Multimed Tools Appl
DOI 10.1007/s11042-013-1816-y

Cell-based visual surveillance with active cameras for 3D human gaze computation

Zhaozheng Hu · Takashi Matsuyama · Shohei Nobuhara

Abstract Capturing fine resolution and well-calibrated video images with good object visual coverage in a wide space is a tough task for visual surveillance. Although the use of active cameras is an emerging method, it suffers from the problems of online camera calibration difficulty, mechanical delay handling, image blurring from motions, and algorithm un-friendly due to dynamic backgrounds, etc. This paper proposes a cell-based visual surveillance system by using N ($N \geq 2$) active cameras. We propose the camera scan speed map (CSSM) to deal with the practical mechanical delay problem for active camera system design. We formulate the three mutually-coupled problems of camera layout, surveillance space partition with cell sequence, and camera parameter control, into an optimization problem by maximizing the object resolution while meeting various constraints such as system mechanical delay, full visual coverage, minimum object resolution, etc. The optimization problem is solved by using a full searching approach. The cell-based calibration method is proposed to compute both the intrinsic and exterior parameters of active cameras for different cells. With the proposed system, the foreground object is detected based on motion and appearance features and tracked by dynamically switching the two groups of cameras across different cells. The proposed algorithms and system have been validated by an in-door surveillance experiment, where the surveillance space was partitioned into four cells. We used two active cameras with one camera in one group. The active cameras were configured with the optimized pan, tilt, and zooming parameters for different cells. Each camera was calibrated with the cell-based calibration method for each configured pan, tilt, and zooming parameters. The algorithms and system were applied to monitor freely moving peoples within the space. The system can capture good resolution, well-calibrated, and good visual coverage video images with static background in support of automatic object detection and tracking. The proposed system performed better than traditional single or multiple fixed camera system in term of image resolution, surveillance space, etc. We further demonstrated that advanced 3D features, such as 3D gazes, were successfully computed from the captured good-quality images for intelligent surveillance.

Z. Hu (✉)
ITS Research Center, Wuhan University of Technology, Wuhan 430063, China
e-mail: zhaozheng.hu@gmail.com

Z. Hu · T. Matsuyama · S. Nobuhara
Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-Ku, Kyoto 606-8501, Japan

Published online: 19 December 2013

Keywords Cell-based surveillance · Active cameras · Camera layout · Cell partition · Cell-based calibration

1 Introduction

Visual surveillance systems are widely applied in almost all aspects related to security and safety issues [7]. For the past decades, a lot of visual surveillance systems have been developed. For example, in the earlier years, surveillance system simply consists of a single fixed camera, which usually suffers the problems of limited surveillance space and low image resolutions. To address these issues, some advanced surveillance systems were develop, which usually consist of multiple fixed cameras or camera network [3, 7, 9, 11, 16]. The multi-camera system can enlarge the surveillance space and increase the image resolutions [3]. Moreover, the multiple cameras can be well calibrated in advanced so that both the camera intrinsic parameters and the relative positions between each can be accurately determined. As a result, the systems can successfully extract advanced 3D features, such as 3D shapes, 3D positions, trajectory, poses, etc., which have important applications for intelligent surveillance, especially for advanced human behavior studies [7, 9, 19]. However, the cost for multi-camera system is usually high, e.g., we need a lot of cameras and computers, or even computer cluster, which hinders it from broader applications. To tackle the high resolution and wide surveillance space conflicts, the active camera based surveillance systems have been developed [1, 12]. The systems can adaptively change the camera parameters such as pan, tilt, and zooming, etc., according to the object positions in the scene. However, the difficulties for applying active cameras in practice are summarized as follow [9, 12, 17, 18]:

1) Camera calibration difficulty; It is very difficult to calibrate active cameras real time, as the online calibration is not accurate and robust in practice. Robust and accurate online camera calibration is still an open problem in the field of computer vision. As a result, it is difficult to compute accurate 3D features from the captured images;
2) Mechanical delay issue; Active cameras in practice have mechanical delays while performing the pan, tilt, or zooming motions. Such mechanical delay should be well compensated in order to track the moving object;
3) Low image quality from motion; Images captured by the cameras in motions usually have bad quality. For example, camera pan, tilt, and zooming motions usually cause image blurring;
4) Algorithm un-friendly; many intelligent algorithms, such as moving object detection, don't favor active cameras. For example, it is much more difficult to model a dynamic background with active camera than static background with fixed camera. Hence, active cameras are not algorithm friendly.

In this paper, we propose a novel cell-based visual surveillance system that can tackle the above issues of using active cameras. Particularly, we deal with the problem of assigning N active cameras to optimally and adaptively observe a moving object in a path-like surveillance space. In this proposal, the N active cameras are evenly divided into two groups with each group observing the cells of even or old number alternatively. The surveillance space is partitioned into cell sequences (or cell chain). We formulate the optimal camera layout, cell partition, and camera control into a unified optimization problem. That is, we try to maximize the mean object image resolution while meeting the various constraints to solve the camera layout, cell partition, and camera control. The optimization problem is solved by a full search

method. With the cell-based setup, each camera is optimally configured with the PTZ parameters for each cell. And the cell-based calibration method is applied to accurately determine both the intrinsic and exterior parameters. The control and switching scheme of active cameras is designed to detect and track moving object to capture high quality video images, which can be used to compute advanced 3D features.

The main contributions of this paper are summarized as follow: 1) propose the conception of camera scan speed map (CSSM) that is particularly useful to deal with the mechanical delay (PTZ delay) problem, and hence effective for active camera system design; 2) formulate the three mutually coupled problems of camera layout, surveillance space partition with non-uniform cells, and camera control, simultaneously into a unified optimization problem by maximizing the object resolution while meeting various constraints; 3) a cell-based calibration method is applied to compute both the intrinsic and extrinsic parameters of each active camera for different cells; 4) propose the camera control and switching scheme to detect and track moving object for high quality video image capture.

2 Related work

From the development of visual surveillance systems, two obvious trends can be evidently observed. On the one hand, visual surveillance systems are becoming more and more intelligent [7, 16]. For example, in the earlier time, the surveillance system was developed for the main purpose of video data recording. People need to review the raw data to investigate the contents. Later on, many intelligent algorithms, such as object detection, recognition, tracking, re-identification, and even high-level surveillance analysis, such as event detection, abnormality detection, etc., have been successfully developed [7, 8, 10, 16, 21]. Recently, 3D detail features, such as 3D depth or position, 3D pose, 3D shape, geometric size, 3D trajectory, etc, are reported to play more and more important roles for intelligent visual surveillance in the literatures [7, 10, 16, 19]. For example, 3D human gazes are crucial to understand people behavior and interests, as show in Fig. 1, where the camera computes human gazes for people visual attention computing. And such system in Fig. 1 can find many applications for advertisements, exhibition, and security sites, etc [19]. Note that, in order to extract accurate and reliable 3D features, the camera (s) should be accurately calibrated [17]. Although many offline calibration methods have been successfully proposed in the literatures, robust and accurate online camera calibration still remains an issue in the field of computer vision [6].

