Published as " Sina G. Davani, Musab Al-Hadrusi, and Nabil J. Sarhan. An Autonomous System for Efficient Control of PTZ Cameras. ACM Transactions on Autonomous and Adaptive Systems, Volume 16, Issue 2, June 2021, Article No.: 6, pp 1–22, DOI: https://doi.org/10.1145/3507658".

An Autonomous System for Efficient Control of PTZ Cameras

SINA G. DAVANI, Wayne State University, USA
MUSAB S. AL-HADRUSI, Wayne State University, USA
NABIL J. SARHAN*, Wayne State University, USA

This paper addresses the research problem of how to autonomously control Pan/Tilt/Zoom (PTZ) cameras in a manner that seeks to optimize the face recognition accuracy or the overall threat detection and proposes an overall system. The paper presents two alternative schemes for camera scheduling: *Grid-Based Grouping* (GBG) and *Elevator-Based Planning* (EBP). The camera control works with realistic 3D environments and considers many factors, including the direction of the subject's movement and its location, distances from the cameras, occlusion, overall recognition probability so far, and the expected time to leave the site, as well as the movements of cameras and their capabilities and limitations. In addition, the paper utilizes clustering to group subjects, thereby enabling the system to focus on the areas that are more densely populated. Moreover, it proposes a dynamic mechanism for controlling the pre-recording time spent on running the solution. Furthermore, it develops a parallel algorithm, allowing the most time-consuming phases to be parallelized and thus run efficiently by the centralized parallel processing subsystem. We analyze through simulation the effectiveness of the overall solution, including the clustering approach, scheduling alternatives, dynamic mechanism, and parallel implementation in terms of overall recognition probability and the running time of the solution, considering the impacts of numerous parameters.

CCS Concepts: • **Applied computing → Surveillance mechanisms**; • **Computing methodologies → Planning and scheduling**.

Additional Key Words and Phrases: Automated Video Surveillance, Autonomous Control of PTZ Cameras, Camera Scheduling, Clustering, Face Recognition.

ACM Reference Format:
Sina G. Davani, Musab S. Al-Hadrusi, and Nabil J. Sarhan. YYYY. An Autonomous System for Efficient Control of PTZ Cameras. *ACM Trans. Autonom. Adapt. Syst.* 1, 1, Article 1 (January YYYY), 22 pages.

1 INTRODUCTION

PTZ cameras have large fields-of-regard (FoR), which can be defined as the set of all possible combinations of the field-of-view (FoV) that can be reached using various settings. The FoV is the extent of the observable view as seen by a camera using a single specific setting. Top-tier video systems employ PTZ cameras, but these cameras are currently controlled by humans or by predetermined scanning tours. The automated control of cameras to provide enhanced security has been a major research problem [11, 12, 19, 25, 27]. That prior work focused on improving metrics,

*This is the corresponding author

Authors' addresses: Sina G. Davani, sina@wayne.edu, Wayne State University, 5050 Anthony Wayne Dr., Detroit, MI, 48202, USA; Musab S. Al-Hadrusi, Wayne State University, 5050 Anthony Wayne Dr., Detroit, MI, 48202, USA, Musab.Hadrusi@wayne.edu; Nabil J. Sarhan, Wayne State University, 5050 Anthony Wayne Dr., Detroit, MI, 48202, USA, nabil.sarhan@wayne.edu.

1556-4665/YYYY/1-ART1 $15.00
https://doi.org/

such as the captured resolution and the number of times a subject is viewed, but we believe that the overall threat detection or recognition accuracy is the ultimate metric in security applications. In addition, that prior work focused on how subjects can be assigned to cameras. In contrast, we propose an autonomous system that seeks to optimize the overall recognition accuracy. As will be discussed later in this section, since this metric depends, among other factors, on reducing the time complexity of the optimization solution and thus improving the system scalability, we consider a broad problem domain, including parallel processing, clustering, timing, and search-space pruning.

This paper addresses the following two main research problems: how to design an overall system for autonomously controlling the PTZ cameras and how to control these cameras in a manner that seeks to optimize the overall threat detection or recognition accuracy. Specifically, considering a security system with multiple PTZ cameras, we seek to determine a plan of camera settings (pan, tilt, and zoom) over time for each camera to optimize the accuracy. The paper focuses on face recognition accuracy but can be extended to other Computer Vision (CV) algorithms. Towards this end, we present and analyze two alternative schemes for camera scheduling: *Grid-Based Grouping* (GBG) and *Elevator-Based Planning* (EBP). These schemes work in 3D environments and consider many factors, including the direction of the subject's movement and its location, distances from the PTZ cameras, occlusion, overall recognition probability so far, and the expected time to leave the site, as well as the movements of cameras and their capabilities and limitations. The camera movement is broadly defined here as the time required to reach the assigned pan, tilt, zoom, and focus settings.

As illustrated in Figure 1, the proposed system employs wide-angle cameras, PTZ cameras, and a centralized parallel processing subsystem. The wide-angle cameras monitor the overall site and send the videos to the processing subsystem, which then identifies the subjects and determines their positions, speeds, and direction of movements. The PTZ cameras capture higher-quality frames of the subjects and send them to the processing subsystem for further analysis, including running the necessary CV algorithms (such as face recognition). Running the CV algorithm on the centralized parallel processing subsystem (rather than the cameras) has many advantages: offering more flexibility in the selection of the cameras to be deployed and reducing their costs and power consumption (especially if the cameras are battery-powered). Additionally, a proxy station is needed regardless to gather and archive the streams from all the cameras. This system operates in a cyclic manner which involves switching from a pre-recording phase to a recording phase. During the pre-recording phase, the system determines the plans for controlling the cameras, whereas in the recording phase the cameras operate according to these plans.

To optimize the recognition accuracy, the system should spend more time capturing (recording phase) and less time planning (pre-recording phase). Hence, the paper proposes a dynamic mechanism for determining the pre-recording time and parallelizes the camera scheduling algorithms so that they run efficiently by the processing subsystem. The paper also utilizes *clustering* to group subjects, thereby enabling the system to focus on the areas that are more densely populated and thus further improve the recognition probability and running time. The paper further addresses the time complexity by proposing enhancements, including *step-sizes of camera settings* and *search-space pruning*, as discussed in Section 3.

We analyze through simulation the effectiveness of the overall solution and the included clustering approach, scheduling alternatives, dynamic mechanism for the pre-recording time, and the parallel implementation in terms of the overall recognition probability and the computation time of the solution. Although the main metric is the recognition probability, we also report the computation time as it impacts the recognition probability. In contrast with [11, 12, 14, 19, 25, 27], we conduct evaluations in 3D environments and utilize a realistic pedestrian speed model [22]. We analyze the impacts of many parameters on performance, including the number of PTZ cameras, site area,

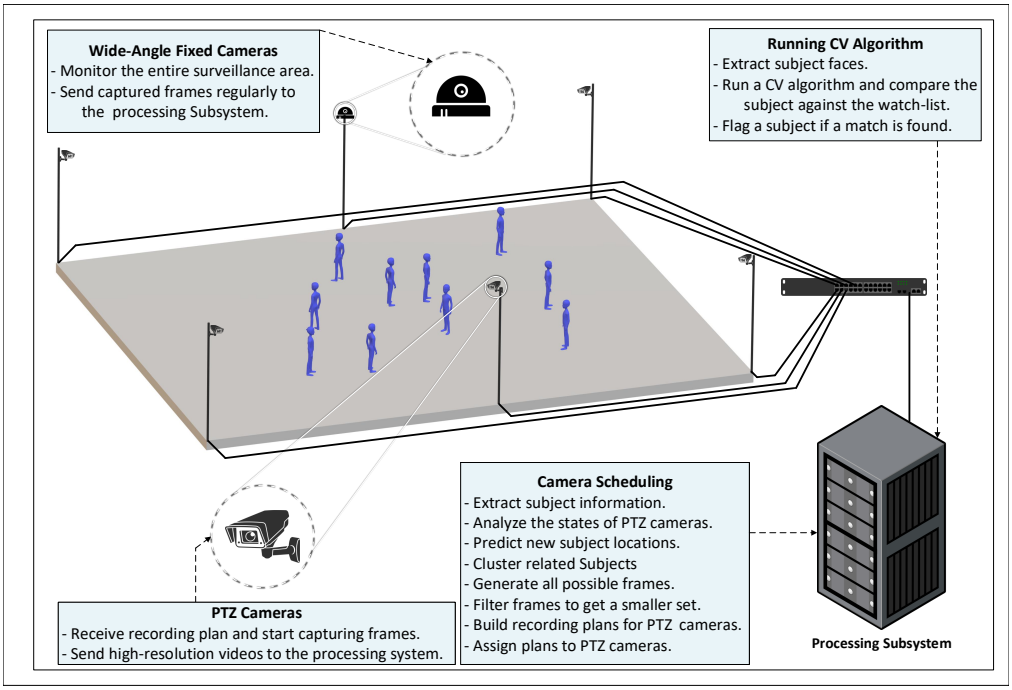


Fig. 1. An Illustration of the Proposed System

subject arrival rate, parallelism (single-threading or multi-threading), and pre-recording time (both dynamic and static). The main results show that EBP scheduling improves the subject recognition probability by 85% on average compared with GBG, while reducing the computational time by 15%. The clustering approach further improves the recognition probability by a factor of 2 on average and reduces the computational time by 60%. The dynamic mechanism provides an additional improvement of more than 80% on average in the recognition probability, while increasing the computational time by only 15%. Finally, using 8 threads on a single 4-core Intel Core i7 workstation (with 8 virtual cores), parallelizing the most-time consuming phases reduces the time of the overall solution by a factor of three or more on average, while improving the subject recognition probability by 20%.

