J Inf Process Syst, Vol.13, No.5, pp.1299~1319, October 2017
https://doi.org/10.3745/JIPS.02.0074

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

# Sector Based Multiple Camera Collaboration for Active Tracking Applications

Sangjin Hong*, Kyungrog Kim**, and Nammee Moon***

**Abstract**

This paper presents a scalable multiple camera collaboration strategy for active tracking applications in large areas. The proposed approach is based on distributed mechanism but emulates the master-slave mechanism. The master and slave cameras are not designated but adaptively determined depending on the object dynamic and density distribution. Moreover, the number of cameras emulating the master is not fixed. The collaboration among the cameras utilizes global and local sectors in which the visual correspondences among different cameras are determined. The proposed method combines the local information to construct the global information for emulating the master-slave operations. Based on the global information, the load balancing of active tracking operations is performed to maximize active tracking coverage of the highly dynamic objects. The dynamics of all objects visible in the local camera views are estimated for effective coverage scheduling of the cameras. The active tracking synchronization timing information is chosen to maximize the overall monitoring time for general surveillance operations while minimizing the active tracking miss. The real-time simulation result demonstrates the effectiveness of the proposed method

**Keywords**

Active Tracking, Master-Slave, Object Dynamics, Sector-Based Representation

# 1. Introduction

Multiple cameras are incorporated into the sensor networks to visually monitor the objects in many areas such as airports and industrial complexes for public safety purposes [1,2]. However, due to the orientation of the visual coverage and the distance between the camera and the objects, the limited resolution images of the objects are often obtained from the detection. Because of such limitation, many post-processing such as face recognition and object trajectory analysis are not effective in many surveillance systems. Hence, it is highly desirable to obtain high resolution images of all objects for effective surveillance.

To avoid generation of the limited resolution images of objects, many works on active tracking the objects with a single camera have been proposed [3-5]. The active tracking selects an object and provides high resolution image by zooming and panning into the object. This process is repeated for all objects visibly detected by the camera. Since the objects are highly dynamic within the visual coverage



Manuscript received April 10, 2017; first revision May 30, 2017; accepted May 31, 2017.

**Corresponding Author:** Nammee Moon (mnm@hoseo.edu)
* Dept. of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, USA (sangjin.hong@stonybrook.edu)
** Dept. of Electronic Engineering, Hoseo University, Asan, Korea (it4all@hoseo.edu)
*** Dept. of Computer Software, Hoseo University, Asan, Korea (mnm@hoseo.edu)

area, most of the prior works on active tracking utilizing single camera were focused on coping with the slow panning and zooming speeds of the cameras to minimize miss-tracking when the objects are densely populated. Moreover, due to the limited visual coverage of the camera, the active tracking system with single camera is not suitable for large area applications. To improve the active tracking coverage, many of these active tracking systems have been extended to incorporate multiple cameras [6-10].

In the active tracking systems employing multiple cameras, the master-slave approach is often utilized where the master camera covers the entire area for the object detection, then the slave cameras perform active tracking based on the priority-based or the round-robin based scheduling [9,11-13]. With the master-slave approach, the process of the camera scheduling and their coordination is straightforward but scalability of the system is not easily achieved without having multiple master cameras since one master camera cannot visually cover the large area.

Hence, the distributed approach is preferable to maximize the camera utilization and achieve scalability of the system. In the distributed approach, there is no designation for the master camera and the slave cameras. Each camera can be either the master or the slave. In addition, the number of the master or the slave cameras dynamically changes depending on the object dynamics and the object density distributions. However, the distributed approach has loose coordination among the cameras. Thus, the information of overall correspondence of the detected objects by the individual camera is necessary in order to minimize the redundant active tracking [14-16].

In this paper, a scalable distributed multiple camera collaboration strategy for multiple camera active tracking in a large area is proposed. The proposed approach is based on distributed mechanism but emulates the master-slave mechanism. The master and slave cameras are not designated but adaptively determined depending on the object dynamic and density distribution. The proposed master-slave emulation combines local object profiles to construct the overall global coverage information. The global coverage information utilizes the estimated dynamics of all objects visible by the local cameras where the information is used to select an object for the active tracking. The load balancing of active tracking operations is incorporated into the mechanism to maximizes active tracking coverage of the objects. The active tracking synchronization timing information is chosen to maximize the overall monitoring time for general surveillance operations while minimizing the active tracking miss. The real-time simulation result demonstrates the effectiveness of the proposed method. While secure important exchanges between the cameras are important issues, the security issue is not considered in this paper.

The remainder of this paper has four sections. Section 2 presents a basic sector-based scheduling strategy for static objects. The relationship between the mapping and scanning is discussed. Section 3 presents the dynamic scheduling for supporting dynamic objects including the object density fluctuations. Section 4 presents the simulation results for the proposed multiple camera active tracking strategy. Finally, our contribution is summarized in Section 5.

## 1.1 Application Model and Motivation

Fig. 1 illustrates a situation where the active tracking system using multiple cameras is coping with the density and dynamic of the objects in real-time. The system is composed of multiple cameras, which form a distributed active camera network instead of using one master-slave camera collaboration mode. The master-slave based monitoring is not suitable for the model since multiple cameras must be used to cover entire range wasting the resources. Each camera tracks objects in its observable region and obtains their zoomed image with high resolution to clearly identify the objects. There may exist overlapping areas between different cameras and they cooperate so that they track the objects effectively in short amount of time.

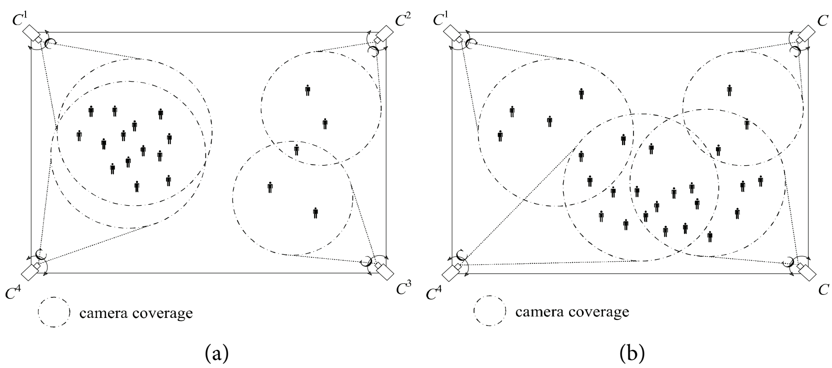



**Fig. 1.** Illustration of a collaborative multiple camera active tracking system to maximize the coverage of the objects. The view angles of the cameras are changed in time to cope with the densities and the dynamics of the objects. (a) At $t_1$ and (b) at $t_2$.

## 1.2 Problem Description and Approach

In the surveillance environment, the object densities and distributions are highly dynamic. Moreover, the PTZ camera translation time is not negligible. The resources must adapt to the object dynamics and efficient estimation of the overall objects is extremely critical for providing the resources to the objects fairly. While, the master-slave approach may obtain the overall dynamics rather easily, in the distributed and time multiplexing of sensors approach, such problem is not trivial and efficient way to obtain the objects dynamics and the scheduling of resources is critical.

Fig. 2 illustrates the system configuration of the proposed system. Each camera monitors the overall area as much as possible for normal surveillance operations. Each of the camera alternates between normal state and zoomed state. The process starts with the camera in the normal state. To effectively balance between the active tracking and normal monitoring, the object distribution pattern as well as the dynamics of the objects to be tracked has significant impacts on the active tracking results.