Fig. 1 A visual surveillance system in an exhibition room for people visual attention computing, where 3D gazes are crucial to study how people are interested in the contents displayed on the screen

And for the proposed visual surveillance system with active cameras, the online calibration problem should be well addressed.

One the other hand, visual surveillance systems are becoming more and more active, adaptive, and object oriented so that high quality image video data is captured in support of intelligent algorithms [1, 7, 9, 12, 16–18]. Basically, a practical surveillance system may be more interested in the moving foreground objects rather than the background. Hence, the system should be active and adaptive to capture high quality data of the object. Actually, these two aspects are greatly related to each other. On the one hand, high quality data can enhance intelligent algorithms, such as detection, recognition, and tracking. On the other hand, the system relies on the computation results from the intelligent algorithms to actively update the camera. As a result, we define four basic criterions for good quality images in the surveillance context as follow

1) Sufficient image resolution for the object. Image resolution is not only crucial for image visual quality but also important for many computations, such as 3D feature computation, accurate trajectory extraction, object detection and recognition, etc. We usually require good image resolution for accurate vision computation in support of intelligent surveillance.
2) Good visual coverage of the object. All parts of the object should be well within the view field of the camera so that the object is not partially cutter.
3) Wide surveillance space. Wide surveillance space is crucial to study the behaviors and patterns of the objects of interest. For example, we usually require the object trajectory with sufficient length to analyze the behaviors.
4) Accurate camera calibration. Camera calibration is an essential step for 3D feature computation. Accurate 3D features greatly reply on precise camera calibration. Moreover, we need to know both the intrinsic and exterior parameters of the camera for each image in case of using active cameras.

To address the above issues, the cell-based conception is proposed [17, 18], which was first applied for 3D video capture. In their proposals, the planar space is first partitioned into a number of cells. And the cameras are grouped into teams and assigned to observe different cell units to capture good-quality video images. In such cell-based system, the three mutually-coupled problems should be well addressed: 1) camera layout; 2) cell partition; 3) optimal camera control. However, in their work, they assume fixed camera layout. Furthermore, they assume identical mechanical delays for different cell units, which is however not practical for a real surveillance system. Therefore, their proposal algorithms are not applicable for a practical surveillance system. Besides, some researchers worked with the layout of multiple cameras to have good visual coverage of an irregular surveillance space [4, 5]. However, the camera layout in this paper is determined not only by the camera parameter control but also by the partition of the cells, which makes the problem in this paper different, or even more complicated, compared to existing ones.

3 The proposed algorithm

First of all, we assume that a single object moves freely on a planar surface that is rectangle-shape or approximated by a rectangle (see Fig. 2). It is called a path-like surveillance space in this paper. Note that we constraint the shape (path-like shape) of the surveillance space but not the size. Actually, the path-like surveillance spaces are commonly available in daily life, such

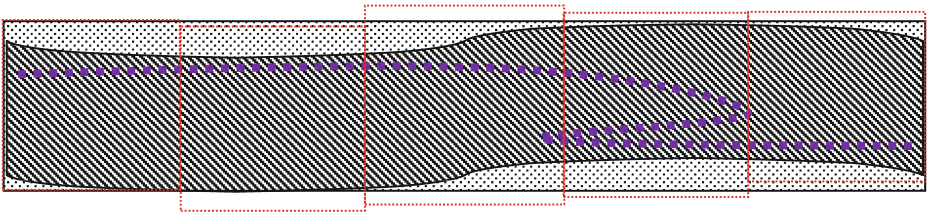

Fig. 2 Approximate a surveillance space with a rectangle and partition it with a number of square-shape cells of different sizes and positions. The object can move freely within the space (see the dash-line trajectory)

as parking lot, roadway traffic, and corridor, etc. The cell-based surveillance conception is briefly introduced as follow. The space is first partitioned into a sequence of cell units (or cell chain), as shown in Fig. 2, so that each cell has at most two neighbors (the head and the tail cells have only one neighbor). Each active camera is then configured with optimized pan, tilt, and zooming parameters for each associated cell (The definition of "optimized" is discussed in Section 3.2). The active cameras are divided into groups to observe the moving object by switching into different cells. First of all, we need to decide how many groups to divide the groups and how many active cameras are assigned into one group. As we need at least one group of cameras to observe the object at any time in any position, we should have at least two camera groups so that one camera group can watch the object while the other is waiting or preparing for waiting for the object. Note that more camera groups can decrease the usage efficiency of the cameras, as the usual scenery is that one group observes the object while the others are waiting and don't capture images of the object. In this paper, we divide the active cameras into two groups. These two camera groups are then controlled and switched to detect and track the moving object within the space to capture high quality images for visual surveillance. The number of cameras in one group is dependent on the application task. For example, in 3D video capture, we usually need six or more camera in one group to observe the object in all direction. And for a surveillance system, one camera in one group can capture object trajectory and 3D human gazes to meet the demand.

Based on the above discussions, we define two basic problems for a cell-based surveillance: 1) How to partition the surveillance space with cell chain, layout the cameras, and control the cameras? 2) How to utilize the designed cell-based surveillance method to detect and track moving object? Especially, for the first problem, we need to meet various constraints, such as mechanical delay, image resolution of the object, visual coverage, camera calibration, etc. And we propose the camera scan speed map (CSSM), which describes how fast the active camera scan across the surveillance space at each position. The CSSM is applied to address the mechanical delay problem.

3.1 The camera scan speed map (CSSM)

An active camera takes different delay time when performing the pan, tilt, or zooming motion (as shown in Fig. 3). As the pan, tilt, and zoom can be set simultaneously for a practical active PTZ camera, the maximum of them is used as the time delay between two points as follows

$$t(P, P') = \max\left(t_p, t_t, t_z\right) \tag{1}$$

where t_p, t_t, t_z are the time delays of pan, tilt, and zoom, between two arbitrary positions in the surveillance space, respectively. The camera scan speed between any two points is thus computed as