The **main contributions** of this paper can be summarized as follows: (1) presenting an overall system design for controlling the PTZ cameras autonomously, (2) presenting an autonomous solution for optimizing the overall recognition accuracy, (3) presenting two alternatives for camera scheduling, (4) utilizing clustering for grouping subjects, thereby enabling the system to focus on the areas that are more densely populated, (5) proposing a dynamic mechanism for controlling the pre-recording time spent on running the solution, (6) developing a parallel algorithm, allowing the solution to run efficiently by the processing subsystem, and (7) analyzing the effectiveness of the overall solution and the included components in recognition probability, considering 3D environments and utilizing a realistic pedestrian speed model.

The rest of the paper is organized as follows. Section 2 discusses the background information and related work. Section 3 presents the proposed solutions, including the formulation of the optimization problem. Section 4 details the performance evaluation methodology. Finally, Section 5 presents and analyzes the main results.

2 BACKGROUND INFORMATION AND RELATED WORK

Most prior studies on automated video surveillance focused on developing robust computer vision algorithms for detection, tracking, and classification [1, 2, 4, 23] (and references within). Face recognition algorithms generally use *statistical* or *deep learning* approaches [9]. The latter can yield better accuracy by using *Convolutional Neural Networks* (CNNs) and include ArcFace [4], SphereFace [15], and FaceNet [23].

Prior work on automatic control focused specifically on how to assign subjects to cameras. Zhou et al. [28] used a fixed wide-angle camera to observe the overall scene and send the video frames to a server, which in turn issues commands to control the PTZ cameras. In that system, each PTZ camera was assigned to exactly one subject. Krahnstoever et al. [12] used pre-calculated plans to schedule cameras. Xu et al. [25] investigated subject grouping using *lattice-based grouping* (LBG). Sinnu et al. [24] addressed the maximization problem of subject detection by video summarization. Lewis et al. [13] analyzed different behavioral strategies in self-organizing smart camera networks in terms of tracking performance, considering homogeneous and heterogeneous configurations of the cameras. Mali et al. [17] considered Wireless Multimedia Sensor Networks (WMSNs) involving camera and scalar sensor nodes for providing information about events occurring in the monitored region. They proposed a topology management-based distributed camera actuation scheme for prolonging the lifetime of scaler sensor nodes and decreasing the data reporting latency. Piciarelli et al. [20] proposed a general, high-level framework for network reconfiguration and presented a short survey of some of the most relevant state-of-the-art works in the field, showing how they can be reformulated in their framework. Esterle et al. [6] also studied the performance of decentralized and self-organized approaches in comparison to centralized ones in terms of geometric coverage maximization.

In contrast with prior work, we focus on the system design and provide a complete and viable solution that seeks to optimize the CV accuracy. Moreover, the aforementioned studies did not consider the parallel processing aspects, had limiting assumptions, and considered small subsets of the factors, as will be elaborated next. Nevertheless, the work of Xu et al. [25] is the most related because their LBG method deals with scheduling subjects to different cameras and thus provides an alternative to the proposed GBG and EBP methods in this paper. Hence, we provide a comparative analysis with that method in Subsection 5.3.

3 PROPOSED SYSTEM AND SOLUTIONS

Since the proposed solution seeks to optimize the overall recognition accuracy, we first formulate the optimization problem in Subsection 3.1 and then provide an overview of the proposed system in Subsection 3.2. Subsequently, we present the individual parts of the solution in detail.

3.1 Formulation of the Optimization Problem

To formulate the overall optimization problem, we need to model the face recognition accuracy. Table 1 summarizes and explains most notations used in this paper. The major factors affecting the recognition accuracy include (*i*) resolution, (*ii*) pose, (*iii*) occlusion, and (*iv*) zoom-distance noise. It is generally best to limit the pose angles (angles between face direction and the camera's pan and tilt axes) to 35°or even 25°relative to pan and tilt. The camera's zoom should also be adjusted to achieve a minimum resolution. In particular, the interocular distance (between eye pupils) should at least be larger than a minimum number (such as 60 or 120 pixels [3]). Far subjects require higher zooms, leading to a smaller FoV and thus fewer covered subjects in the frame. Additionally, zooming-in for long-range distances produces a blurring noise [26]. Although some image enhancement algorithms may be applied, avoiding longer zooms is preferable. Finally, partially occluded faces

Table 1. Summary of Notations

Notation	Description
F_{max}	The number of selected top projected frames for each camera (used in plan building)
P	Aggregate recognition probability of all subjects
$P_{rec}(i)$	The overall recognition probability of subject i so far
$P_{rec}(i,j)$	The recognition probability of subject i captured by camera frame j
$P_{tot}(j)$	Total recognition probability of all subjects included in camera frame j
$R_{pose}(\theta)$	The recognition probability R with respect to θ ($\theta \in \{\theta_{pij}, \theta_{tij}\}$)
R_{thresh}	Threshold after which the subject is considered to be sufficiently recognized
$R_{zoom}(z_{ij})$	The recognition probability R with respect to zoom z_{ij}
S_{i_T}	The number of times subject i is captured
T_e	Time spent on recording one frame
T_{leave}	The subject's expected time to leave the site
$T_{move,cam\rightarrow currentFrame}$	Time to move camera cam to capture projected frame $currentFrame$
T_{MSL}	Required pre-recording time for the best-selected plan for all the cameras
T_{now}	The current time
T_P	Pre-recording time
T_{PR}	Unused available time in the pre-recording phase
T_R	Recording phase time
T_{RR}	Unscheduled available time in the recording phase
T_{SL}	Required pre-recording time for the best-selected plan so far for one specific camera
θ_{pij}	The angle between subject i (face center) and the optical axis of camera j (pan-wise)
θ_{tij}	The angle between subject i (face center) and the optical axis of camera j (tilt-wise)
$u_{cov}(i,j)$	Subject i coverage in frame j
$u_{occ}(i,j)$	Subject i occlusion in frame j
$W(i)$	The weighting factor for subject i, used in the recognition probability calculation
z_{ij}	The zoom adopted by camera j to capture subject i

are also problematic. Although many face recognition algorithms try to address this issue [29], it is more beneficial to avoid capturing occluded faces from the beginning [21].

We formulate the recognition probability $P_{rec}(i,j)$ of subject i captured by camera frame j as follows:

$$P_{rec}(i,j) = R_{pose}(\theta_{tij}) \times R_{pose}(\theta_{pij}) \times R_{zoom}(z_{ij}) \times u_{cov}(i,j) \times u_{occ}(i,j), \tag{1}$$

where R_{pose}, R_{zoom}, u_{cov}, and u_{occ} are functions that can be derived to capture the impacts of the pose, zoom, coverage, and occlusion, respectively. Note that the same subject may be viewed by multiple cameras from different views. Based on the empirical data in [8, 12] and by limiting the poses to ±25°pan or tilt boundaries, we derive the following model for the impact of pose on recognition probability:

$$R_{pose}(\theta_{tij}) = \begin{cases} e^{(-\frac{\theta_{tij}}{b_1})^2} & |\theta_{tij}| \leq 25 \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where θ_{tij} is the angle between subject i (face center) and the optical axis of camera j in the tilt direction, and b_1 is a constant. A similar function can be derived for the pan angle (θ_{pij}), but with a different constant. From the empirical data in [26], we can model the zoom as a linear function of the distance, and then based on the empirical data in [26], we derive the following model for the relationship between the recognition probability R and the zoom z:

$$R_{zoom}(z_{ij}) = a \times e^{(b_2 \times z_{ij})}, \tag{3}$$

where z_{ij} is the zoom adopted by camera j to capture subject i with a resolution of 60 inter-ocular pixels, and a and b_2 are constants.

Since a subject may be out of a camera's FoV or occluded by structures or other subjects, functions $u_{cov}(i, j)$ and $u_{occ}(i, j)$ can be used to assess the coverage and occlusion of the subject, respectively:

$$u_{cov}(i, j) = \begin{cases} 1 & \text{if subject } i \text{ is fully covered by frame } j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$u_{occ}(i, j) = \begin{cases} 1 & \text{if subject } i \text{ is not occluded in frame } j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

We form a utility function $P_{tot}(j)$ to measure the total recognition probability of all subjects included in camera frame j: $P_{tot}(j) = \sum_{i=1}^{N_s} P_{rec}(i, j)$, where N_s is the number of subjects in frame j and $P_{rec}(i, j)$ is the recognition probability of subject i captured by camera frame j. This equation sums the recognition probability value contributed by each covered subject i in frame j. Any mentioning of the frame recognition probability in this paper refers to this equation.

We propose to assign a weighting factor $W(i)$ to each subject i with the weight function being formulated to reflect the assessed threat level in the subject's location, the expected time for the subject to leave the site, and the overall recognition probability of the subject so far.