In order to emulate the master-slave method for the distributed approach, the global view of all objects are obtained by combining the local view of individual cameras. The environment is divided into a set of sectors where each sector is mapped to an image of the camera. The image of a camera is also equally divided into multiple rectangular regions. To establish the correspondence between local sectors of multiple cameras, the relationship between the global sectors to the local sectors of the cameras is maintained.

Each camera detects the objects that are visible on their local view. Using the global sector to local sector correspondence, all visible objects are projected to the global view. Based on the global object distribution, the load balancing and active tracking assignments are determined.

The load balancing and active tracking ordering are determined by the object trajectory estimation where the object disappearing early is tracked first. If there are significant changes in object density distribution, a new set of the camera positions are computed and they are placed for minimizing the miss tracking. When the new placements of cameras are computed, the finite translation time of the cameras is taken into consideration where the cameras may not be able to participate in monitoring as well as active tracking during the translation. The proposed system minimizes the miss and redundant tracking while using the overall monitoring coverage factor as the constraint.

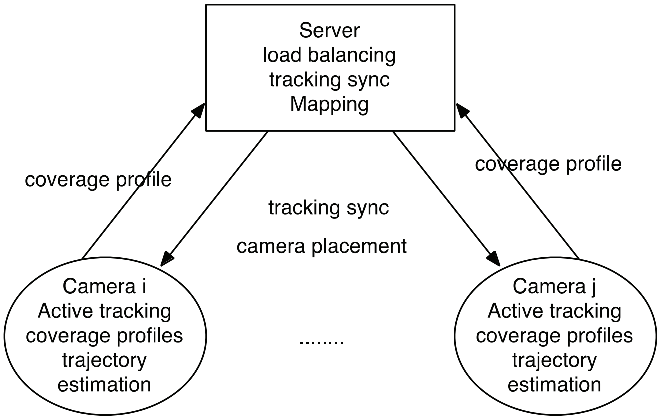



**Fig. 2.** The functional interaction between the server and the cameras. The camera provides the local view of the object profiles and the server controls the cameras for the active tracking synchronization as well as the coverage.

## 2. Basic Collaborative Sector Distribution

### 2.1 Global and Local Sector Distribution Strategy

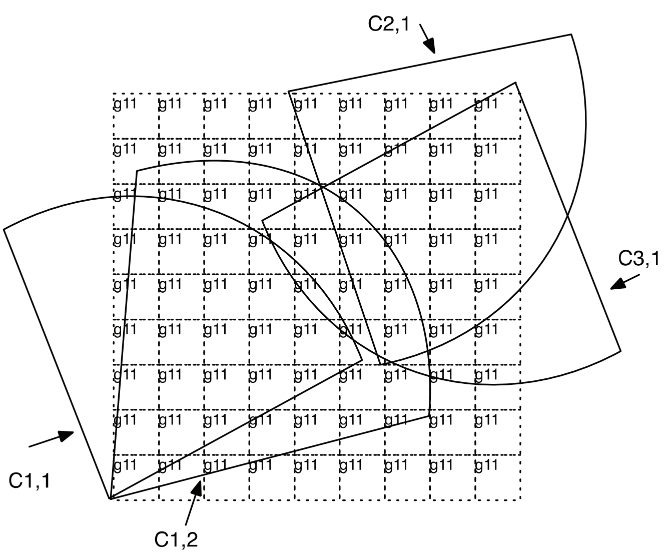
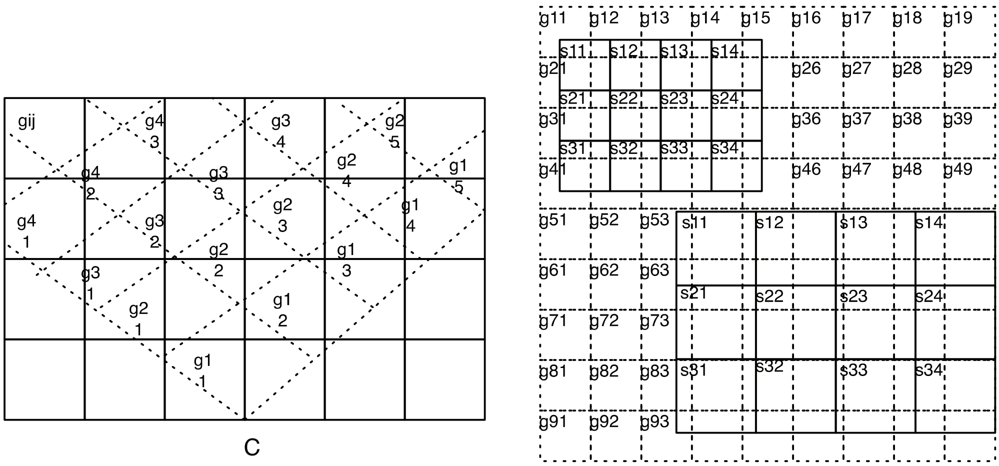



**Fig. 3.** Illustration of the camera coverage. Each camera *i* has many possible different PTZ setting *j*. Different camera setting produces different global area coverage.




(a) (b)

**Fig. 4.** (a) Local view perspective illustration of that the projection of global sectors to camera local view may have a distortion (the local view of camera is shown). (b) Global view perspective illustration of the sector correspondence between the camera sectors and the global sectors.

| Grid Index | Camera i | | | ... | Camera i | | |
|---|---|---|---|---|---|---|---|
| | Position i | ... | Position k | ... | Position i | ... | Position k |
| g11 | s11, s23 | ... | s11, s23 | ... | s11, s23 | ... | s11, s23 |
| g12 | g12 | ... | g12 | ... | g12 | ... | g12 |
| g12 | g12 | ... | g12 | ... | g12 | ... | g12 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| gkm | gkm | ... | gkm | ... | gkm | ... | gkm |

(a)

| Sector Index | Position i Grid List | ... | Position k Grid List |
|---|---|---|---|
| s11 | g11, g23 | ... | g11, g23 |
| s12 | ... | ... | ... |
| s12 | ... | ... | ... |
| ... | ... | ... | ... |
| skm | ... | ... | ... |

(b)

**Fig. 5.** (a) Global view data structure maintaining the global sectors to the local sectors for the individual placement position of all cameras. (b) Local view data structure maintaining the local camera sectors to the global sectors for individual placement position.

As shown in Fig. 3, the global area is divided into a set of sectors and each camera covers a subset of the global sectors. Each camera *i* has many possible different camera settings *j*. The camera coverage depends on a specific set of PTZ parameters. Various view coverages are possible with the different combination of PTZ parameters. The global sectors are uniquely indexed so that the coverage of each camera is represented by a set of the global sectors. The global sector index is used to determine the coverage correspondences between multiple cameras. In order to establish the coverage correspondence between the global view and the camera local view, the local view is also divided in a set of sectors as illustrated in Fig. 4(a). Because the size of global sectors projected on the local sector may not correspond to the sizes of the local sectors, multiple global sectors may be mapped to multiple local sectors as illustrated in Fig. 4(b). Thus, redundancy is possible between the sectors and grids mapping. The effects of this redundancy on computation is minimized in the algorithm.

