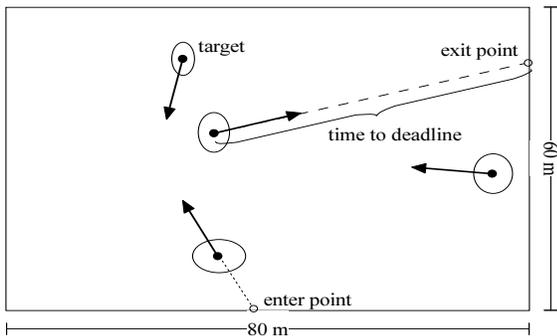


Fig. 3. Simulated scene. Each object is represented as an ellipse and enters the scene from one of the four sides following a Poisson process. The orientation is bounded within  $[-40^\circ, 40^\circ]$  with respect to the perpendicular of the side. The object maintains constant velocity and its time to exit the scene is predicted.

where  $\hat{d}_i$  is the predicted deadline for  $i$ -th object to exit the FOV and  $0 < \rho < 1$  is a parameter that controls how quick the emergency increases. When  $t \rightarrow \hat{d}_i$ ,  $\mu_i(t) \rightarrow 1$ , as maximum.

To design  $\nu_i(t)$  we need to first define the accumulative unweighted satisfaction (AUS)  $\eta_i(t)$ ,

$$\eta_i(t) = \sum_{j=1}^p \sum_{t_k \leq t} \frac{s(c_j(t_k), r_i(t_k))}{\omega_i(t_k)}, \quad (11)$$

where the variable  $t_k$  refers to the discrete times when cameras take frames. The AUS essentially reflects how well an object is satisfied in history. We design  $\nu_i(t)$  as,

$$\nu_i(t) = \max\left(1 - \frac{\eta_i(t)}{n_e}, 0\right), \quad (12)$$

where  $n_e$  is a parameter indicating the extent to which an object need to be observed. When  $\eta_i(t) \geq n_e$ ,  $\nu_i(t)$  is zero and we contend the object is fully satisfied and needs no observation any longer. Both  $\mu_i(t)$  and  $\nu_i(t)$  are bounded in range  $[0, 1]$ , which keeps the satisfaction metric in (4) a standard metric.

## V. EXPERIMENT

We carry out a simulation for evaluating the scheduling scheme based on random inputs. The system is programmed in Microsoft Visual C++. The simulation is carried out on a Windows XP desktop PC with 2.0 GB RAM, 300 GB hard disk space and a 3.2 GHz Pentium CPU.

### A. Simulation setup

As shown in Figure 3, a simulated  $80 \times 60$  meters scene is constructed. There are 4 entrances on each side. The size of the entrance is 30 meter. Each object enters the scene through one side and maintains a constant velocity. Seven random numbers are needed to characterize each object. First, a random integer number ranging from 1 to 4 is generated to indicate which side the object enters through. Then a random real number in  $[0, 1]$  is generated to indicate the entering point along the side. After that, the orientation of the object is determined by a random angle within the range

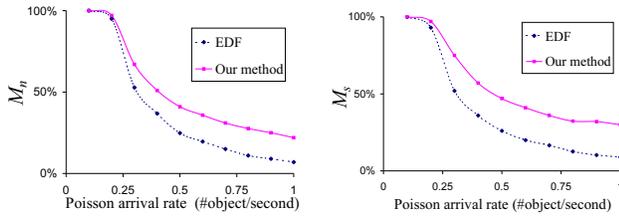
$[-40^\circ, 40^\circ]$  with respect to the perpendicular of the side. The object speed is generated from a truncated Gaussian with a mean of 1.5 m/s and standard deviation of 0.5 m/s, which is basically the speed of a walking people. The lengths of the two axes of the ellipse that represents the object are randomly generated from a range  $[1.5, 2.5]$  m. Finally, the desirable resolution of the object is generated from a range  $[1, 21]$  (magnification), which is also the Panasonic HCM280 camera zoom range. The cameras run in 10 fps, which means  $\tau = 0.1$  s. Then  $\alpha = 30^\circ$  and  $\beta = 20^\circ$ . 5000 objects arrive in the scene following a Poisson process with arrival rate  $\lambda$ , which represents the congestion level of the scene. We set the lead time  $\delta_l = 4$  s, which guarantees that in the request assignment phase, camera adjustment is completed before cameras intercept the objects. We set  $\delta_r = 6$  s, which is equivalent to  $n_r = 60$  frames. We set the parameter  $n_e = n_r$  in (12) and  $\rho = 0.5$  in (10) and  $\epsilon = 0.25$ . Two PTZ cameras are used, i.e.,  $p = 2$ . We set the approximation bound  $\epsilon = 0.25$ .