As a subject may be captured by multiple cameras, we need to capture the necessary data fusion. Assuming N_c cameras each taking a frame, the overall probability of subject i can be estimated by $(1 - \prod_{j=1}^{N_c}(1 - P_{rec}(i, j)))$. The optimization problem can be specified as maximizing the overall weighted recognition probability of all subjects by finding the set of all PTZ settings of cameras over time (***CamSettings***), considering a set of constraints in camera placement (***CamPlacement***), camera capabilities (***CamCapabilities***), and subject attributes over time (***SubjectAttributes***):

$$\begin{aligned} \underset{\boldsymbol{CamSettings}}{\text{Maximize}} \quad & \sum_{i=1}^{N_s} W(i) \times (1 - \prod_{j=1}^{N_c}(1 - P_{rec}(i, j))) \\ s.t. \quad & \{\boldsymbol{CamPlacement}, \boldsymbol{CamCapabilities}, \boldsymbol{SubjectAttributes}\}. \end{aligned} \tag{6}$$

A brute-force approach examining all the possible camera frames should generate and examine all possible PTZ settings for each camera. As this approach is time-prohibitive, we present an efficient solution for this problem.

3.2 Overview of the Proposed System and Solution

The main objective of the proposed solution is to optimize the overall subject recognition probability by controlling the PTZ settings of various cameras, based on the direction of the subject's movement and its location, distances from the cameras, occlusion, overall recognition probability so far, and the expected time to leave the site, as well as the movements of cameras and their capabilities and limitations.

As illustrated in Figure 1, the proposed system employs wide-angle and PTZ cameras and a processing subsystem. The wide-angle cameras monitor the overall site and send the videos to a processing subsystem, which determines the positions, speeds, and direction of movements of all subjects in the site [25, 28] and subsequently controls the PTZ cameras by solving the optimization problem (Equation (6)). The PTZ cameras then capture higher-quality frames of the subjects. The processing subsystem also runs the CV algorithms (including face recognition) on the frames received from the PTZ cameras. That subsystem is centralized with the work being distributed among multiple processors with minimal communication overhead.

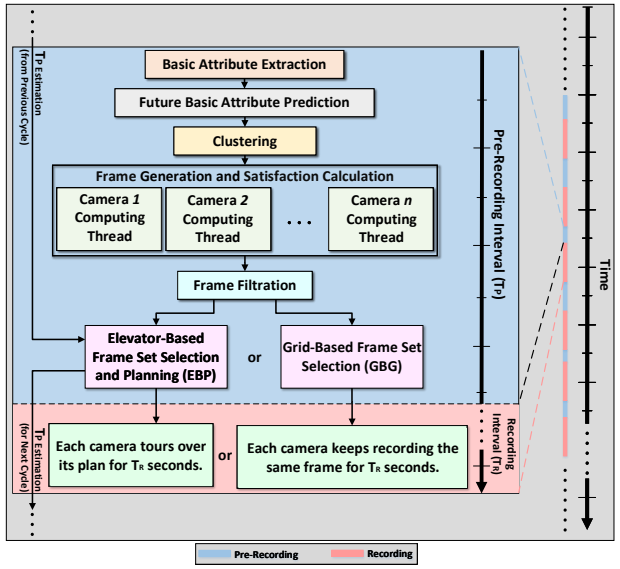


Fig. 2. Illustration of the Proposed Solution

As illustrated in Figure 2, this solution works in a cyclic mode, with each cycle being divided into two phases: *pre-recording period* (T_p) and *recording period* (T_R), as in [25]. T_p is the preparation time required by the processing subsystem to run all phases of the camera control solution. In contrast, during T_R, the PTZ cameras capture the subjects using the assigned settings and send them to the processing subsystem for analysis. T_p and T_R can be pre-determined, but we present a dynamic mechanism for T_p in Subsection 3.6, resulting in potentially different values in different cycles. The dynamic mechanism employs a feedback loop for determining T_p of the next cycle based on the extracted information from the current cycle.

As illustrated in Figure 2, the proposed camera control solution operates in the following stages.

- *Basic Attribute Extraction:* The system utilizes the information provided by the fixed wide-angle cameras to detect all captured subjects. It then uses one of the widely researched pedestrian detection algorithms [5] to determine their *basic attributes*, including, location, speed, and direction of movement.
- *Future Basic Attribute Prediction:* The processing subsystem forecasts the locations of the subjects when the PTZ cameras are scheduled to begin the next recording period based on the current speeds and directions of movement of the subjects.
- *Subject Clustering:* As will be detailed in Subsection 3.3, we utilize clustering to group subjects based on different attributes, thereby enabling the system to focus on the areas that are populated with more subjects.
- *Frame Generation:* The system then uses the prediction results to generate the set of all possible *projected frames* that can be captured by each PTZ camera through the examination of different combinations of the camera settings, utilizing the actual capabilities and limitations of the PTZ cameras. We define the projected frame as the expected frame that will be captured by a PTZ

camera with a specific PTZ setting. Therefore, the projected frame is the predicted projection of the view as seen by the camera at a specific FoV. Additional details are given in Subsection 3.4.

- *Satisfaction Calculation:* The system predicts the *recognition-influencing* attributes of the subjects for each generated projected frame and then determines the expected overall recognition probability gained by that frame using Equation (1). The recognition-influencing attributes include the pose angle, interocular distance in pixels, and occlusion. By examining the different camera settings, the set of projected frames encompassing the entire fields-of-regard (FoR) from all PTZ cameras will be assessed. In this stage, a projection triangulation is used to estimate the projected 2D frame characteristics from the 3D site. Rotation and translation matrices are used to map the 3D coordinates to each camera coordinates [7]. Additional details are given in Subsection 3.4.
- *Frame Filtration:* After generating the sets of all possible projected frames, we filter these frames by eliminating the frames that will not capture any subject and by selecting only the best projected camera frame among the sets of frames that capture the same set of subjects. The selected projected frame is the one that achieves the maximum aggregate recognition probability of all subjects. Therefore, effective pruning of the search space is achieved.
- *PTZ Camera Scheduling:* Using the set of filtered projected frames, the system then determines the best set of PTZ camera settings that achieve the best overall recognition accuracy (considering all subjects). The proposed schemes for this task are detailed in Subsection 3.5. When the dynamic mechanism is utilized, the pre-recording time estimation from previous cycles is incorporated in building the plan.
- *Recording and Tracking:* The PTZ cameras apply the assigned settings by camera scheduling to start recording and tracking these subjects.

We address the computational complexity by the clustering approach (Subsection 3.3), search-space pruning through projected frame filtration, examining the PTZ camera settings in pre-configured step-sizes, and presenting a parallel solution (Subsection 3.4). The dynamic mechanism, clustering approach, and parallelism allow the system to properly handle changes in the monitored environment.

3.3 Subject Clustering

To populate projected frames efficiently, we utilize clustering as a pre-step for generating these frames. We define the cluster by three attributes: (*i*) cluster center, (*ii*) cluster direction of movement, and (*iii*) cluster size. A subject is placed in a cluster if the following conditions are met: (1) the distance between the cluster center and the subject is within threshold $DIST\_TH$, (2) the angular offset between the direction of movement of the subject and that of the cluster center is within threshold $ANGLE\_TH$, and (3) the perpendicular distance between the subject and the cluster direction of movement is within threshold $WIDTH\_TH$.

Grouping subjects can utilize one of a wide variety of clustering algorithms [10], such as *Lloyd's*. That algorithm maps N nodes to K centers by proceeding into two steps: (1) assigning nodes to clusters by finding the nearest cluster center to each node and (2) computing the new cluster center. As the number and the locations of the K centers are unknown in the considered clustering problem, we use a variation of *Dendrogram Hierarchical Clustering* [16], by which each entity is initially considered as a cluster center. Subsequently, each pair of nodes with a minimum in-between distance are grouped to form one cluster. We then determine the centers for the newly formed clusters and repeat the process. To predict recognition probability in the recording phase, cluster trajectory is considered. Clusters are dynamically reevaluated and re-calculated in each cycle.

```
01. //Input: subjectList[], List of the active subjects
02. //Output: clusterList[], Cluster center list of the grouped subjects
03. Find-N-Clusters(){
04.   for (i = 0; i < subjectList.size(); i++){
05.     clusterList[i].Center = subjectList[i].Loc;
06.     clusterList[i].Dir = subjectList[i].Dir; // Direction angle
07.   }
08.   for (i = 0; i < clusterList.size(); i++){
09.     mergIndex = −1;
10.     for (j = 0; j < clusterList.size(); j++){
11.       Distance = DIST(clusterList[i].Center, clusterList[j].Center);
12.       // Calculate the perpendicular distance
13.       DistancePer = DISTPER(clusterList[i].Center, clusterList[j].Center);
14.       Direction = DIR(clusterList[i].Dir, clusterList[j].Dir);
15.       if (Direction > ANGLE_TH || Distance > DIST_TH || DistancePer > WIDTH_TH)
16.         continue;
17.       totalDist = Distance/DIST_TH + DistancePer/WIDTH_TH + Direction/ANGLE_TH;
18.       if (j == 0)
19.         minDist = totalDist;
20.       if (totalDist > minDist)
21.         continue;
22.       minDist = totalDist;
23.       mergIndex = j;
24.     } End of loop j
25.     if (mergIndex == −1)
26.       continue;
27.     clusterList[i] = Merge(clusterList[i], clusterList[mergIndex])
28.     clusterList[i].Center = Average(clusterList[i].Center, clusterList[mergIndex].Center)
29.     clusterList[i].Dir = Average(clusterList[i].Dir, clusterList[mergIndex].Dir)
30.     Remove(clusterList[j])
31.   } End of loop i
32. } // End of Find-N-Clusters()
```

Fig. 3. Simplified Algorithm for Cluster Formulation

Figure 3 shows a simplified algorithm for cluster formulation. Lines 11 - 14 calculate the Cartesian distance, the perpendicular distance, and the angular offset. Figure 4 illustrates the cluster formulation process. Cluster $R1$ is initially formed and then an additional subject at location A is added to form cluster $R2$. X_A and X_B represent the Cartesian distances between the center of cluster $R1$ and the subjects at locations A and B, respectively, while X_C and X_D represent the corresponding perpendicular distances. The subjects at locations F and G cannot be added to $R2$ because the angular offsets for their directions of movement are larger than $ANGLE\_TH$ relative to that of the cluster center. Moreover, the subject at location B cannot be added either because it has a perpendicular distance X_D that is larger than $WIDTH\_TH$.