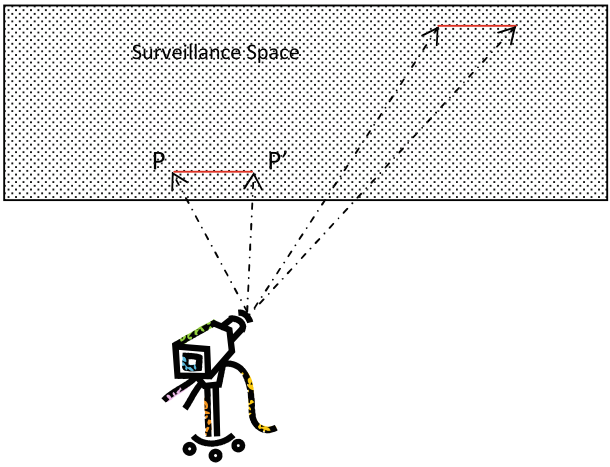

Fig. 3 Different scan speed at different spots due to the different distances and view angles

$$v_{cam}(P,P') = \left\|P-P'\right\|/t(P,P') \tag{2}$$

Let $P' \rightarrow P$, the scan speed at the point P is thus derived as

$$v_{cam}(P) = \lim_{p' \rightarrow P} \left\|P-P'\right\|/t\left(P,P'\right) \tag{3}$$

However, the scan speeds vary in different directions. A practical computation is to use the mean speed along the eight directions instead. As shown in Fig. 4, we can get eight neighborhoods for the point p within a 3×3 window and compute the scan speed for each direction $v_{cam}(P, P^i)$ with Eq. (2). The scan speed at P is thus approximated as

$$v_{cam}(P) \approx \sum_{i=1}^{8} \frac{v_{cam}(P,P^i)}{8} \tag{4}$$

With Eq. (4), we can compute the scan speed for each position in the surveillance space and finally derive a camera scan speed map. Obviously, the scan speed map is dependent on the

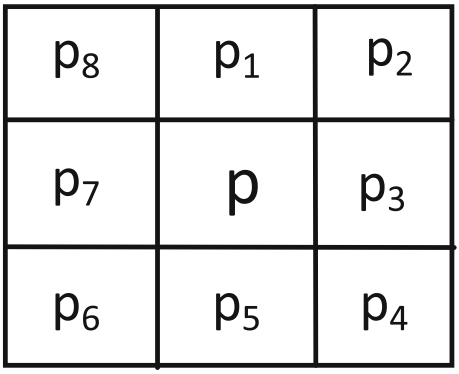

Fig. 4 An arbitrary point P and the eight neighborhoods

camera position. The camera scan speed should be fast enough in order to track a moving object. For the cell-based method, the cameras' FOVs (see Fig. 5) in two neighboring cells should have enough overlap (as shown in Fig. 6) because of two reasons. First, the camera need buffer time to compensate the mechanical delay of pan, tilt, or zooming so as to track the moving object smoothly. Second, the overlap is crucial to have good visual coverage for the object, when the object is at the boundary of two neighboring cells.

However, we don't expect that the FOVs of the cells i and i+2 have any overlap (as shown in Fig. 7) so as to avoid logic confusing when switching and controlling the active cameras. Thus, the overlap ratio should be within the range of:

$$0 < r \leq 0.5$$

In the following, we discuss how to set a reasonable overlap ratio and derive the scan speed requirement to design the system. We first set the forward requirement. When the object crosses the line (see the right black dash line in i+1 cell in Fig. 7) and enters the FOV of cell i+2, the camera viewing cell i should switch to cell i+2 shortly. Before the object enters the cell i+2, the camera should have successfully switched to view cell i+2. Hence, we can set the following constraint for the overlap ratio and the camera scan speed

$$\frac{r}{v_{\max}} \geq \frac{2}{v_{cam}} \tag{5}$$

Re-arrange the above equation and we can get

$$v_{cam} \geq \frac{2}{v_{\max} r} \tag{6}$$

And we also have backward requirement. The object crosses the line (see the right black dash line in cell i+1 in Fig. 7) and immediately decides to go back. However, the camera viewing cell i is already triggered to switch to cell i+2. Before the object moves back to cell i, the camera should have successfully switched back to cell i. Hence, we can have the following constraint from the backward requirement

$$\frac{1-2r+r+r}{v_{\max}} \geq \frac{4}{v_{cam}} \tag{7}$$

Hence, the scan speed should also satisfy:

$$v_{cam} \geq 4 v_{\max} \tag{8}$$

By combing the three requirements above, we can the following constraints for the overlap ratio and scan speed

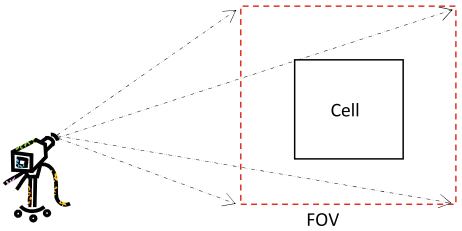

Fig. 5 A cell unit and the field of view (FOV) of an active camera that observes the cell

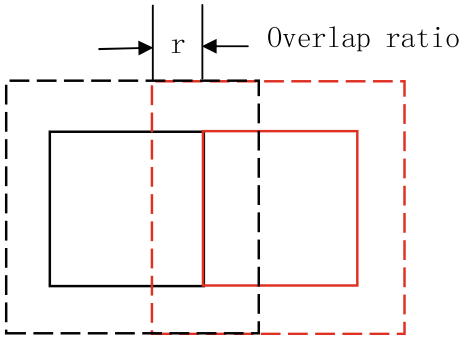

Fig. 6 FOV overlap (the dash squares) of neighboring cells, where r is the overlap ratio (assume unit cell width)

$$\begin{cases} r = 0.5 \\ v_{cam} \geq 4 v_{\max} \end{cases} \tag{9}$$

The above requirements state that the neighboring cells should have half overlap field of view. Moreover, the camera scan speed should be four times faster than the maximum speed of the object. And in the following, we take these two constraints into account to design the active camera system.

3.2 Camera layout, cell partition, and camera control

Camera layout, surveillance space partition with non-uniform cells, and camera control are the three core problems for the cell-based active surveillance [17, 18]. The following constraints are set to solve these three mutually-coupled problems: 1) customized requirements for specific surveillance situations, e.g., the cameras are mounted on the ceiling for corridor surveillance ; 2) mechanical delay of active cameras should be compensated to track the moving object, which means that the camera should have the scan speed four times faster than the maximum object speed; 3) minimal object resolution ($S_{\min}$) requirement, e.g., minimal face image resolution for 3D gaze computation; 4) full object visual coverage; 5) camera's fields of view (FOV) in neighboring cells should have half overlap for object at cell boundary and buffer time for mechanical delay, as derived in Eqs. (9); 6) the space is fully covered by a sequence of square cell (or cell chain) so that each cell has at most two neighbors.