### B. Metric and results

We compare our scheduling scheme with the earliest deadline first (EDF) policy proposed in [6]. EDF is a heuristic scheme where the camera always picks the object with earliest deadline. With each congestion setting, 20 trials are carried out for average performance. We first compare the two schemes based on the ratio of number of objects that are observed for at least  $n_r/2$  times to the total number of objects pass through the scene. We term this metric as  $M_n$ . This metric essentially indicates how many objects the system can capture and observe for a period of time. Figure 4(a) shows the comparison result. It is shown that when the Poisson arrival rate  $\lambda$  is small, i.e., there are few objects in the scene, both scheduling schemes can reach almost best possible ratio (100%). When  $\lambda$  increases, i.e., the traffic in the scene becomes heavy, the performance of EDF deteriorates significantly quicker than our method. In the heavy traffic scenario, our method outperforms the EDF by over 200%.

We also compare based on the satisfaction to the objects since it takes into account not only the times that an object is observed, but also the resolution of the observation. As mentioned earlier, the AUS as defined in (11) indicates how well an object is satisfied. We define the second metric  $M_s$  as the ratio of average AUS to the maximum possible satisfaction for each object (i.e.,  $n_e$ ). Figure 4(b) summarizes the comparison based on  $M_s$ . It is shown that our method deteriorates even slower as  $\lambda$  increases. In the heavy traffic scenario, our method outperforms the EDF by 250%. This is not surprising since in heavy traffic situations, objects tends to be close to each other, where multi-object coverage has much greater advantage.

The computation time for both request assignment and camera parameter selection depends on the value of  $\lambda$ . In the heaviest scenario (i.e.,  $\lambda = 1$ ), the maximum number of object in the scene at any time is less than 100. In this case, the computation time for request assignment is less than 0.5



(a) Comparison of scheduling policies based on  $M_n$ . (b) Comparison of scheduling policies based on  $M_s$ .

second which is significantly less than  $\delta_l$ . The computation time for all PTZ cameras' parameter selection is less than 0.05 second which is also less than  $\tau$ .

Careful analysis reveals that our satisfaction formulation in (4) is actually a generalization of many existing scheduling schemes. For example, if we tune parameter  $\rho$  in (10) approaching to zero, then the change in  $\mu_i(t)$  dominates the change in the overall weight. That means we extremely care the emergency of the request and thus the scheduling converges to the earliest deadline first (EDF) policy [6]. Also, given we set the requested resolution close to highest camera resolution, or we change the resolution ratio term  $\min(\frac{z_i}{z}, 1)$  in (4) to indicator function  $I(z_i \geq z)$ . This means we only accept the images with least requested resolution. Then the frame selection algorithm would assign at least single object to each PTZ camera in the worst case, which is exactly the scheduling scheme based on single object tracking as in almost all existing works. This is also one reason our scheduling scheme outperforms the existing work.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we present an autonomous vision system that consists of multiple robotic PTZ cameras and a fixed wide-angle camera for observing multiple objects simultaneously. We present the system with observation request generation, request assignment and PTZ camera parameter selection modules. We formulate the PTZ camera scheduling as a sequence of request assignment and camera parameter selection problems with objective of maximizing the satisfaction to requests. The problems are solved by our recent algorithms on frame selection problem. We compare the system with an existing work based on simulation. The simulation results show our system significantly enhances the observation performance especially in heavy traffic situations.