The correspondence between the camera local sectors to the global sectors is established with two data structures. The first data structure maintains the mapping information from the global sector perspective as illustrated in Fig. 5(a). For each global sector, the corresponding local sectors for each camera position are maintained. Multiple local camera sectors may be mapped to a single global sector (or vice versa). The other data structures maintain the mapping information form the camera local sector perspective as illustrated in Fig. 5(b). In this data structure, the list of corresponding global grid list is provided for the sector for each camera position. With the mapping table for each camera, the global to local view relationship is clearly specified. Also, by checking which local sectors of different cameras contain the same global sector indices, which means that they cover the same global sectors, the correspondence between different camera views could be easily achieved.

## 2.2 Sector Coverage Solution Selection

As discussed in the previous section, the mapping method generates many possible solutions. Moreover, the mapping method discussed previously does not consider the object distribution and their dynamic. When selecting a mapping solution, the load balancing must be considered at the same time. The main objective of the load balancing is to distribute the object evenly among the cameras so that the total time it takes to cover the objects is evenly distributed.

Fig. 6 illustrates the overall strategy for estimating the objects distribution on the global view by projecting the locally detected objects (for simplicity, the discussion considers the case with perfect projections, that is, the local views have no distortion in terms of global views. This will not affect our algorithms and the results). As shown in the figure, only the objects that are visibly covered by the cameras are considered in the estimation process. Hence, the camera coverage at the time of the estimation is very critical in obtaining the accurate estimation of the object distribution. Each object covered by the camera are projected to the global sector by utilizing the data structure discussed in the previous section. This process is done for all cameras. Once the projected objects are obtained, the load balancing and mapping solution selection can be performed.

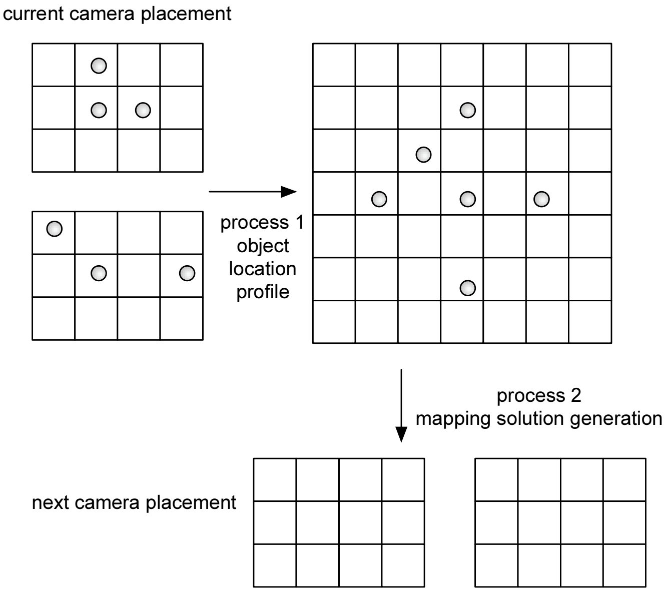



**Fig. 6.** Estimating the object distribution on global view using projection of the locally detected objects. From the global estimation, the mapping solution as well as load balancing is achieved.

Camera i, Position j

| Sector Index | Number of Objects |
|---|---|
| s11 | 0 |
| s12 | 1 |
| s13 | 0 |
| ... | ... |
| s34 | 0 |

...

Camera k, Position m

| Sector Index | Number of Objects |
|---|---|
| s11 | 1 |
| s12 | 0 |
| s13 | 0 |
| ... | ... |
| s34 | 0 |

| Grid Index | Number of Objects |
|---|---|
| g11 | 0 |
| g12 | 0 |
| g13 | 0 |
| ... | ... |
| g77 | 0 |

**Fig. 7.** Illustration of the data structure maintaining the estimated number of the objects in the local sectors. Object information on Fig. 6 is used for the table construction.

Fig. 7 illustrates the detection process by the local camera. The number of objects detected by the camera is annotated. After the individual annotation, the number of objects is projected to the global sector using the global-to-local section relationship defined in the previous section. The object distribution is used for the load balancing. These are only the estimation and the degree of accuracy depends on the size of sectors and sector correspondence.

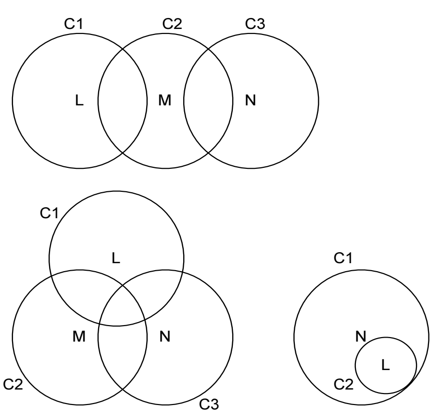



**Fig. 8.** Illustration of different coverage configuration for the load balancing. Chained shared region configuration, multiple shared region configuration, and inclusive shared region configuration.

Given the placement and the position of the cameras, several possible overlapping scenarios are possible as shown in Fig. 8. The first case is the daisy-chained coverage configuration. The second case is the multiple-coverage overlapping configuration. The last one is complete enclosure of one coverage view inside of the other. In the most realistic situation, the actual mapping is a combination of these three. Algorithm 1 illustrates the load balancing strategy. Each camera initially selects the objects that are not covered by the other cameras. The load balancing mechanism divides the shared objects among the cameras and repeats the process until no shared objects are available. Special case is when one camera coverage view is completely enclosed within the other camera coverage view. In this case, all objects detected in the enclosed camera are considered as the objects that can be shared. When selecting the mapping, the balancing must be done simultaneously.

**Algorithm 1** Mapping and Load Balancing

Select a mapping solution with optimum overall coverage and lowest load for each camera;<br>
**for** (each camera) **do**<br>
    Count the number of objects in each local sectors;<br>
    Project the objects onto the global sector;<br>
    Ignore multiple count for the shared global sectors;<br>
**end for**<br>
Generate all possible mapping solutions;<br>
**for** (for each mapping solution) **do**<br>
    **for** (each camera) **do**<br>
        Project the objects on the global sectors to the local sectors;<br>
        Maintain the data structure by separating local only objects and shared objects for each camera;<br>
    **end for**<br>
    Set no currently solution;<br>
    **while** (No more shared objects) **do**<br>
        Determine average number of local only objects among all camera, $N_{average}$;<br>
        Determine the camera $C_{min}$ with the lowest local only object, $N_{min}$;<br>
        Assign $N_{average} - N_{min}$ to the $C_{min}$ and treat the assigned shared object as the local only objects for $C_{min}$;<br>
        Update the data structure;<br>
        Determine the overall coverage;<br>
        Keep this solution if better than the previous solution;<br>
    **end while**<br>
**end for**

## 2.3 Scanning and Selecting

After load balancing, the system will then perform sector scanning and object selection for active tracking. Each camera scans through their local sectors sequentially. Whenever there is a detected object in a certain sector, the camera selects the object and initiates the zooming process. Since object tracking is not possible during the zooming process, whenever there is a sector with multiple detected objects, the scanning process randomly selects one of the objects and initiates the zooming process. The cameras also keep counters for each sector indicating the number of the objects within the sectors. Each time the camera scans and tracks the sector, the counter for that sector will decrease by one. Since the two objects still have the same probability to be selected during the next scanning iteration, possible missed and redundant detections are likely to occur.