3.4 Projected Frame Generation, Satisfaction Calculation, and Their Parallel Implementation

After clustering the subjects, the system generates the set of all possible projected frames and then determines the expected subject recognition probability achieved by each projected frame. As the synchronization and communication among different parallel threads of execution should be minimized to achieve high performance, we examined hundreds of shared global variables used in various phases to determine if they can be made local. Based on extensive timing analysis (shown in Subsection 5.2), we found out that the projected frame generation and satisfaction calculation phases contribute the most to the computational time of the overall solution. Accordingly, we parallelize these phases, as shown in Figure 2, and set the granularity of parallel implementation as the number of cameras in the system to improve system scalability.

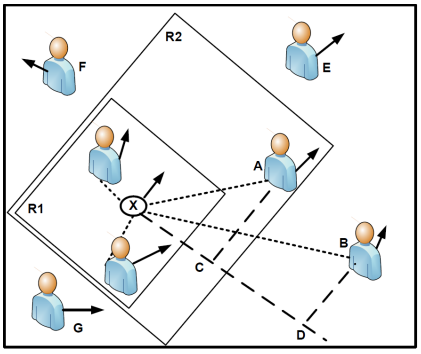


Fig. 4. Illustration of Cluster Formation

```
01. //Input: subjectList[], clusterList[], cameraList[]
02. //Output: frameList[], a projected frame list
03. generateFrameList(){
04.   simultaneouslyFor (c = 0; c < cameraList.size(); c++){
05.     FindOcclusion(subjectList); // Location
06.     for (i = 0; i < clusterList.size(); i++){ // for each cluster
07.       tempFrame.RecProb = 0;
08.       for (j = 0; j < subjectList.size(); j++){ // for each subject
09.         if (subjectList[j].Invalid()) continue;
10.         //add the subject to the projected frame
11.         tempFrame.subList.push(subjectList[j]);
12.           //add recognition prob. of the subject to the projected frame
recognition prob.
13.         tempFrame.RecProb += subjectRecProb(subjectList[j]);
14.       } End of loop j
15.       tempFrame.PTZ = FindPTZ(cameraList[c],tempFrame.subList);
16.       frameList.Add(tempFrame)// add current projected frame to
the projected frame list
17.     } End of loop i
18.   } End of loop c
19. } // End of populateFrameList()
```

Fig. 5. Simplified Algorithm for Projected Frame Generation and Satisfaction Calculation

Figure 5 shows a simplified parallel algorithm for projected frame generation and satisfaction calculation. It simultaneously generates all projected frames for each camera, considering its capabilities. It also determines the overall subject recognition probability for each projected frame (as discussed in Subsection 3.1). We pick the cluster centers and identify the camera that can cover the included subjects with the best overall recognition probability. By considering all the subjects for every cluster, we effectively re-map the subjects in the provided clusters to form more precise groups, without changing the cluster centers. Further, the algorithm results in generated sub-clusters with the same centers and higher resolutions.

3.5 Proposed PTZ Camera Scheduling

We present two schemes for camera scheduling: *Grid-Based Grouping* (GBG), and *Elevator-Based Planning* (EBP).

3.5.1 GBG Scheme. This strategy is an adaptation of the lattice-based approximation algorithm in [25] with several enhancements. That algorithm finds a projected frame-set that maximizes the overall satisfaction by discretizing the solution space and checking for minimum overlap among the selected projected frames, with the satisfaction computed based on the captured resolution and coverage. It assumed a frame domain derived from 2D scenes and that all cameras can capture the same set of frames in the same fashion. In addition to optimizing the recognition probability, we enhance that algorithm in the following other aspects: (*i*) dealing with cameras having different FoRs, (*ii*) generating projected frames according to cameras capabilities and not site area dimensions, (*iii*) grouping based on the 3D scene by considering each camera's FoV and the locations of subjects, and (*iv*) assessing the overlap among projected frames in 3D scenes.

3.5.2 EBP Scheme. This scheme refines the filtered projected frames and uses them to build detailed plans for each camera. It captures the current settings of each camera, and its moving, zooming, and focusing speeds. In contrast with GBG, EBP allows each camera to view a different set of subjects at different times during a single recording period rather than repeatedly tracking the same subjects. This algorithm is inspired by the elevator algorithm for hard disk drives as it seeks to reduce the

00. *// Input: Filtered Projected Frames Domain*
00. *// Output: $planSeeds_{arr}$ list of top frame-sets*
01. **chooseFrames()**{
02. Sort each camera's projected frames according to their recognition probability ψ;
03. Get best F_{max} projected frames for each camera;
04. Generate all $\binom{F_{max}}{seeds}$ frame-sets combinations;
05. Compute overall recognition probability $\sum_{j=1}^{seeds} \psi(j)$ for each set combination per each PTZ camera;
06. Choose the best set of seeds for each PTZ camera and store them in $planSeeds_{arr}$;
07. } *// end chooseFrames()*

Fig. 6. A Simplified Algorithm to Generate Seed Frames for Building Plans in EBP

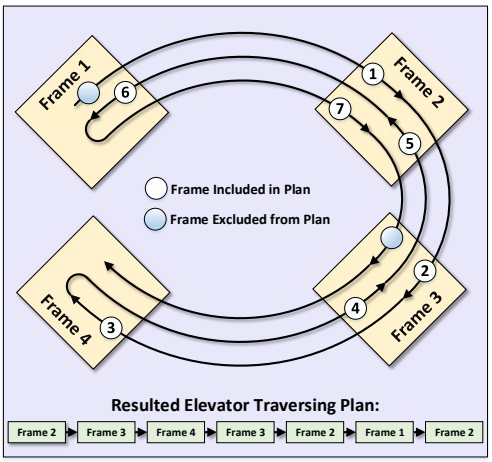


Fig. 7. Creating a Plan by Using a Seed Frame-Set

camera movement time (i.e., the time to apply the PTZ setting) like the classical algorithm, which minimizes the disk head movement [18].

As shown in Figure 6, EBP proceeds into two main stages: (*i*) generating plan seeds for each PTZ camera (Lines 1-4) and (*ii*) building the plan for each PTZ camera from these seeds (Lines 5-7). The algorithm sorts projected frames in each camera list according to their overall recognition probabilities $P_{tot}(j)$ and picks only the best F_{max} frames for each camera to reduce the complexity of subsequent steps. F_{max} can be selected so that the computational time of the overall solution is within T_p seconds. For each camera, the algorithm then selects a subset of *seeds* frames (where $seeds < F_{max}$) out of the F_{max} frames such that this subset achieves the best recognition probability of all subjects. The aggregate probability is measured by considering the recognition probability of each subject only once at its maximum occurrence in the subset:

$$P = \sum_{i}^{S_N} \sum_{j}^{Seeds} P_{rec}(i,j) \times u(i)_{max}, \tag{7}$$

where *seeds* is the number of projected frames in the subset, S_N is the number of subjects in the subset, and $u(i)_{max}$ for subject i is 1 only when its $P_{rec}(i)$ is at its maximum-value occurrence in the subset and 0 otherwise. The subset of projected frames is stored in the array $planSeeds_{arr}$.

Subsequently, EBP generates the plan list for each camera based on the selected subset of projected frames ($planSeeds_{arr}$). EBP considers the movement time spent by a camera to *reach* a projected frame by applying the necessary PTZ setting to capture it. This time can be given by

$$T_{move} = max(T_{Pan}, T_{Tilt}, T_{Zoom}) + Focus_{time}. \tag{8}$$

Note that the camera can work on each P, T, and Z setting simultaneously, and thus the maximum time of them is used and added with the focus time to determine the overall movement time. EBP assigns a plan for each specific camera separately by examining all possible plans formed with different starting projected frames and then selecting the plan achieving the maximum recognition probability. The first projected frame in the plan must be reached before the end of the current pre-recording time. Besides, the first projected frame must be captured for t_e seconds and all other frames in the plan must be reached and then captured each for t_e seconds during the next recording time.

Figure 7 demonstrates the elevator-based plan-building process using a seed frame-set with four elements. Frame 1 is not eligible as the starting projected frame because there is inadequate time left in the pre-recording time to reach it from the present PTZ camera setting, but Frame 2 is eligible. EBP continues adding seed projected frames to the plan as long as the recording time permits. It scans the projected frames clockwise and counter-clockwise until the process finishes at frame 3 (specifically its third scanning occurrence) as the recording time no longer permits. The same process is repeated using all other eligible starting projected frames, and finally, the plan achieving the highest recognition probability is selected and adopted by the camera during the next recording period. Examining more recording cycles leads to additional computational cost and less accurate predictions as the future time is further ahead of the current pre-recording phase.