Actually, the above constraints are dependent on the three problems of camera layout, cell partition, and camera control. To take these constraints into account, the

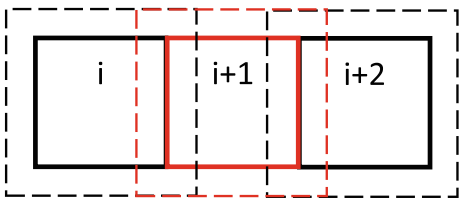

Fig. 7 Three neighboring cells (see the solid squares) and the corresponding FOVs (see the dash squares)

following cost function is defined to evaluate camera layout, cell partition, and camera control

$$R(C,\Theta,K)=\begin{cases}0 & if\ \ S(i,C,\Theta)<S_{\min}\\ \sum_{i=1}^{N}\dfrac{S(i,C,\Theta,K_i)}{N}\end{cases} \tag{10}$$

where C, Θ, and K_i are the camera position, cell partition, and the PTZ parameters of the active camera for the i$^{\text{th}}$ cell with $K=\{K_1,K_2,\cdots K_i\cdots\}$. $S(i,C,\Theta,K_i)$ is the i^{th} face image resolution from N standard faces (e.g., 25 cm×15 cm) that are uniformly distributed in the surveillance space. Hence, $R(C,\Theta,K)$ is the mean face image resolution. The goal is to maximize $R(C,\Theta,K)$ to solve the three mutual-coupled problems of camera layout, cell partition, and camera control as follows.

$$(C^*,\Theta^*,K^*)=\arg\max_{C,\Theta,K}R(C,\Theta,K) \tag{11}$$

We propose a full search method to solve the above optimization problem. As these three problems are greatly dependent to each other, we first search the possible camera layout positions. And for each layout position, the surveillance space is partitioned into cell sequences. Based on the camera layout position and cell partition results, the active cameras are optimally configured with PTZ parameters. As a result, we can compute a score from Eq. (10). Once we try all the camera layout positions, we can finally derive the optimized C, Θ, and K by using Eq. (11). In this paper, each group of cameras is placed in one sampled position (see Fig. 8). As we usually use the identical active cameras in the system, each active camera in the same group have the same layout solution and PTZ parameter configurations.

In order to implement the full search method, we need to solve the three problems: 1) how to define the searching space for camera layout? 2) How to partition the space into cell sequences for a given camera layout; 3) how to optimized setting the PTZ parameters for the active cameras.

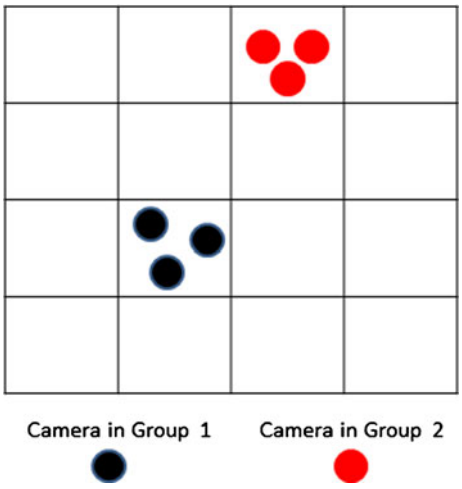

Fig. 8 Sample the space (see the square) for camera layout. One group of cameras is placed in one sampled space. And the two camera groups are place in two different sampled spaces

For the first problem, we need to define a closed searching space so that we can generate finite sampling positions. In practice, the closed searching space can be determined by the minimum object resolution and the customized constraints. For example, the minimum object resolution can determine the maximum distance of the cameras. The customized constraints are the specific requirements from practical surveillance system setup. From these two constraints, we can sample a set of positions to place the two camera groups. In this paper, we try to place one group of cameras into one sampled position. Moreover, the two camera groups are placed in different sampled positions (as shown in Fig. 8). Hence, we can derive two scan speed maps for the two camera groups with the first scan speed map for the first camera group and the second map for the second group. And we assign the first camera group to observe the cells of odd number and the second group for the even number. Note that the number of cameras used in each group depends on specific tasks. And for 3D gaze computation task, at least one camera is required in each group. And for real-time 3D reconstruction task, two cameras or more are required to reconstruct the object.

The cell partition for one camera layout is presented based on the proposed camera scan speed maps. As derived in Section 3.1, the camera scan speed should be at least four-time faster than the maximum object speed. In practice, we try to use the mean speed within a partition cell. Hence, we re-formulate the mechanical delay requirement in camera scan speed for a partition cell as follows

$$\bar{v}_{cam}(C_i) = \sum_{P \in C_i \bigcap S_0}^{M} v_{cam}(P)/M \geq 4v_{\max} \tag{12}$$

where S_0 is the surveillance space. The surveillance space is thus partitioned as follows. We first define a coordinate system from the approximated rectangle (see Fig. 9). And we show the partition of the space with the first cell by using the first camera scan speed map. The range for cell widths is set as follows. The max cell width ($W_{\max}$) is half of the camera FOV with the shortest focal length. The min width ($W_{\min}$) is the height of the rectangle so as to satisfy the sixth constraint. Hence, the range for the cell width is set

$$W_{\min} \leq W \leq W_{\max} \tag{13}$$

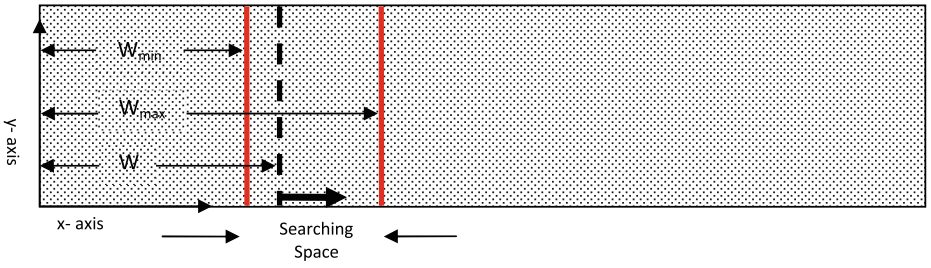

Fig. 9 Search for the cell width by using the camera scan speed map

Hence, we can search from $W_{\min}$ to $W_{\max}$ to compute the cell width until the speed constraint is satisfied (see Fig. 9). The mean scan speed within the area is computed with the following equation

$$s(W) = \frac{\iint_{CS} v_{cam}\left(p_x, p_y\right) dp_x dp_y}{\iint_{CS} dp_x dp_y} \tag{14}$$

where $P=(p_x,p_y)$ and CS is the surveillance space, which is covered by the rectangle (see the area marked by the black dash line) as follows

$$CS : \{P \in S_0 | 0 < p_x < W\} \tag{15}$$

And the cell width is the smallest W that satisfies the scan speed requirement

$$W^* = \underset{s(W) \geq 4v_{\max}}{\operatorname{argmin}} W \tag{16}$$

Once we define the size of the first cell, we get set the x- positions, which is half of the computed cell width. Moreover, we can compute the position along the y- axis as

$$H^* = \underset{CS^* \in square\left(\frac{W^*}{2}, H, W^*\right)}{\arg\min} \|H - y_0\| \tag{17}$$

where CS^* is the partitioned space defined as follows

$$CS^* : \{P \in S_0 | 0 < p_x < W^*\} \tag{18}$$

And y_0 is the y-axis coordinates of the central point of the partitioned space

$$y_0 = \frac{\iint_{CS^*} p_y dp_x dp_y}{\iint_{CS^*} dp_x dp_y} \tag{19}$$

As a result, we can determine the first partition cell the position and the

$$C^1 = square\left(x^1, y^1, W^1\right) = square\left(\frac{W^*}{2}, H^*, W^*\right) \tag{20}$$