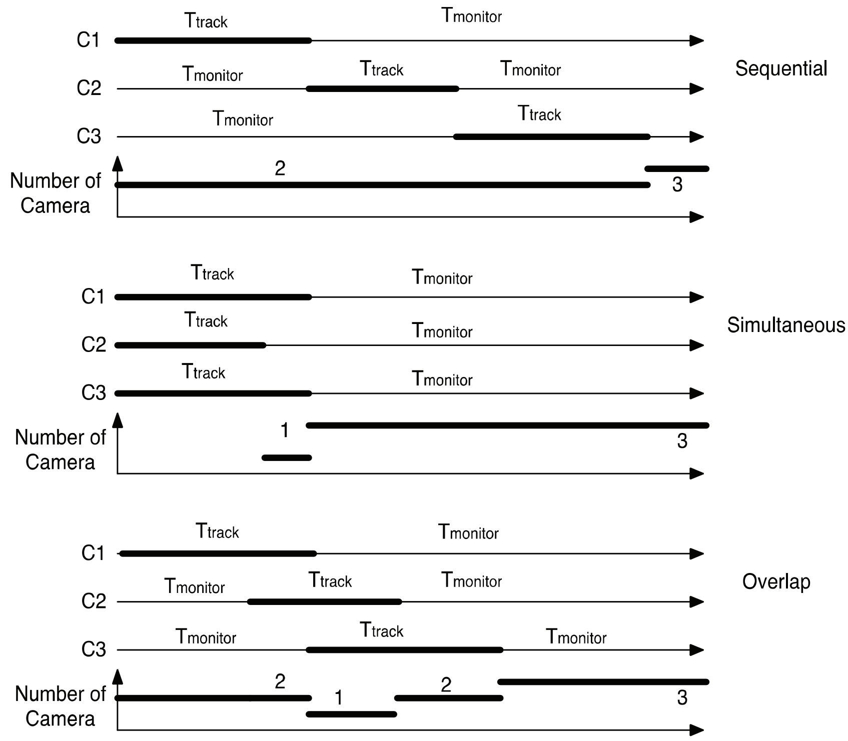



**Fig. 9**. Illustration of three different cameras collaborations strategies based on the start time of the active tracking process: sequential, simultaneous, and partial overlap.

The discussion of the single camera scanning and selection strategy can be extended to the case for the multiple cameras from the timing characteristic and the overall coverage perspectives. Fig. 9 illustrates the active tracking collaboration strategies for different start time of the active tracking process. Two key parameters are used in the illustration. The parameter $T_{monitor}$ represents the time duration the camera is monitoring the area. And the parameter $T_{track}$ represents the time duration that camera is performing the active tracking all objects assigned to the cameras. The figure assumes that the cameras $C_1$ covers five objects, $C_2$ covers four objects, and $C_3$ cover four objects, respectively. The first strategy illustrates that the camera is sequentially performing the active tracking operation where the number of cameras that are participating in overall monitoring is limited to two in the illustration. The second strategy is suitable when all cameras are performing active tracking operations simultaneously. Since the cameras are busy with the active tracking operations, the number of cameras covering the area is zero initially but all cameras are available for monitoring when the active tracking is completed. The last strategy combines both strategy where they overlap the monitoring and active tracking during the overall operations.

It is assumed that all the objects assigned to the cameras are actively tracked consecutively once the server signals the camera. However, even though this scheme works well for static objects, main issue with this scheme is that there is no coverage capability by the camera between each active tracking. Hence overall coordination of each active tracking needs to be controlled by the server.

Algorithm 2 illustrates the scanning, selection and tracking strategies under the condition when the object dynamic is slow (i.e., objects stays within the view of each camera for long period). Based on the object distribution profiles and load balancing, the camera selects an object for active track. The object selection is performed by sequentially covering the sectors. If more than one objects are within a sector, an object is randomly chosen.

**Algorithm 2** Scanning, Selection, Tracking

Wait for (server events) or (object counts);
While waiting for the active tracking signal, perform normal monitoring operation and check the number of objects within the sectors;

Change in the number of objects;
**if** (object counts) **then**
    Notify the server for possible mapping and load balancing;
**end if**

Active tracking operation;
Server Synchronization;
**if** (server event) **then**
    Scan the local sectors and select an object;
    If multiple objects in the sector, select the object randomly;
    Active track the object, mark the sector is covered;
**end if**

## 2.4 Complex and Partial Shared Coverages

The effects of the partial coverages are discussed from two perspectives: the mapping and load balancing perspective, and the scanning and tracking perspective. From the mapping perspective, if the sector sizes partially overlap between the local sector of a camera and the global sector, the algorithm considers the global sector map to the local sector if any portion of the sector area overlaps.

The elimination of the redundant sector counting, is carried out during the mapping process. Hence, their relative sizes of the sectors are critical from the load balancing perspective. Fig. 10 illustrates the relationship between the global and local sectors. There are two possible cases depending on the relative sector sizes. Unlike the discussion in the previous section, in realistic situations, one-to-one correspondence between global and local sectors is not possible.

If the size of the global sectors is smaller than the size of local sectors, it is hard to decide which global sector should be assigned. If the same object is detected by another camera, the double counting of the same object is possible (i.e., single object may be estimated as two in the global view). This will create an incorrect load balancing in terms of the number distribution. If the size of the local sector is smaller than the size of the global sector, the double counting problem is not an issue. Thus, the system prefers smaller local sector size.

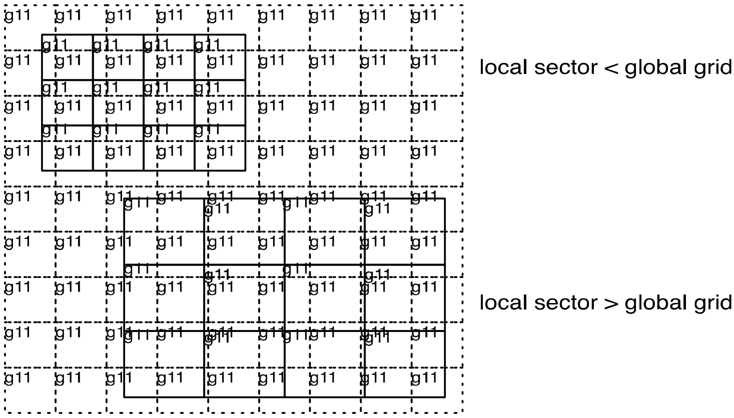



**Fig. 10**. Illustration of two cases where the sector sizes do not correspond between the local and global.

# 3. Load Balancing and Synchronization

## 3.1 Active Tracking Dynamic Objects

In actual environments, the objects are highly dynamic and the dynamics of the objects visible on the local view have great influence on the coverage solution selection process. This is because that the objects that are currently visible on the local view of a camera may disappear as the time passes even though the object may move with the same speed on the global view. The projected speed on the object on the local view causes the solution selection very difficult. There are many different possible scenarios as illustrated in Fig. 11. The new objects may enter the local view while some of the objects are exiting. Of the objects that are exiting, some of them may enter the local view of another cameras or leave complete out of the coverage.

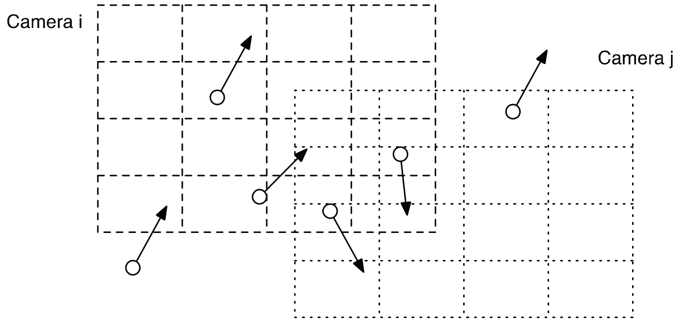
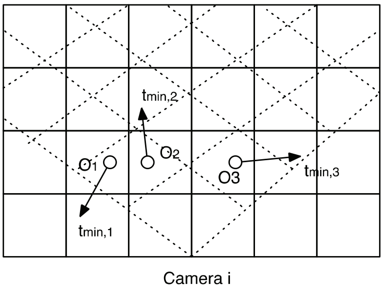



**Fig. 11.** Illustration of the object dynamics on local camera coverage. The objects may enter or exit the local view of the cameras.