Subsection 3.7 presents and discusses the EBP algorithm utilizing the dynamic mechanism presented in the following subsection.

3.6 Proposed Dynamic Mechanism for Determining the Pre-recording Time

We introduce a mechanism to determine the pre-recording time dynamically and in a manner that works well with the parallel implementation. Figure 8 illustrates the dynamic approach. As discussed earlier, T_P is determined based on the anticipated computation time of the camera control solution. If the computation time turns out to be shorter than anticipated, we propose starting the recording phase immediately. Due to parallelism, the algorithm time for different cameras may take different times. Thus, the recording time can start only after the current *Maximum System Lead Time* (T_{MSL}), which represents the longest *actually-spent* time running the control solution for different cameras. To improve the accuracy of predicting the subject attributes (such as locations) at the next recording time, T_P is adjusted dynamically based on the accuracy of its previously anticipated value:

$$T_P^{(new)} = ({T_{MSL}}^{(current)} + C \times T_P^{(current)})/2, \tag{9}$$

where $T_P^{(new)}$ is the new estimated value of the pre-recording time, ${T_{MSL}}^{(current)}$ and $T_P^{(current)}$ are the actually-spent and estimated values for the current period, respectively, and C is a constant.

As illustrated in Figure 8, Camera 2 effectively sets this value and thus the incoming recording phase will be pushed to an earlier point in time as there are no remaining tasks. Using a value of C greater than 1 enables the solution to increase T_P, according to later changes in the site, especially if there is a sudden increase in the number of subjects. This mechanism is balanced in preserving the unused pre-recording time and increasing the required estimated pre-recording time according to any increase in the number of subjects.

3.7 Proposed EBP Algorithm Utilizing the Dynamic Mechanism

Figure 9 shows a simplified EBP algorithm utilizing the dynamic mechanism. It proceeds as follows. (1) Lines 1-7 initialize different data structures. (2) Line 8 checks whether the selected seed node in the current camera seed array is eligible to be the first projected frame in the plan. (3) Lines 10-14 add the first seed frame to the plan list and update the related values based on the selected seed frame. (4) Lines 15-25 add consecutive seed frames to the plan from the sorted seed array as long as the recording time permits. (5) Lines 16-19 implement the clockwise and counter-clockwise examinations of the seed frames. (6) Lines 20-23 add the currently considered seed frame to the plan if there is still time left in the recording period. (7) Lines 27-31 check whether the currently selected starting seed frame results in the highest recognition probability. (8) Lines 33-35 ensure that the maximum required time for pre-recording is captured, which covers the operational time windows of all different cameras. (9) Line 36 adjusts the starting time of the incoming recording period ($nextTrackTime$) by using the maximum value of T_{SL} among various cameras. (10) Line 37 adjusts the value of T_p using Equation (9).

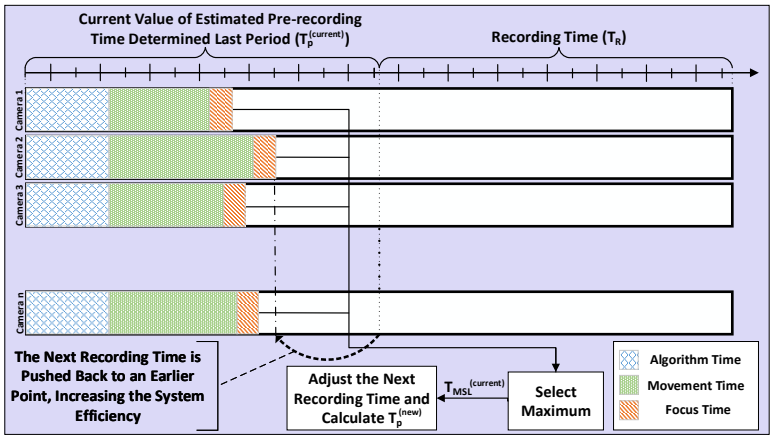


Fig. 8. Illustration of the Dynamic Mechanism for Determining Pre-recording Time

4 PERFORMANCE EVALUATION METHODOLOGY

We analyze the proposed solution in terms of the *subject recognition probability* and *computational time* by developing a detailed simulator in C++. Table 2 shows the default values of the main parameters. To reflect the most common deployment scenarios (e.g., train stations, sidewalks, and hotel lobbies), we assume a rectangular monitored site with the PTZ cameras being placed around the edges, as shown in Figure 10. If the system has fewer than 16 cameras, the larger camera numbers are dropped. The field of view is determined by the field of view's horizontal and vertical angles, which are 50 and 30 degrees, respectively. The cameras could overlap as this is the case in real deployment scenarios.

In contrast with our prior work, we use a realistic inclusive speed model [22] to update the speeds of subjects, considering many important factors, including the number of pedestrians that can be accommodated in the minimal length of the facility, pedestrian density, maximum/jam density, and free-flow speed of pedestrians. Subjects arrive randomly from any of the four directions and enter the site with a random direction and a random speed. Poisson distribution is used to model subject arrivals, while a truncated Gaussian distribution is used for the initial speed, and a truncated uniform distribution is used for the directions of movement. The PTZ cameras are located at the perimeter of the site with different (x, y, z) locations. Furthermore, a detailed realistic characterization for each camera is implemented, including the refocusing speed, the sensor dimensions, the speeds of pan, tilt, and zoom, their maximum and minimum limits, and their step-size. To ensure stability, the results are reported only after 6000 different subjects appear within the site. In Figure 11, we changed the stopping criterion to 1000 subjects instead of 6000 subjects to illustrate the impact of repeated runs and show that the stopping criterion of 6000 subjects is strict. The simulations were executed on a workstation with a 64-bit 3.6-GHz Quad-core CPU and 16 GB RAM.

5 RESULT PRESENTATION AND ANALYSIS

5.1 Analysis of the Camera Scheduling Alternatives and Effectiveness of the Clustering Approach

Let us now analyze the effectiveness of the two proposed scheduling schemes and the clustering approach in terms of recognition probability and algorithm time. We refer to GBG and EBP

```
00. //Input: planSeedSet, a Set of Seed frames for each camera
00. //Output: ResultedPlan, Calculated plan, generated for all PTZ camera
00. BuildPlans(){
01.  for (each Camera cam){ // for every PTZ camera
02.   resultedPlan[cam] = {}; maximumRecognituinProbability = 0;
03.   // sort seed frames in every camera list based on the zoom value
04.   sort(planSeedSet[cam][ ], zoom);
05.   for (each seed f){ //for every frame
06.    tempPlan = {}; planRecognitionProbability = 0; // initialize variables
07.    currentFrame = f; // set the starting projected frame node for current camera plan
08.    if (T_move,cam→currentFrame > T_PR) then
09.     continue; // The required time to move to projected frame currentFrame violates T_PR
10.    else {
11.     tempPlan.push(planSeedSet[cam][currentFrame]);
12.     planRecognitionProbability+ = P_tot(currentFrame);
13.     forwardScanning = 1;
14.     T_RR = T_R − T_e;
15.     while (T_RR > 0){
16.      if (currentFrame == firstSeedFrame) then forwardScanning = true;
17.      if (currentFrame == lastSeedFrame) then forwardScanning = false;
18.      if (forwardScanning == 1) then currentFrame + +;
19.      else currentFrame − −;
20.      if ((T_move,cam→currentFrame + T_e) < T_RR) then{
21.       tempPlan.push(planSeedSet[cam][currentFrame]);
22.       planRecognitionProbability+ = ψ(currentFrame);
23.       T_RR = T_RR − T_e;
24.      } End of if
25.     } End of while (T_RR > 0)
26.    } End of else T_move is valid
27.    if (planRecognitionProbability > maximumRecognitionProbability){
28.     maximumRecognitionProbability = planRecognitionProbability;
29.     resultedPlan[cam] = tempPlan;
30.     T_SL = CurrentSystemLeadTimeForCamera(cam)
31.    } End of if
32.   } End of frame loop j
33.   if (T_MSL < T_SL)
34.    T_MSL = T_SL;
35.  } End of camera loop cam
36.  nextTrackTime = nextTrackTime − T_P + T_MSL;
37.  AdjustT_P(T_P^(current), T_MSL)
38. } // End of BuildPlans()
```

Fig. 9. Simplified Enhanced EBP Algorithm for Building the Scheduling Plan for Each Camera Utilizing the Dynamic Mechanism

scheduling schemes with the clustering approach enabled as *GBG-C* and *EBP-C*, respectively. Figure 11 shows the results as the number of PTZ cameras is varied. The subject arrival rate is 16 subjects/second and each test case has 1000 subjects passing through the monitored area. The standard deviations for the recognition probability and computational time are shown using 10 different simulation runs for each data point. As the standard deviation is fairly small, we included such results in only some figures. EBP-C achieves the best overall performance, followed by EBP. GBG, GBG-C, and EBP have higher computational complexity than EBP-C, and thus after increasing the number of cameras beyond a certain value, they are unable to finish the required algorithmic