After partition with the first cell with the first scan speed map, we can derive an update surveillance space by eliminating the part, which is covered by the first cell. As a result, we can repeat the above procedures to partition the update surveillance space and determine the second cell by using the second scan speed map. In the same way, we can define the remaining cells, until the space is fully covered by the cells. In summary, the steps of cell partition approach are described as follows:

(1) Set the initial surveillance space S_0, represents it with a rectangle, and set the coordinate system;
(2) Compute the two camera scan speed maps for the two camera groups;
(3) For the cell i, if the number i is odd, use the first camera scan speed. Otherwise, use the second speed map;

(4) Search from the minimum to maximum cell widths to compute the cell width by using Eq. (16). If no solution from Eq. (16), the space can be partitioned and exit the process;
(5) Define the cell position from the computed cell width and determine the cell as

$$C^i = square\left(\sum_{n=1}^{i} x^n, y^i, W^i\right) \tag{21}$$

(6) Update the surveillance space $S^{i+1}=S^i-C^i$;
(7) Repeat the steps of (2), (3), and (4) until the space is fully covered by the cells.

Camera control is based on the camera layout C and the cell partition Θ. For example, the zoom is set so that the camera FOV is twice the cell width (as Constraint 5)

$$f = \frac{I_w d}{2W} \tag{22}$$

where I_w is image width and d is the distance to the cell center. The pan and tilt are set so that the optical axis passes the cell center.

For one sampled position, we can compute a score by using Eq. (10) with the solved camera layout, cell partition, and camera control. Once we iterate all the sampled positions, we can compute the corresponding scores by using Eq. (10). The three problems of camera layout, cell partition, and camera control are finally solved by finding the maximum score.

3.3 Cell-based calibration of active cameras

Once we solve the camera layout, camera parameter control, and cell partition problems, we can apply the cell-based method to calibrate the active camera system. In this step, we need to compute the camera intrinsic parameters for each associated cell units (called intrinsic parameter calibration). Moreover, the exterior parameters of the active cameras for different cell units should also be determined, e.g., in a reference coordinate system (called exterior parameter calibration). As a result, we can establish a lookup table, as shown in Table 1 to store the parameters for different cameras in different cells. From Table 1, we can easily control and configure the cameras according to the cells. Furthermore, we can map the 3D features computed from these images into a reference coordinate system.

Because each active camera has fixed PTZ parameters for different cells, we can apply the offline calibration method to determine both the intrinsic and exterior parameters. For example, we can apply the chessboard calibration method proposed by Zhang [20] to compute the

Table 1 PTZ parameter configuration and calibration of active cameras for different cells

Cam\Cell	C^1		C^2		…	C^N	
Cameras in Group I	Config	Calibration	Config	Calibration	…	Config	Calibration
	P_1^1, T_1^1, Z_1^1	K_1^1, R_1^1, t_1^1	N/A	N/A	…	P_1^N, T_1^N, Z_1^N	K_1^N, R_1^N, t_1^N
Cameras in Group II	Config	Calibration	Config	Calibration	…	Config	Calibration
	N/A	N/A	P_2^2, T_2^2, Z_2^2	K_2^2, R_2^2, t_2^2	…	N/A	N/A

intrinsic parameters of each active camera for different cells. In order to determine the exterior parameters, we place the chessboard pattern into the overlap area of two neighboring cells. The relative pose (rotation and translation) are computed by expanding the chessboard plane into a reference coordinate system. Once we compute the relative poses for all neighboring cells, we can map them into a world coordinate system. As a result, all the exterior parameters of all the cameras for different cells are well calibrated. Note that the offline calibration method is not restricted to the chessboard calibration method. It can be other calibration method for different surveillance condition. However, as we use the cell-based conception, we can successfully solve the online calibration of active cameras by using different offline calibration methods.

3.4 Moving object detection and tracking

Object of interest in the surveillance space is detected by combing both the appearance and motion features. In the proposed surveillance system, we are interested in the moving people in the surveillance site. Hence, we can use Haar features for effective human face detection [15]. The detection results are then refined by using motion feature validation for accurate face detection.

The motion features can be detected by using the well-known background subtraction method [14]. One advantage of the proposed system is that all the video images captured by the active cameras have static backgrounds, because we discard all the captured images while the camera is in motion. Hence, the system is algorithm-friendly and we can apply a lot of existing intelligent algorithms based on static background for moving object detection, recognition, and tracking. For example, the background can be efficiently modeled by using the Gaussian Mixture Model (GMM) [14] as follows

$$P(x_N) = \sum_{j=1}^{K} w_j \eta(x_N; \theta_j) \tag{23}$$

where a Gaussian component is represented by the mean and the covariance matrix as follows

$$\eta(x; \theta_j) = \eta\left(x; \mu_j, \sum\nolimits_j\right) = \frac{1}{(2\pi)^{\frac{D}{2}} \left|\sum\nolimits_j\right|^{\frac{1}{2}}} e^{-(x-\mu_j)^T \sum_j^{-1} (x-\mu_j)/2} \tag{24}$$

where μ_j is the mean and $\sum_j = \sigma_j^2 I$ is the covariance of the j^{th} Gaussian component.

In practice, the background is first modeled from the at least one hundred image (s) in advance after the cell-camera configuration and calibration. The background is then updated in real-time. From the background model, the foreground object can be automatically detected and segmented by background subtraction from the input frame. The morphological operations are then performed to further remove the isolated noises and filled the holes inside the objects.

As a result, the object can be accurately detected from the motion pixels and the appearance detection results. Once the object is detected and segmented, it can be tracked with time, e.g., with the well-known "continuously adaptive mean shift" (CAM-Shift) approach [2]. The CAM-Shift algorithm is based on the mean-shift algorithm, which is able to handle dynamic distributions by re-adjusting the search window size for the next frame based on the zero-th moment of the current frames distribution [2].