The object dynamic issue is illustrated in Fig. 12. In Fig. 12(a), three objects move in different direction on the local view where the third object may exit before the first object. The projected speeds of these objects are different depending on the view angle of the camera. In the proposed approach, the parameter $t_{min,i}$, for each object $i$, specifies the minimum time appeared on the local view before the object leaves the coverage view of the camera. These values for each object for each camera are continuously estimated and tabulated by the server. are estimated by the local cameras for all objects and these values are tabulated in the data structure by the server as illustrated in Fig. 12(b).




(a)

| Object ID | Global Position | Camera 1 $t_{min}$ | Camera 2 $t_{min}$ | ... | Camera k $t_{min}$ |
|---|---|---|---|---|---|
| $O_1$ | g1,1 | 0.1 | - | ... | 3 |
| $O_2$ | g1,1 | 0.7 | - | ... | 2 |
| $O_3$ | g1,1 | 60 | 0.3 | ... | - |
| ... | ... | ... | ... | ... | ... |
| $O_N$ | g1,1 | 20 | - | ... | - |

(b)

**Fig. 12.** (a) Illustration of the $t_{min}$ depending on the objects location on the local view. The time varies significantly depending on the direction of movement. (b) The table storing the estimated $t_{min}$ for each object for all cameras.

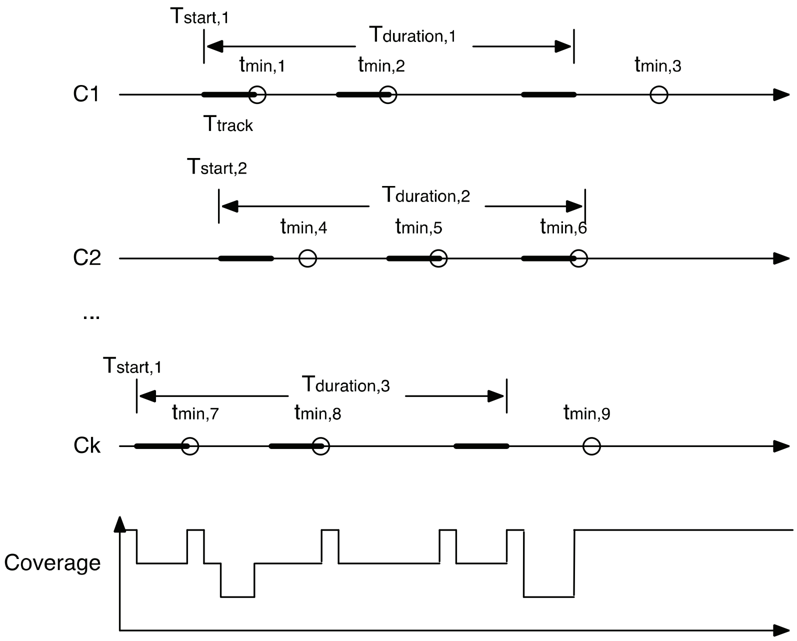



**Fig. 13.** Illustration of the scheduling orders received by the server. Each camera sends the $T_{exit}$ of all remaining objects to the server. The overall coverage is influenced by the active tracking synchronization.

For effective active tracking of all objects, the cameras must continuously exchange information with the server. Each camera sends the information about the object dynamics such as possible tracking ordering to the server in order to determine the best time to active track an object. This process minimizes the possibility that multiple cameras participate in the active tracking at the same time which reduces the monitoring capability. This is illustrated in Fig. 13. Three cameras are used in the illustration and each camera has three objects to track. Each object *i* has its $t_{min,i}$ has the latest time the object must be tracked before missing the active tracking. If the time is violated, the object will not be active tracked. The server synchronizes each camera with the parameters, $T_{start,i}$ signal and $T_{duration,i}$ for camera *i*. The first signal indicates that the active tracking process for a particular camera must be initiated. The second parameter indicates the duration of active tracking of all objects for a particular camera. As shown in the figure, each active tracking of an object *i* must be completed before the $t_{min,i}$. Also indicated in the figure, the entire active tracking process of all objects must be completed by $T_{duration,i}$ for camera *i*. Within the duration, multiple objects may be tracked.

## 3.2 Collaboration for Remapping of Coverage

When there are the global sectors which are not currently covered by the cameras, the objects in those global sectors will not be considered for the active tracking. In order to provide fairness of the coverage, each global sector has a timer, $T_{last,i}$ for global sector *i*. Initially, the value of $T_{last,i} = 0$. After the overall system process begins, the value of $T_{last}$ will increase in time. The larger value of $T_{last}$ implies that the sectors have not been covered for that duration. Depending on the application scenario, $T_{last,\, max}$ must be specified.

The server maintains the global sector coverage status including the last time the sector was covered as well as the current coverage. Based on this status information, the server may need to remap the global view for coverage. The remapping is necessary to cover the areas, which are not currently being monitored. If a camera completes the active tracking of all objects, then the camera is free to be allocated. If the coverage of some of the global sectors exceeds $T_{last,\, max}$, the remapping is performed. If remapping requires the camera currently not completed with the active tracking (i.e., some of the objects are not covered), then the remapping considers these uncovered objects during remapping. After the remapping, one of the issues that may arise is that the remapped cameras may try to track the same objects, which were tracked previously by the same camera or the other cameras. To avoid this redundancy, the server maintains the covered global sectors. This implies that whenever the camera performs the active tracking, the server maintains the sector so that the objects in those sectors will tend to have lower priority during load balancing.

As shown in the Fig. 14(a), the camera locally processes the object detection and sends the object local sector positions and the timing information to the server. After the load balancing is completed and maintains the order, the camera waits for the active tacking synchronization signal from the server. Each time the active tracking is performed, the tracked object information is sent to the server. Algorithm 3 summarizes the camera process algorithm.

On the other hand, Fig. 14(b) illustrates the server interacting with the multiple cameras for generating synchronization signals for active tracking. The server maintains the global coverage as much as possible. By updated status information, the server may dynamically perform the load balancing. In case the remapping is necessary, the server generates new placement schedule to cover the previously uncovered areas. Algorithm 3 summarizes the server process algorithm.