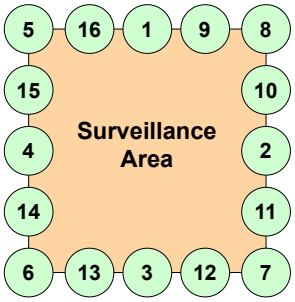


Fig. 10. The Camera Placement and Order in the Monitored Area

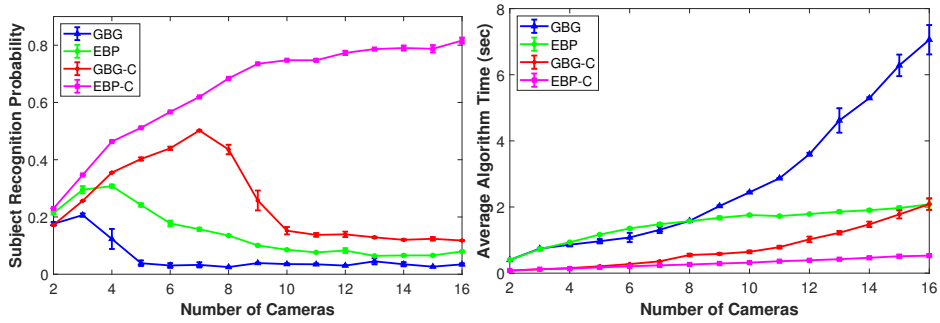
Parameter	Default Value(s)
Request Arrival Distribution	Poisson
Initial Request Speed Distribution (free flow speed)	Truncated Normal (mean=1.5 Req./sec, std=0.5)
Site Area, Focus Latency	$80m \times 60m$, 0.5 sec
Mean Request Arrival Rate (μ)	2-38, Default = 18 Req./sec
Entrance Angle	-40° - 40° with $\perp$ to entrance side
Number of Cameras	8
PTZ Tilt, Zoom, Pan	0°-90°, 1-50, 0°-180°
Pre-recording Time	2-15 sec, Default = 3
Recording Time	2-6 sec, Default = 3
Independence Threshold	0.5 second
Density of pedestrians in a jammed state	2 pedestrians/m^2
Lateral spacing required for a pedestrian to move (L_s)	1.55m
Pedestrians that can be fit in the minimal length of the facility	(minimal length - 1.07) / L_s
Constant for computing $T_P^{(new)}$ (C)	1.2
b_1 Constant in Eq. (2)	48.32
a and b_2 Constants in Eq. (3)	0.6927, -0.01544

Table 2. Default Values of Simulation Parameters

computations before the beginning of the next recording period. EBP performs better than GBG in the two metrics. This performance enhancement in EBP is attributed to allowing each camera to view a different set of subjects in a recording phase as opposed to GBG, which constantly covers the same frame during the recording phase. The reduced computational time, however, is due to significantly reducing the search space by selecting only the best combination of seed frames for each camera.

The average subject recognition probability generally improves with the number of PTZ cameras, but the computational workload increases. With GBG, the recognition probability and the subject coverage decrease after the number of cameras becomes larger than three. Meanwhile, the computational workload starts to increase more sharply, enforcing the system to finish the pre-recording period before fully executing the control solution, thereby resulting in poor performance during the recording phase. The higher computational workload is due to the increase in the search domain, as the additional FoRs have to be examined to get the top projected frame-sets. The maximum number of cameras that can be selected for GBG depends on many factors and thus the number of PTZ cameras should be selected based on the performance-time tradeoff.

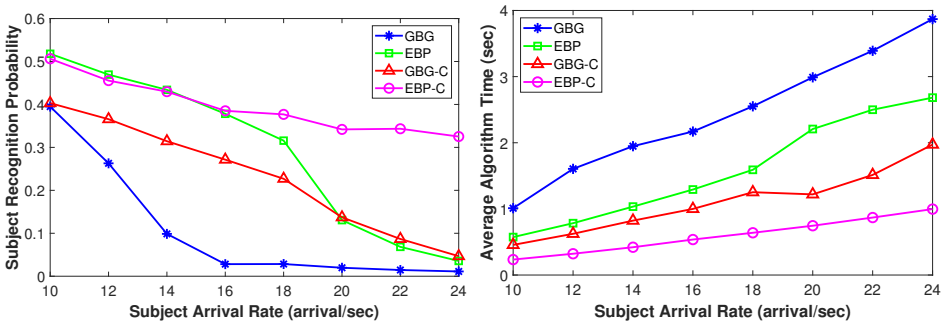
Figure 12 shows the comparative results as the subject arrival rate changes. As expected, the two metrics become worse as we increase the arrival rate as there is a higher chance for a subject to depart the site without being captured by any camera, and the average time spent on capturing a subject decreases. EBP continues to be better than GBG, with the performance gaps in the two metrics widening in comparison as the arrival rate increases (but up to a certain point in the first metric). *To be able to handle higher arrival rates, the system should be sized appropriately in terms of both the number of PTZ cameras and computational power.* GBG-C and EBP-C perform significantly better than their non-clustering counterparts as the arrival rate increases. With clustering, the system scalability improves because of the reduction in computational complexity due to focusing only on the subject-dense areas of the monitored site. EBP-C behaves better than GBG–C when the arrival rate is high. Note that the average number of subjects that are simultaneously present



(a) Average Subject Recognition Probability (b) Average Time

Fig. 11. Comparing the Effectiveness of Clustering by Varying the Number of PTZ Cameras

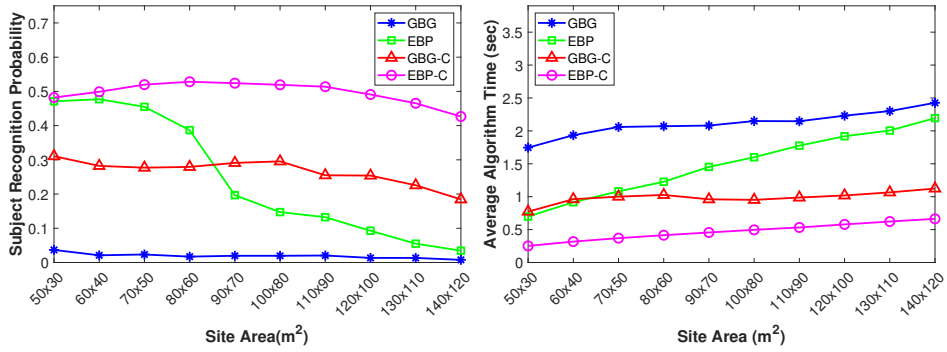
in the area increases with the subject arrival rate, assuming that the area is fixed. Consequently, the probability of a subject being occluded by other subjects increases, thereby worsening the recognition probability. This trend persists until the arrival rate is high enough to the extent that most of the subjects lose any chance of being covered by any camera, and thus the recognition probability becomes practically zero. As expected, higher subject arrival rates lead to a more crowded site, thereby complicating the pre-recording calculations of the algorithm.



(a) Average Subject Recognition Probability (b) Average Time

Fig. 12. Comparing the Effectiveness of Clustering by Varying the Mean Arrival Rate

Figure 13 analyzes the impact of clustering as the site area is varied. As the site area increases, the PTZ cameras cannot cover the entire area as effectively. Interestingly, there is a small initial improvement in subject recognition probability with EBP-C as the site area increases. This initial increase is attributed to the subject concentrations in the site. With EBP-C, the first three site dimensions are not spacious enough and the subjects are too close to each other. Therefore, the clustering process in these small dimensions is not as effective because the objects inside a cluster tend to occlude each other more significantly, thereby reducing the chance of subject recognition by the PTZ cameras. By increasing the site area, the average algorithm time increases as it takes a longer time for a subject to leave the site, and subsequently, it will contribute more to the computational workload of the algorithm.



(a) Average Subject Recognition Probability (b) Average Time

Fig. 13. Comparing the Effectiveness of Clustering by Varying the Site Area

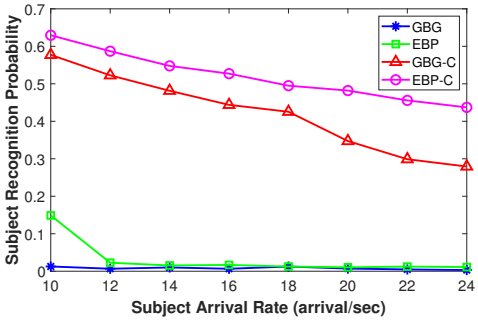


Fig. 14. Effectiveness of the Clustering Approach by Considering a Terminal-like Arrival Pattern

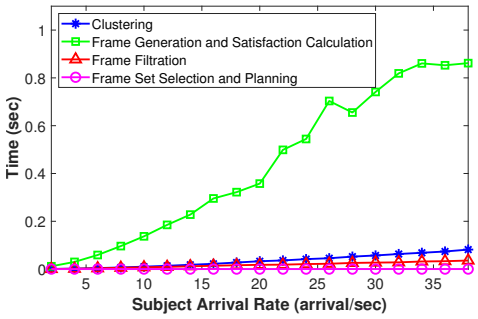


Fig. 15. Computation Times for Different Units of the Control Solution [EBP-C]

Figure 14 shows the impact of clustering as the subject arrival rate is varied, considering a special arrival pattern for subjects mimicking a scenario similar to that in places such as airport terminals, where the entrance is restricted and the walking angle of people entering the area is less varied and mostly towards the check-in gates ahead. We assume here that the entrance area is only 10% of that of the monitored site and the angle variation is limited to ±20° from the perpendicular line at the entrance. As expected, the clustering approach is even more effective in cases where subjects occupy only a small portion of the monitored area and tend to walk in the same direction as other adjacent subjects. The clustering methods significantly outperform the non-clustering counterparts as clustering substantially decreases the computational burden in these scenarios. Additionally, it can be observed that EBP-C outperforms GBG-C as it has the advantage of handling dense environments through better computational efficiency.