From the tracking results, we can calculate the object position in the image and map it into the 3D position. In practice, we can use homography mapping to compute the object 3D position by assuming a planar ground plane, which is usually the case for many surveillance

applications. For simplicity, we used the following equation to compute the object position from its image correspondence:

$$P_i^k(t) = map\left(x_i^k(t), y_i^k(t)\right) \tag{25}$$

where k and i are the indexes for the k^{th} cell observed by the i^{th} camera, $\begin{bmatrix} x_i^k(t) & y_i^k(t) \end{bmatrix}^T$ are the object position (unit in pixel) in the image for the corresponding camera and cell. Hence, the computed 3D position $P_i^k(t)$ is in the local camera coordinate system. Note that since we've calibrated each cell for each camera, we can map the 3D coordinate into the world coordinate system real time. The cell-camera calibration allows the system to compute the absolute position of the moving object. Hence, we can calculate the object position in the world coordinate system as:

$$P(t) = R_i^k P_i^k(t) + t_i^k \tag{26}$$

Note that the moving object can be observed by two groups of active cameras at the same time so that we can compute two 3D positions (both in the world coordinate system) from both groups in two cells (neighboring cells). In such a case, we use the mean of them as the computed 3D position. From the 3D position of the object, we can determine the current cell, where the object lies, by using the following mapping equation:

$$C^t = \underset{C^i}{\operatorname{argmin}}\, dist\left(P(t), C^i\right) \text{ with } P_t \in S \tag{27}$$

Note that if the object position is not within the surveillance space, the surveillance task for the object is finished. And we will reset the system to be prepared for the next object. Based on the above definitions, we update the states of the cameras and the cells based on the updated cell as follow:

We define three states for a group of cameras. As we set identical properties of the active cameras in one group, we try to explain the three states by using the state of one active camera in the group: 1) **Work**: the camera view current cell or its neighbors. And the object is within the view zone of the camera; 2) **Motion**: when the camera is the process of changing its pan, tilt, or zooming parameters; 3) **Wait**: the state other than work and motion. The two groups of active cameras are set to view the first two cells, respectively. Hence, the initial states of them are both set to wait states. Among the three states, the active cameras capture video images unless in the motion state.

We also define three states of a cell as follow. 1) **View**: the cell is assigned and observed by one camera; 2) **Alarm**: if the object is within the alarm area and the cell is not in the view state. The view zone of a cell becomes "alarm zone" if the cell is current cell or its two neighbors; 3) **Idle**: the state other than view and alarm. The initial states for the first two cells are set to view states, since they are watched by two camera groups. The other cells are set to idle states.

Once we initialize the states of the cameras and the cells, the system begins to work. The control and switch strategy is then based on the updated cameras and cells' states. From each frame captured, the system will automatically detect if any moving object in the cells. Once a moving object is detected, it is recognized and tracked. Hence, we need to compute the object position from each frame and map it into the world coordinate system. From the computed position, we can update the states of the cameras and cells based on the above definitions. Finally, we can control and switch the two groups of active cameras based on the updated states.

The flowchart of the control and switching of the active cameras is illustrated in Fig. 10. Note that in the sixth step, we check if the object is within the surveillance space. If the object

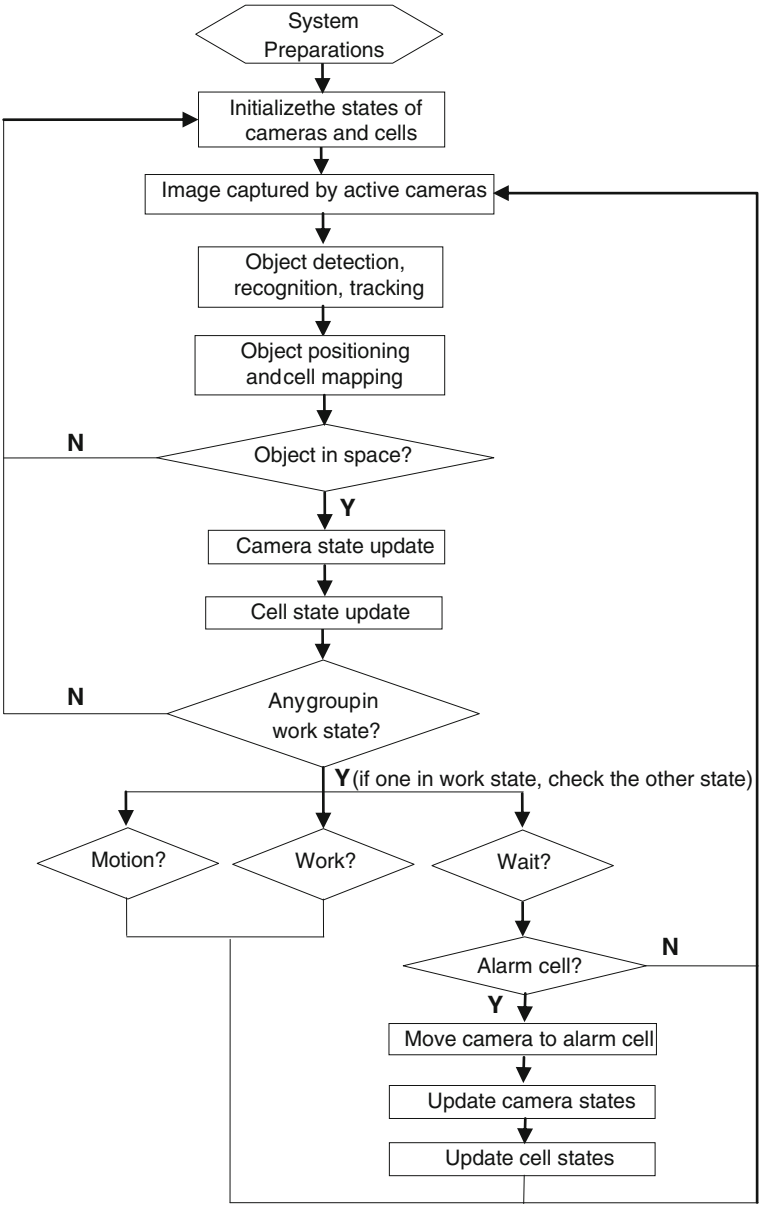

Fig. 10 Object detection and tracking by controlling and switching active cameras across different cells

is out of the space, there are two likely cases: 1) un-expected errors; 2) the object has left the surveillance space and the surveillance job for the object is finished. For either case, we will reset the system and be prepared to monitor the next incoming object.

With the active camera system setup, and the control and switching scheme, the proposed system can monitor moving object within a wide surveillance space and capture high quality video images for visual surveillance. Especially, all the captured images are well calibrated with fine image resolutions for the object, and with static backgrounds.

3.5 3D gaze computation

As we set optimized parameters for each active camera for the corresponding cell unit, the captured images are guaranteed to have high resolution of the object and well calibrated. Also, based on the system design objective, the captured image data allows the accurate computation of the 3D human gazes. The computation of 3D gazes usually requires a good model of human face and camera intrinsic parameters. For example, Shi et al proposed the 3D face model based method to compute 3D gazes with face symmetry prior. And gaze computation requires a relative low face image resolution by reconstructing accurate 3D face model with symmetry prior. More computation details are referred to the literature [13].