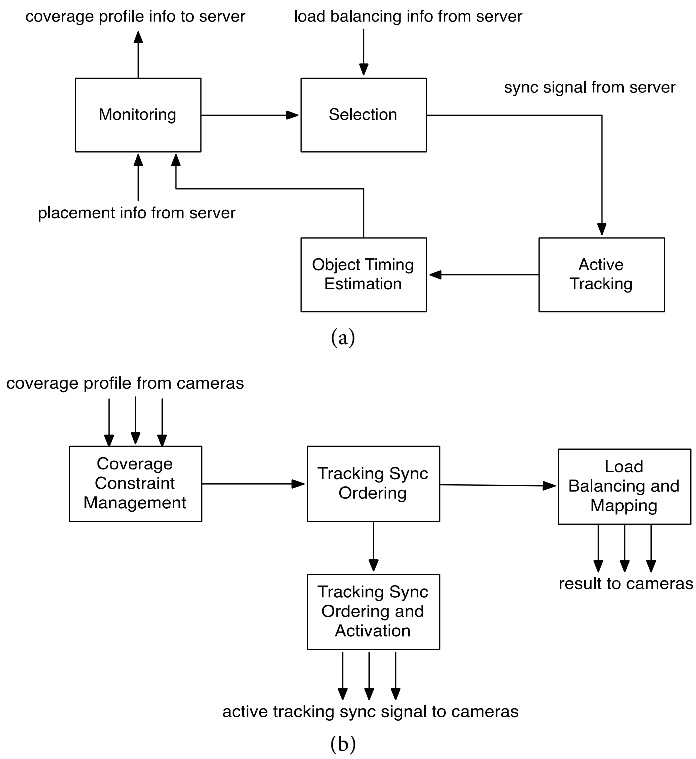



**Fig. 14.** (a) Illustration of the information flow of the system from the camera perspective. (b) Illustration of the information flow of the system from the server perspective.

**Algorithm 3** Camera Process Algorithm

Wait for (server events);
Server event types; new placement, sector coverage, active track synchronization;

Normal operation;
**while** (while waiting for any event) **do**
  Detect all objects, track all objects, and estimate $T_{min}$ of all objects;
  Notify the timing of remaining object if timing violation;
  Notify the status of the number of objects and their trajectories (entering and leaving);
**end while**

Server event process;
**if** (active track event) **then**
  Select the next object and active track;
  Mark the sector and object are covered;
  Go back to the normal operation;
**end if**
**if** (placement event) **then**
  Revise the current camera setting;
  Go back to the normal operation;
**end if**
**if** (coverage event) **then**
  Revise the current balanced sectors;
  Go back to the normal operation;
**end if**

# 4. Evaluation and Analysis

## 4.1 Simulation Setup

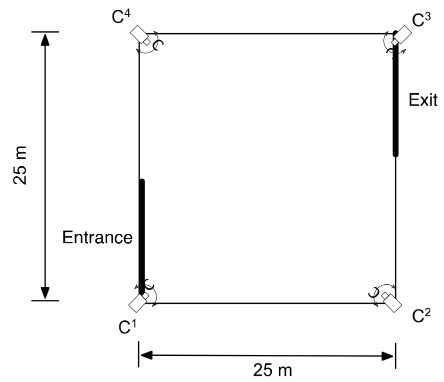



**Fig. 15.** The illustration of the simulation setup. Four cameras are placed at different locations to consider view dependency in the simulation. The objects enter through the entrance and leave through the exit. The times the objects are within the environment are randomly chosen.

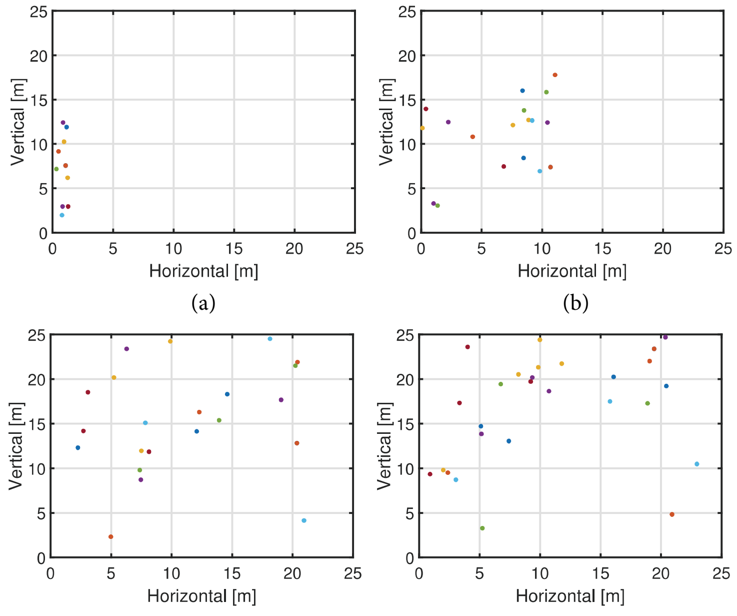



**Fig. 16.** Illustration of the object distribution as a function of time. All objects initially enter through the entrance and leave through the exit. After 0 seconds (a), 20 seconds (b), 40 seconds(c), and (d) 60 seconds.

Fig. 15 illustrates the simulation setup used to evaluate the effectiveness of the proposed method. Multiple cameras with PTZ capability and the finite active tracking time $T_{track,min}$ are assumed in the evaluation. Their placements are selected to create coverage overlap. The densities and dynamics of multiple objects are also varied to illustrate their effects on the active tracking performance. The performances are measured in terms of redundant tracking and miss tracking given overall coverage time constraint $T_{all,min}$. The overall coverage time constraint indicates that during the active tracking collaboration, the minimum amount of time that all cameras must cover the entire area at the same time must be larger than $T_{all,min}$. The object density and diffusion distribution are illustrated in Fig. 16. Maximum total number of objects used in the setup is set to 50.

## 4.2 Effects of Normal Coverage Constraints

In this section, the performance of active tracking and coverage with coverage constraints are evaluated and analyzed. In the simulation, the camera active tracking translation time is set to 0.5 seconds and the minimum camera normal monitoring time is set to 2 seconds. Fig. 17 illustrates the active tracking behaviors for two different types of synchronization: asynchronous and synchronous. In the asynchronous, each camera determines which object to track based on the synchronization generated by the server. The active tracking timing is illustrated in Fig. 17(a). As shown in the figure, high level of overall normal coverage is maintained. The average coverage is illustrated in Fig. 17(b).

On the other hand, in the synchronous active tracking, all cameras initiate active tracking at the same time and its active tracking behavior is illustrated in Fig. 17(c). Note that the active tracking process for all cameras starts at the same time while the number of objects are varied. For synchronous scheme where all cameras perform active track at the same time, the average coverage is higher but the temporary coverage steeply drops at times. While multiple low temporary coverage durations are presented, the average coverage is slightly better in the synchronous case.

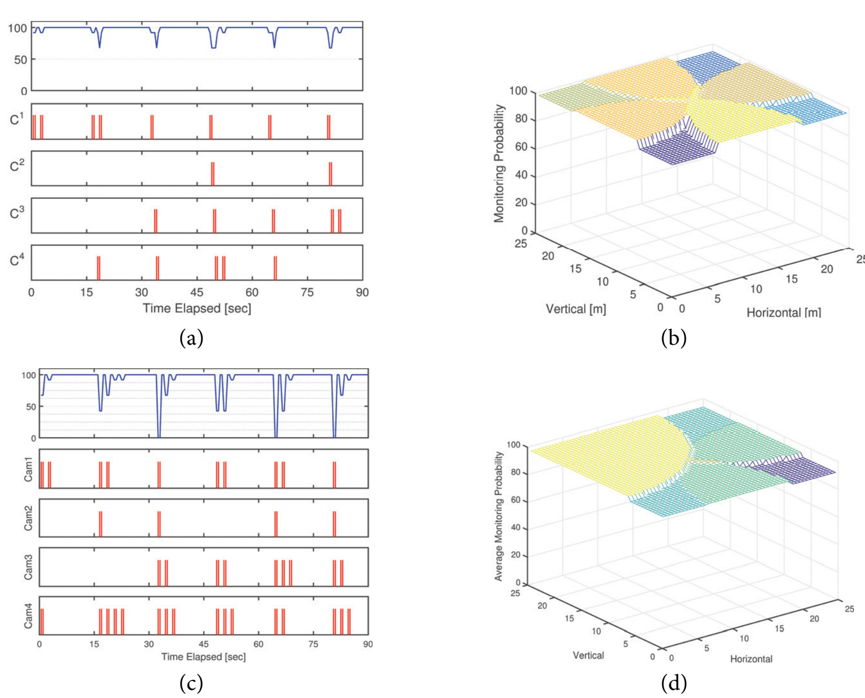



**Fig. 17.** Illustration of the active performance of active tracking with coverage constraints. (a) Asynchronous active tracking timing. (b) Asynchronous active tracking coverage. (c) Synchronous active tracking timing. (d) Synchronous active tracking coverage.