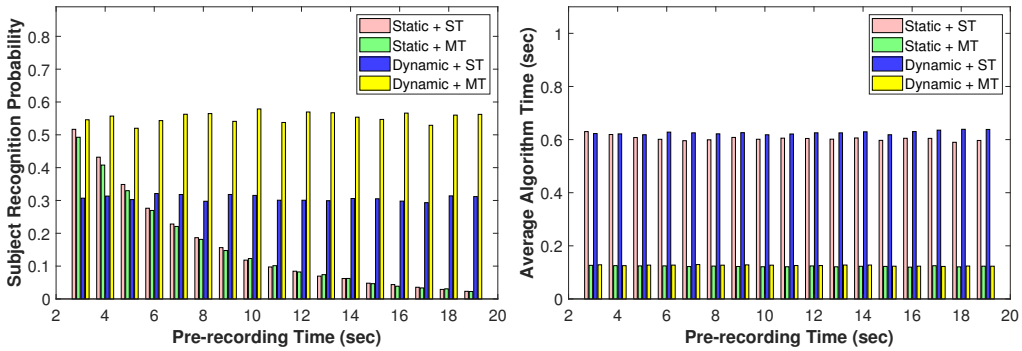
5.2 Timing Analysis and Effectiveness of the Parallel Implementation and Dynamic Mechanism

For the proposed solution incorporating the best scheduling scheme (EBP-C), let us now profile and compare the computational times of the various operational units: (*i*) clustering, including subject attribute extraction and prediction, (*ii*) projected frame generation and satisfaction calculation, (*iii*) projected frame filtration, and (*iv*) projected frame-set selection and camera planning (i.e. camera

scheduling). We studied the behavior multiple times with different system characteristics, but only the main results are shown. Figure 15 shows the computation time for various system units as the subject arrival rate increases. The projected frame generation and satisfaction calculation unit presents itself as the best candidate for parallel implementation because it dominates the overall algorithm time. Additionally, the difference in computation time between this unit and other units increases greatly with the arrival rate. Figure 2 shows that this unit is developed in a parallel manner where there is a new thread of execution for each camera in the system. These threads could be spawned on multiple CPUs in the centralized parallel processing subsystem. This makes the system scalable on the order of cameras.

Let us analyze the effectiveness of the parallel implementation of the projected frame generation and satisfaction calculation unit as well as the dynamic mechanism for determining the pre-recording time in terms of recognition probability and algorithm time. In particular, we compare four alternative implementations in the achieved overall subject recognition probability and computational time: (*i*) static pre-recording time with single-threading (ST), (*ii*) dynamic pre-recording time with ST, (*iii*) static pre-recording time with multi-threading (MT), and (*iv*) dynamic pre-recording time with MT.

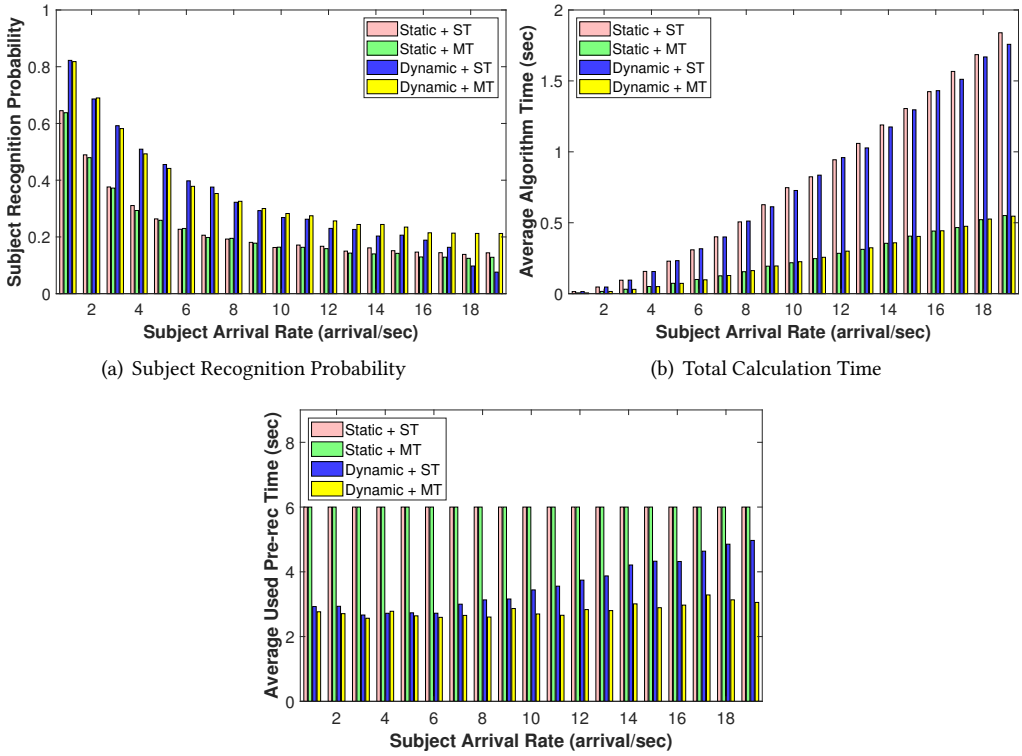
Figure 16 shows the comparative results as the pre-recording time is varied. The combination of the dynamic mechanism with MT consistently outperforms other alternatives in recognition probability. The performance of the dynamic mechanism with ST is steady, but it is worse than that with MT because of the increased computational time, thereby lowering the chance for early completion of the algorithm in the pre-recording phase. The average overall computational time for the MT implementation is significantly shorter than ST. Static alternatives have shorter computational times because of the reduction in the number of subjects who pass through the site and become sufficiently recognized and involved in various processes and calculations. Note that the gap in the subject recognition probability between the dynamic and static alternatives is wider for longer pre-recording times.



(a) Subject Recognition Probability (b) Total Calculation Time

Fig. 16. Comparing the Effectiveness of the Dynamic Mechanism and Parallel Algorithm (MT) when the pre-recording Time is Varied [EBP-C, Subject Arrival Rate = 10 requests/sec]

Figure 17 shows the comparative results of the four alternative implementations as the subject arrival rate is varied. The dynamic mechanism generally outperforms the static. The dynamic alternatives perform nearly the same until the arrival rate reaches 16 arrivals/sec. After this point, the dynamic mechanism with MT significantly outperforms the ST alternative. The reason can be



(a) Subject Recognition Probability

(b) Total Calculation Time

(c) Average Actually-Used Pre-Recording Time

Fig. 17. Comparing the Effectiveness of the Dynamic Mechanism and Parallel Algorithm (MT) when the Mean Subject Arrival Rate is Varied [EBP-C, Pre-recording: 6s, Recording: 6s]

attributed to the fact that the computational workload before this value is not challenging enough to favor the MT implementation. There is almost no difference in performance between the static alternatives as both do not use any available spare time in the pre-recording time, even if one finishes all necessary computations sooner than the other.

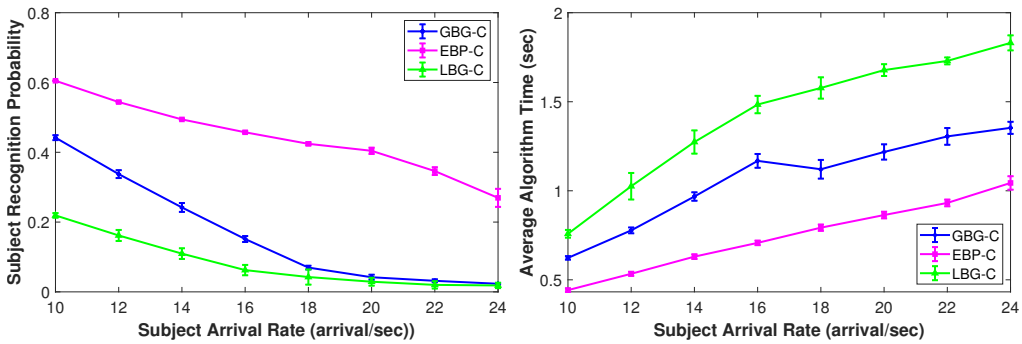
Figure 17(c) shows the average pre-recording time that is actually used by the four alternative implementations. The static methods use the entire pre-recording phase regardless of the actual computational time for generating the plan. The dynamic mechanism coupled with multithreading (Dynamic + MT) consumes the least amount of time in the pre-recording phase and consequently spends the most time in the recording phase, resulting in the highest recognition probability. This behavior is due to better scalability where the increased workload is handled more efficiently by parallel processing. The dynamic mechanism with ST cannot take advantage of the scalability present in MT. Consequently, the amount of time consumed in the pre-recording phase increases with the subject arrival rate. The figure demonstrates the special characteristic of the dynamic mechanism in utilizing the feedback loop for determining the pre-recording time (Equation (9)).

In summary, the dynamic mechanism when coupled with multithreading provides the best overall results in recognition probability and algorithm running time. The multithreaded method provides the benefit of a faster calculation time in the pre-recording phase. The dynamic mechanism can

take advantage of this early calculation completion and then by rescheduling the recording phase to an earlier point in time, it prevents wastage of time in the pre-recording phase.