The computation complexity of the cell-based surveillance system is discussed here. Among the four parts of the system, the first one (camera layout, cell partition, and camera control) is implemented offline. The computation is hence one-time processing. And the complexity is dependent on the size of the surveillance space, and the possible space to layout the cameras. That is, the sampling number of the space of possible camera layout and the number of standard faces uniformly distributed in the surveillance space are the two key parameters to the complexity. And for the second part (cell and camera calibration part), the computation is also offline and the complexity only depends on the number of cells. The object detection and tracking part is online computation. And the complexity depends on the number of cameras and the number of cells. The same conclusion goes to the gaze computation part, which is also an online computation process.

4 Experimental results

The proposed algorithm and system were tested in an indoor surveillance experiment. The surveillance site is a 1.0×4.1 (m) path in the office (see Fig. 11). We set a rectangle space with 1.6 m height above the floor for partition (about the height of human eyes). We tried to detect the passing people, capture high-quality video images, and extract advanced 3D features (e.g., 3D gazes) for intelligent surveillance.

In the system setup, each active camera set consists of a fire-wire camera, a pan-tilt mounting unit, a tripod, etc. (see Fig. 11), etc. We used two SONY 1,394 DFW VL-500 digital cameras to capture the images. The cameras have the resolution of 640×480 (in pixel). Zooming is automatically controlled by sending commands to the embed motor from the host computer. The zooming values range from 40 to 1,450 units, corresponding to 5.4 to 64.8 mm focal length. For each camera, we used a high-precision Direct Perception PTU-C46 pan-tilt unit so that the camera can perform pan and tilt motions. The PTU model is connected to the host PC by serial port via a control box. It has two degrees of freedom and can perform the pan

Fig. 11 The DFW-VL500 digital camera, surveillance with two active cameras, and the in-door surveillance site

and tilt rotations with high accuracy. The rotation angles for pan and tilt are 180 and 90 degrees, respectively. The speeds for pan and tilt rotations are 50.3 degrees per second. As a result, one camera set can undergo pan, tilt, and zoom operations, to meet different requirements. These two cameras were divided into two groups. And each group has one camera. The proposed system was then applied to monitor passing people inside the surveillance space and extract advanced 3D features, such as 3D gazes, in support of intelligent surveillance. And the maximum walking speed is 3.0 m per second. The pan and tilt speeds are 2.60 r/s, and the zoom speed is 500unit/s, for both active cameras. In this paper, the minimal face resolution for gaze computation is 96×60 (pixel) [13]. All these parameters play important roles to solve the problems of camera layout, cell partition, and camera control for surveillance system design.

We first solved the three problems of camera layout, cell partition, and camera parameter control. As the camera scan speed map is crucial to solve the camera mechanical delay problem, we illustrate two camera scan speed maps with the camera position $[170.0\ 0.0\ 250.0]^T$ and $[125.0, 0.0, 250.0]^T$ (cm) in Fig. 12. As shown in Fig. 12, the dark area represents low scan speed, while the bright for fast speed. We applied the full search strategy to solve the three mutually-coupled problems. The sampling resolution of the space for camera layout is 30×30×30 (cm). The optimal camera position was calculated as $[170.0\ 0.0\ 250.0]^T$ and $[125.0, 0.0, 250.0]^T$(cm) in the coordinate system defined by the surveillance space, as shown in Fig. 12(c), where the y- axis coincides with the ground plane normal. The corresponding camera scan speed maps are shown in Fig. 12. The space was partitioned into four cells with the proposed algorithms. The cell widths are 1.0, 1.0, 1.0, and 1.1 m, respectively. And the neighboring cells have about half meter overlap. The mean scan speeds within the four cells are 956.9, 800.1, 802.1, 977.1 cm/s, respectively, which satisfy the camera scan speed requirement from Eq. (8). And the mean face resolution is 120×90 (pixel), which can be used for 3D gaze computation [13].

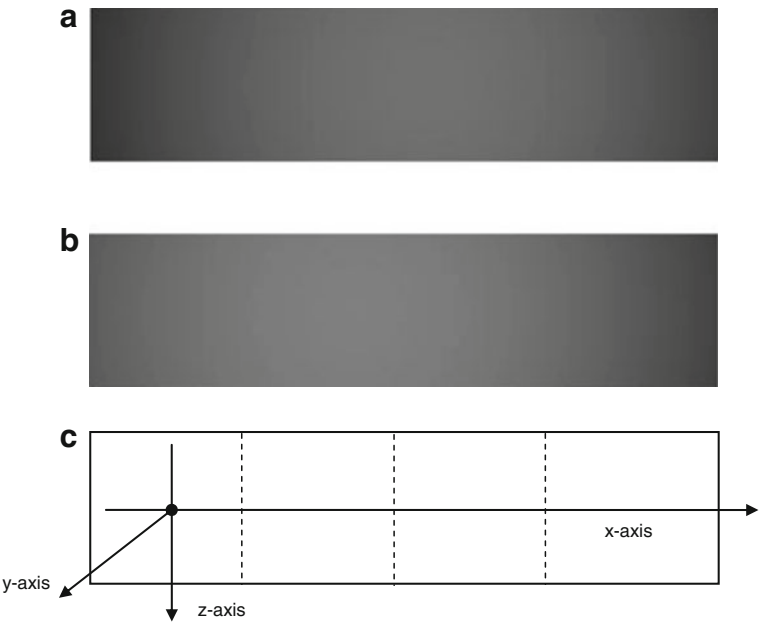

Fig. 12 The camera scan speed maps (dark area for low scan speeds) for two different camera layout positions (**a**) (**b**), and the coordinate system expanded by the rectangle and the cell partition results (**c**)

Table 2 PTZ parameters configuration of active cameras for different cells

Cell No.		I	II	III	IV
Cameras in Group I	Zoom	578	N/A	416	N/A
	Pan	−250	N/A	430	N/A
	Tilt	0	N/A	0	N/A
Cameras in Group II	Zoom	N/A	330	N/A	655
	Pan	N/A	−55	N/A	530
	Tilt	N/A	0	N/A	0

Meanwhile, the cameras were configured with different PTZ parameters for different cells, as presented in Table 2. In the table, we assigned the first camera to observe the first and third cells (odd numbers) and the second camera for the second and fourth ones (even numbers). Table 2 is a lookup table of pan, tilt, and zooming control that was established in advance. According to Table 2, we can online configure and control the active cameras in practice. Note that we also set other parameters of optical imaging, such as focus, white balance, exposure, iris, etc., for active cameras in different cells.