## 4.3 Rescheduling and Collaboration with Limited Cameras

In this section, the tracking performance is evaluated with the limited number of cameras. Each camera has three coverage positions and the position is selected by the proposed algorithm. The camera placement is given such that the overall area cannot be simultaneously covered with the given set of cameras at any given time. To cover the entire area with the limited set of cameras, the area should not be left uncovered more than the minimum time for coverage, $T_{mtc}$.

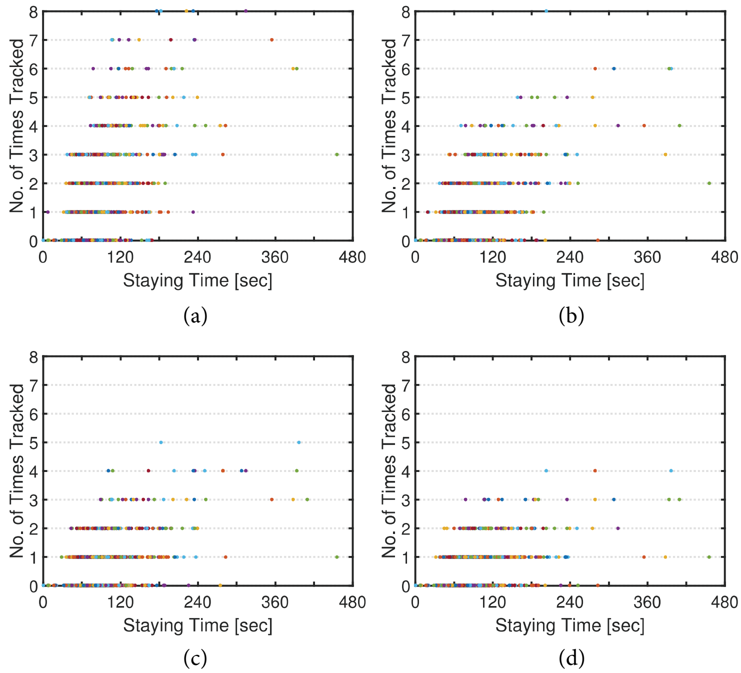



**Fig. 18.** Illustration of the active tracking profiles for the objects. Two cameras positioned 1 and 3 are used and 50 randomly moving objects are considered in the evaluation. (a) $T_{mtc}$ = 8 seconds, (b) $T_{mtc}$ = 16 seconds, (c) $T_{mtc}$ = 24 seconds, and (d) $T_{mtc}$ = 32 seconds.

Fig. 18 illustrates the overall active tracking performance with the cameras that are positioned at 1 and 3. Each point in the figure represents an object. The x-coordinate indicates the time the corresponding object left the area and the y-coordinate indicates the number of time the object was selected and tracked. As shown in the figure, as the objects stay longer period within the area, redundant active tracking is evident. As for the objects that stay within the area for shorter duration of time, there is a high chance that the active tracking may not occur. Because of the object distribution, many objects are missed due to short staying time. Note that the shorter value of $T_{mtc}$ generates a fewer miss active tracking but more redundant active tracking. A similar evaluation is done with different set of camera positions as illustrated in Fig. 19. Since the objects trajectories are diffused from one entering location, the active tracking performance depends on the camera locations.

Fig. 20(a) illustrates the cumulative tracking probability where the cameras are positioned at 2 and 4. In the evaluation, the values of $T_{mtc}$ are 8, 16, 24, 32 seconds and the translation speed of the cameras is 0.5 seconds. The cumulative probability is represented as a percentage, which depends on the amount of time the objects stayed within the area as well as the camera positions. Since the objects were diffused from one location, the performance depends on the camera position. In Fig. 20, 100% cumulative tracking probability is not possible since some of the object which entered with very short staying time is permanently missed by the system. This figure indicates that the smaller value of $T_{mtc}$ produces the lower the miss tracking. Fig. 20(b) illustrates the same evaluation with the faster translation time, 0.25 seconds. When the cameras are limited, the performance with the faster translation camera is highly desirable.

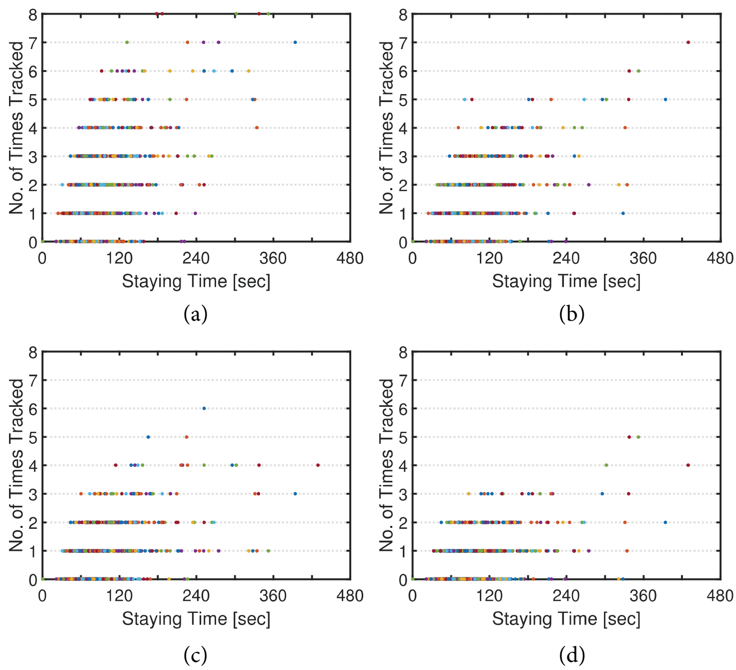



**Fig. 19.** Illustration of the active tracking profiles for the objects. Two cameras positioned at 2 and 4 are used and 50 randomly moving objects are considered in the evaluation. (a) $T_{mtc}$ = 8 seconds, (b) $T_{mtc}$ = 16 seconds, (c) $T_{mtc}$ = 24 seconds, and (d) $T_{mtc}$ = 32 seconds.

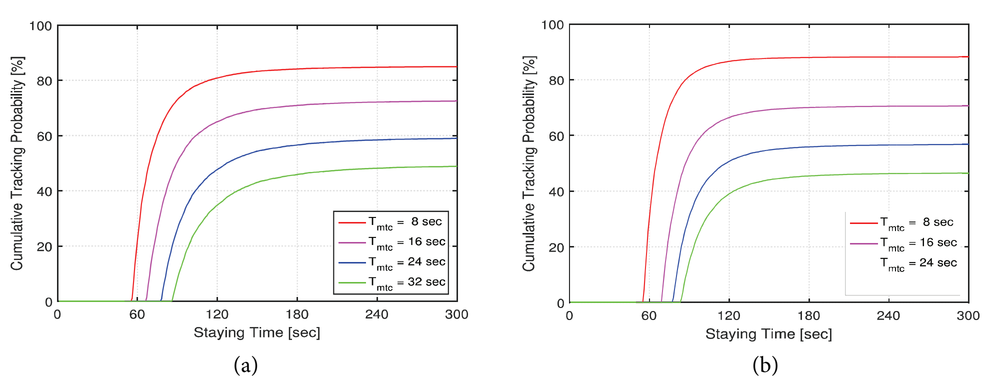



**Fig. 20.** (a) Illustration of the cumulative active tracking performance as a function of $T_{mtc}$ with the camera translation time of 0.5 seconds. (b) Illustration of the cumulative active tracking performance as a function of $T_{mtc}$ with the camera translation time of 0.25 seconds.