5.3 Comparison with Existing Work

As discussed in Section 2, the work of Xu et al. [25] is the most related to this study because their LBG method deals with scheduling subjects to different cameras and thus provides an alternative to the proposed GBG and EBP methods. Figure 18 compares the effectiveness of EBP-C and GBG-C with LBG. To ensure fair evaluation, our clustering approach is applied to LBG as well, and thus we refer to it as LBG-C. The figure shows that the proposed GBC-C and EBP-C methods perform significantly better than LBG-C in both the recognition probability and computational time. EBP-C performs the best as it tours over different frames during the recording period, as opposed to continually recording only one frame in the case of GBG-C and LBG-C. GBG-C performs better than LBG-C, especially for lower subject arrival rates. This behavior can be contributed to considering different FoRs for different cameras, utilizing a 3D coordination system to determine the location of subjects, and considering the overlap among projected frames. As the subject arrival increases, the computational workload of both GBG-C and LBG-C increases, and the time spent in the pre-recording period is inadequate for these methods to produce an effective and complete recording plan for the recording phase, thereby greatly impacting their performance. In contrast, EBP-C uses a more efficient and less computationally complex method to generate the recording plans, and consequently, it is more likely to properly finish building a recording plan during the limited pre-recording phase.



(a) Average Subject Recognition Probability

(b) Average Time

Fig. 18. Comparing GBG-C and EBP-C with LBG-C

6 CONCLUSIONS

This paper has presented a system that autonomously controls the PTZ settings of cameras in a manner that seeks to optimize the overall threat detection or recognition accuracy. As an integral part of the overall solution, the paper presents two alternative schemes for camera scheduling: *Grid-Based Grouping* (GBG), and *Elevator-Based Planning* (EBP). Besides, it utilizes clustering to group subjects, thereby enabling the system to focus on the areas that are more densely populated with subjects. Moreover, it proposes a dynamic mechanism for controlling the pre-recording time spent on running the solution. Furthermore, the paper develops a parallel algorithm, allowing the most time-consuming phases to be parallelized.

The main results can be summarized as follows. (1) The system should be sized appropriately in terms of the number of PTZ cameras and computational capacity based on site characteristics, including the area and subject arrival rate. (2) The proposed EBP scheduling scheme achieves the best overall performance in terms of recognition probability and computational time and offers the best scalability when increasing the subject arrival rate or the site area. Specifically, it improves the measured subject recognition probability value by 85% on average in comparison to that of GBG while reducing the computational time by 15%. (3) Clustering further improves the recognition probability by a factor of 2 on average and reduces the computational time by 60%. (4) The recognition probability increases with the number of PTZ cameras and decreases with the subject arrival rate. The chance of covering a subject with a high-resolution frame decreases as the site area is widened, thereby reducing the recognition probability. (5) Some parameters have conflicting impacts on different metrics. For example, while the subject recognition probability improves with the number of PTZ cameras, the plan-building time becomes longer. (6) The dynamic mechanism provides an additional improvement of more than 80% on average in the recognition probability by efficiently using the pre-recording time, while increasing the computational time by only 15%. (7) Using 8 threads on a single 4-core Intel Core i7 workstation (with 8 virtual cores), parallelizing the most-time consuming phases reduces the time of the overall solution by a factor of three or more on average while improving the subject recognition probability by 20%.

REFERENCES

[1] Michele Benetti, Massimo Gottardi, Tobias Mayr, and Roberto Passerone. 2018. A Low-Power Vision System With Adaptive Background Subtraction and Image Segmentation for Unusual Event Detection. *IEEE Transactions on Circuits and Systems I: Regular Papers* 65, 11 (2018), 3842–3853.

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* 2004.10934 (2020). arXiv:2004.10934 [cs.CV] https://arxiv.org/abs/2004.10934v1

[3] Bastiaan J. Boom, G.M. Beumer, Luuk Spreeuwers, and Raymond N. J. Veldhuis. 2006. The Effect of Image Resolution on the Performance of a Face Recognition System. In *Proceedings of International Conference on Control, Automation, Robotics and Vision.* 1–6.

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[5] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. 2009. Pedestrian Detection: A Benchmark. In *Proceedings of Computer Vision and Pattern Recognition conference (CVPR).*

[6] Lukas Esterle. 2017. Centralised, Decentralised, and Self-Organised Coverage Maximisation in Smart Camera Networks. In *2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO).* 1–10.

[7] David A. Forsyth and Jean Ponce. 2011. *Computer Vision: A Modern Approach (Second Edition).* Pearson.

[8] Ralph Gross, Simon Baker, Iain Matthews, and Takeo Kanade. 2004. Face Recognition across Pose and Illumination. *Handbook of Face Recognition* (2004).

[9] Hayder Hamandi and Nabil J. Sarhan. 2020. QRMODA and BRMODA: Novel Analytical Models of Face Recognition Accuracy in Terms of Video Capturing and Encoding Parameters. *In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)* (July 2020), 1–6.

[10] Jaskaranjit Kaur and Harpreet Singh. 2015. Performance evaluation of a novel hybrid clustering algorithm using birch and K-means. In *Proceeding of Annual IEEE India Conference (INDICON).* 1–6.

[11] Dongchil Kim, Kyoungman Kim, and Sungjoo Park. 2019. Automatic PTZ Camera Control Based on Deep-Q Network in Video Surveillance System. In *Proceeding of International Conference on Electronics, Information, and Communication (ICEIC).* 1–3.

[12] Nils Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, and Peter Tu. 2008. 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.*

[13] Peter R. Lewis, Lukas Esterle, Arjun Chandra, Bernhard Rinner, Jim Torresen, and Xin Yao. 2015. Static, Dynamic, and Adaptive Heterogeneity in Distributed Smart Camera Networks. *ACM Transactions on Autonomous and Adaptive Systems* 10, 2, Article 8 (June 2015), 30 pages.

[14] Chih-Wei Lin, Kuan-Wen Chen, Shen-Chi Chen, Cheng-Wu Chen, and Yi-Ping Hung. 2015. Large-Area, Multilayered, and High-Resolution Visual Monitoring Using a Dual-Camera System. *In Proceeding of ACM Transactions on Multimedia*

Computing, Communications, and Applications (TOMM) 11 (01 2015), 1–23.

[15] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. 2017. SphereFace: Deep hypersphere embedding for face recognition. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. 1.

[16] Oded Z. Maimon and Lior Rokach. 2005. *Data Mining and Knowledge Discovery Handbook.* Springer.

[17] Goutam Mali and Sudip Misra. 2017. Topology Management-Based Distributed Camera Actuation in Wireless Multimedia Sensor Networks. *ACM Transactions on Autonomous and Adaptive Systems* 12, 1, Article 2 (April 2017), 33 pages.

[18] Chindukuri Mallikarjuna and P Chitti Babu. 2016. Performance Analysis of Disk Scheduling Algorithms. *International Journal of Computer Sciences and Engineering* 4, 5 (2016), 180–184.

[19] Joao C. Neves and Hugo Proena. 2015. Dynamic camera scheduling for visual surveillance in crowded scenes using Markov random fields. In *Proceeding of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).* 1–6.

[20] Claudio Piciarelli, Lukas Esterle, Asif Khan, Bernhard Rinner, and G.L. Foresti. 2015. Dynamic Reconfiguration in Camera Networks: A Short Survey. *IEEE Transactions on Circuits and Systems for Video Technology* 26 (01 2015), 1–1. https://doi.org/10.1109/TCSVT.2015.2426575

[21] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti. 2010. Occlusion-aware multiple camera reconfiguration. In *Proceedings of IEEE International Conference on Distributed Smart Cameras (ICDSC).* 88–94.

[22] Khalidur Rahman, Noraida Abdul Ghani, Anton Abdulbasah Kamil, Adli Mustafa, and Md Ahmed Kabir Chowdhury. 2013. Modeling pedestrian travel time and the design of facilities: A queuing approach. *PloS one* 8, 5 (2013), e63503.

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 815–823.

[24] Sinnu S. Thomas, Sumana Gupta, and Venkatesh K. Subramanian. 2017. Smart surveillance based on video summarization. In *Proceeding of IEEE Region 10 Symposium (TENSYMP).* 1–5.

[25] Yiliang Xu and Dezhen Song. 2010. Systems and algorithms for autonomous and scalable crowd surveillance using robotic PTZ cameras assisted by a wide-angle camera. *Autonomous Robots* 29, 1 (2010), 53–66.

[26] Yi Yao, Besma R. Abidi, Nathan D. Kalka, Natalia A. Schmid, and Mongi A. Abidi. 2008. Improving long range and high magnification face recognition: Database acquisition, evaluation, and enhancement. *Computer Vision and Image Understanding* 111 (August 2008), 111–125. Issue 2.

[27] Jinhua Zeng, Jinfeng Zeng, and Xiulian Qiu. 2017. Deep learning based forensic face verification in videos. In *Proceeding of International Conference on Progress in Informatics and Computing (PIC).* 77–80.

[28] Xuhui Zhou, Robert T. Collins, Takeo Kanade, and Peter Metes. 2003. A master-slave system to acquire biometric imagery of humans at distance. In *Proceedings of ACM SIGMM International Workshop on Video surveillance (IWVS)* (Berkeley, California). ACM, New York, NY, USA, 113–120.

[29] Zihan Zhou, Andrew Wagner, Hossein Mobahi, John Wright, and Yi Ma. 2009. Face recognition with contiguous occlusion using Markov random fields. In *Proceedings of IEEE International Conference on Computer Vision.* 1050 –1057.