In the future, we will investigate how different frame selection formulation would impact the system performance and how they fit human user need in practice. Another interesting extension is to consider the camera traveling time within the request assignment. Intuitively, asynchronized observation by multiple PTZ cameras would further enhance the system performance. The camera content delivery through internet would be another interesting topic especially when number of camera increases.

## ACKNOWLEDGEMENT

We would like to thank N. Amato, F. van der Stappen, N. Papanikolopoulos, R. Volz and K. Goldberg for their in-

sightful input, Q. Ni for implementing the motion detection, and C. Kim, J. Zhang, A. Aghamohammadi and Z. Bing for their contribution to the Networked Robot Lab at Texas A&M University.

## REFERENCES

- [1] X. Zhou, R. T. Collins, T. Kanade, and P. Metes, "A master-slave system to acquire biometric imagery of humans at distance," in *IWVS '03: First ACM SIGMM international workshop on Video surveillance*. New York, NY, USA: ACM, 2003, pp. 113–120.
- [2] N. Krahnstoeber, T. Yu, S.-N. Lim, K. Patwardhan, and P. Tu, "Collaborative real-time control of active cameras in large scale surveillance systems," in *Proc. Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2)*, Marseille, France, October 2008.
- [3] A. Hampapur, S. Pankanti, A. Senior, Y.-L. Tian, L. Brown, and R. Bolle, "Face cataloger: multi-scale imaging for relating identity to location," *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pp. 13–20, July 2003.
- [4] R. Bodor, R. Morlok, and N. Papanikolopoulos, "Dual-camera system for multi-level activity recognition," *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1, pp. 643–648 vol.1, Sept.-2 Oct. 2004.
- [5] L. Fiore, D. Fehr, R. Bodor, A. Drenner, G. Somasundaram, and N. Papanikolopoulos, "Multi-camera human activity monitoring," *J. Intell. Robotics Syst.*, vol. 52, no. 1, pp. 5–43, 2008.
- [6] C. J. Costello, C. P. Diehl, A. Banerjee, and H. Fisher, "Scheduling an active camera to observe people," in *VSSN '04: Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*. New York, NY, USA: ACM, 2004, pp. 39–45.
- [7] S.-N. Lim, L. Davis, and A. Elgammal, "Scalable image-based multi-camera visual surveillance system," *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pp. 205–212, July 2003.
- [8] A. D. Bimbo and F. Pernici, "Distant targets identification as an on-line dynamic vehicle routing problem using an active-zooming camera," *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 97–104, 2005.
- [9] S.-N. Lim, L. S. Davis, and A. Mittal, "Constructing task visibility intervals for video surveillance," *Multimedia Systems*, vol. 12, no. 3, pp. 211–226, 2006.
- [10] C. Zhang, Z. Liu, Z. Zhang, and Q. Zhao, "Semantic saliency driven camera control for personal remote collaboration," *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, pp. 28–33, Oct. 2008.
- [11] E. Sommerlade and I. Reid, "Information-theoretic active scene exploration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, USA, 2008.
- [12] D. Song, *Sharing a Vision: Systems and Algorithms for Collaboratively-Teleoperated Robotic Cameras*. Springer, 2009.
- [13] D. Song, A. F. van der Stappen, and K. Goldberg, "Exact algorithms for single frame selection on multi-axis satellites," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 1, pp. 16–28, January 2006.
- [14] D. Song and K. Goldberg, "Approximate algorithms for a collaboratively controlled robotic camera," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1061–1070, October 2007.
- [15] D. Song, N. Qin, and K. Goldberg, "Systems, control models, and codec for collaborative observation of remote environments with an autonomous networked robotic camera," *Autonomous Robots*, vol. 24, no. 4, pp. 435–449, 2008.
- [16] Y. Xu, D. Song, J. Yi, and F. van der Stappen, "An approximation algorithm for the least overlapping p-frame problem with non-partial coverage for networked robotic cameras," in *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, May 2008, pp. 1011–1016.
- [17] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *6th European Conference on Computer Vision (ECCV)*, vol. 2, Dublin, Ireland, June/July 2000, pp. 751–761.
- [18] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, May 2003.
- [19] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.