## 5. Conclusion

In this paper, a scalable multiple camera collaboration strategy for active tracking applications for large areas is proposed. The proposed method emulates the master-slave operation using multiple distributed cameras. The master and slave cameras are not designated but adaptively determined depending on the object density and distribution. Moreover, the number of cameras emulating the master is not fixed. Each camera maintains its own local information and the server combines the local information to construct the global information for optimum active tracking performance. Based on the global information, the number of active tracking operations are evenly distributed among cameras to maximizes active tracking of the dynamic objects. The dynamics of all objects visible in the local camera views are estimated for effective coverage scheduling of the cameras. The active tracking synchronization timing information is chosen to maximize the overall monitoring time for general surveillance operations while minimizing the active tracking miss. The performance of the proposed mechanism is evaluated with a densely populated situation. It is also demonstrated that faster translation time of the cameras greatly influences the overall active tracking performance.


## Acknowledgement

This work was supported Research Projects for Senior Researchers through the Ministry of Education of Korea and the National Research Foundation (No. 2017-2017008886). Also, this work was supported by the National Research Foundation of Korea grant funded by the Korea government (MSIP) (No. 2016015499).


## References


[1] J. Lee, H. Chae, and K. Hong, “A fainting condition detection system using thermal imaging cameras based object tracking algorithm,” *Journal of Convergence*, vol. 6, no. 3, pp. 1-15, 2015.

[2] M. S. Gaur and B. Pant, “Trusted and secure clustering in mobile pervasive environment,” *Human-centric Computing and Information Science*s, vol. 5, no. 32, pp. 1-17, 2015.

[3] S. H. Cho, Y. Nam, S. Hong, and W. Cho, ”Sector based scanning and adaptive active tracking of multiple objects," *KSII Transactions on Internet & Information Systems*, vol. 5, no. 6, pp. 1166-1191, 2011.

[4] T. B. Dinh, N. Vo, and G. Medioni, “High resolution face sequences from a PTZ network camera,” in *Proceedings of the IEEE International Conference on Automatic Face Gesture & Recognition and Workshops (FG 2011)*, Santa Barbara, CA, 2011, pp. 531-538.

[5] P. D. Z. Varcheie and G. A. Bilodeau, “People tracking using a network-based PTZ camera,” *Machine Vision and Applications*, vol. 22, no. 4, pp. 671-690, 2011.

[6] L. Zhao, L. F. Liu, Q. Y. Wang, T. J. Li, and J. H. Zhou, ”Moving target detection and active tracking with a multicamera network,” *Discrete Dynamics in Nature and Society*, vol. 2014, no. 2014, article ID 976574, 2014.

[7] C. M. Huang and L. C. Fu, “Multitarget visual tracking based effective surveillance with cooperation of multiple active cameras," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 234-247, 2011.

[8] M. Hoffmann, M. Wittke, Y. Bernard, R. Soleymani, and J. Hahner, “DMCtrac: distributed multi camera tracking,” in *Proceedings of the 2nd ACM/IEEE International Conference on Distributed Smart Cameras*, Stanford, CA, 2008, pp. 1-10.

[9] N. Krahnstoever, T. Yu, S. N. Lim, K. Patwardhan, and P. Tu, “Collaborative real-time control of active cameras in large scale surveillance systems," in *Proceedings of the Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2)*, Marseille, France, 2008.

[10] C. D. W. Ward and M. D. Naish, “Scheduling active camera resources for multiple moving targets,” in *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE 2009)*, St. John's, Canada, 2009, pp. 528-532.

[11] C. J. Costello, C. P. Diehl, A. Banerjee, and H. Fisher, “Scheduling an active camera to observe people," in *Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks (VSSN 2004)*, New York, NY, 2004, pp. 39-45.

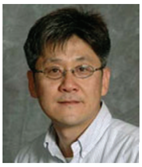
[12] F. Z. Qureshi and D. Terzopoulos, “Surveillance camera scheduling: a virtual vision approach,” *Multimedia Systems*, vol. 12, no. 3, pp. 269-283, 2006.

[13] A. Del Bimbo and F. Pernici, “Towards on-line saccade planning for high-resolution image sensing,” *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1826-1834, 2006.

[14] S. Khan, O. Javed, Z. Rasheed, and M. Shah, “Human tracking in multiple cameras,” in *Proceedings of 8th IEEE International Conference on Computer Vision*, Vancouver, Canada, 2001, pp. 331-336.

[15] S. Khan and M. Shah, “Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1355-1360, 2003.

[16] B. A. Sarif, M. T. Pourazad, P. Nasiopoulos, V. C. Leung, and A. Mohamed, “Fairness scheme for energy efficient H.264/AVC-based video sensor network,” *Human-centric Computing and Information Sciences*, vol. 5, no. 7, pp. 1-29, 2015.

**Sangjin Hong**

He received the B.S. and M.S. degrees in Electrical Engineering & Computer Science from the University of California, Berkeley. He received his Ph.D. in Electrical Engineering & Computer Science from the University of Michigan, Ann Arbor. He is currently with the department of Electrical and Computer Engineering at Stony Brook University, New York (SUNY). Before joining SUNY, he has worked at Ford Aerospace Corp. Computer Systems Division as a systems engineer. He also worked at Samsung Electronics in Korea as a technical consultant. His current research interests are in the areas of low power system design of multimedia wireless communications and digital signal processing systems, reconfigurable Systems on Chip design and optimization, and VLSI signal processing. Prof. Hong served on numerous Technical Program Committees for IEEE conferences. He is currently serving as an editorial board member in EURASIP Journal of Advances in Signal Processing, Journal of Signal Processing Systems, and Scientific World Journal. Prof. Hong is a Senior Member of IEEE.

**Kyungrog Kim**

He received his B.S. in Information and Computer Engineering from Ajou University in 1998 and his M.S. in Digital Media from Seoul University of Venture & Information in 2007. Also He received his Ph.D. in Computer Technology in 2012 from the Hoseo Graduate School of Venture. He joined the Department of Electronic Engineering at Hoseo University as an assistant professor. His current research interests include eLearning, learning analytics, data mining, and metadata.

**Nammee Moon** http://orcid.org/0000-0003-2229-4217

She received the B.S. degree in Computer Science from Ewha Womans University and the M.S. degree in Data Base from Ewha Womans University, in 1985 and 1987, respectively. Also she received the Ph.D. degree in Computer Technology in 1998 from Ewha Womans University. From 2000 to 2003, she served as an assistant professor of research in the Ewha Womans University. Then, she joined the Department of Digital Media at Seoul University of Venture & Information as an associate professor from 2003 to 2008. After that, she joined Hoseo University as a professor from 2008. Her current research interests include interactive digital business model, HCI, service oriented interactive video application, metadata and data mining.