The active camera system was calibrated afterward. We calibrated both intrinsic and exterior parameters of the active cameras for different cells. Figure 13 illustrates the cell-based calibration approach using the traditional chessboard-based calibration method. We tried to calibrate the first camera viewing the first cell by randomly placing the chessboard pattern in the first cell. The intrinsic parameters including camera focal length, principle point, skew, and aspect ratios were determined by using Zhang's method [5]. Figure 14 shows the focal length calibration results by setting different zooming values, as specified in Table 2 above. It can be observed clearly that the relationship between the zoom values and focal lengths are not linear. Hence, it is difficult to establish a uniform relationship to describe the relationship between zoom value and focal length. And the cell based calibration method is a good solution to deal with the online camera calibration problem.

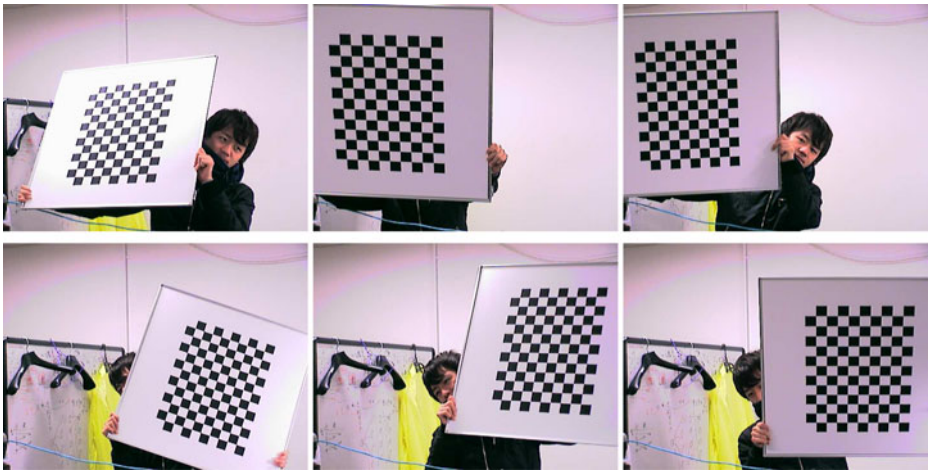

Fig. 13 Cell-based camera calibration of the first camera for the first cell

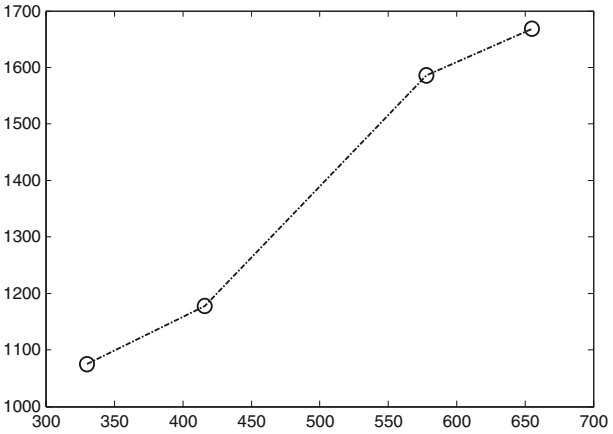

Fig. 14 Focal length calibration results of active cameras, where the x- and y- axes are for the zoom values and calibrated focal lengths, respectively

Calibration of the extrinsic parameters of active cameras for different cells is illustrated in Fig. 15 below. We placed the chessboard pattern and then fixed it in the overlap area of the first and second cells so that it can be observed simultaneously by the first and the second active cameras. A reference coordinate system is then established from the chessboard plane by setting the pattern plane as the X-O-Y plane. Afterwards, the relative poses (both the rotation and translation) can be computed for the two different active cameras with respect to the reference coordinate system. Finally, we could determine the relative pose between the two active cameras for two different cells. For example, the relative pose was computed from Fig. 15 as follows

$$R = \begin{bmatrix} 0.9890 & 0.0214 & 0.0560 \\ -0.0110 & 0.9990 & -0.0025 \\ -0.0561 & 0.0026 & 0.9883 \end{bmatrix}; \quad t = \begin{bmatrix} 455.2 \\ -90.1 \\ -47.6 \end{bmatrix}$$

By calibrating the active cameras for all different cells, we computed the relative poses for the cameras viewing all neighboring cells. And finally they can be mapped into a world coordinate system. Therefore, we can map the computed 3D features, such as object 3D positions, gazes, etc., into a reference frame in practice.

With the system designed and calibrated, it is ready to capture high-quality images for advanced 3D feature computation. The GMM method was applied to model the static background for each cell-camera unit (see Fig. 16(a)), from which the motion pixels were subtracted from the raw image (see Fig. 16(b)) [14]. The results were combined with those from the Ada-

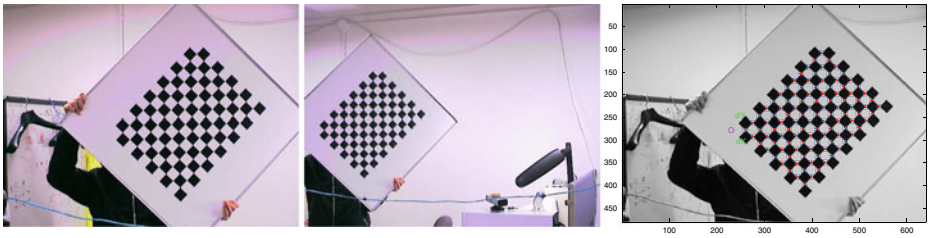

Fig. 15 Exterior parameter calibration for neighboring cells with chessboard board pattern

Boost method for accurate face detection (see Fig. 16(c)) [15]. The detected face was further tracked and the image positions were recorded [15]. The 3D positions were calculated in the 3D reference frame and mapped into the cell. The states of the active cameras and cells were then updated. The two groups of active cameras were controlled and switched following Fig. 10. The system could track the moving people with arbitrary trajectory inside the area as long as the walking speed is less than 2.0 m/s. The mean face resolution is 120×90 (pixel). All the images have static backgrounds (see Fig. 17). Figure 17(a)-(d) shows the captured images of a person moving through the surveillance area from the first cell to the fourth one. The two groups of active cameras were initially set to view the first and second cells, respectively. The cameras adaptively switched to the third and fourth cells according to the object positions. Human faces were detected by combing the appearance and motion features, and then tracked in different frames (see the detection and tracking results marked by red square in Fig. 17).

The performance of the proposed system was evaluated in term of object image resolution, surveillance space, visual coverage, and camera calibration. Especially, we compared the proposed system with the single fixed camera and two fixed camera systems. The results are presented in Table 3. It is expected that the performance of the proposed system can be significantly improved for larger surveillance space, which can is partitioned with more cells. Figure 18 shows two typical real face images captured by the proposed system and by using single fixed camera, respectively. The image captured by the proposed system has much better texture details with high face image resolution. And the image captured by single camera is not capable for advanced 3D feature extraction due to the low face image resolution. It clearly shows that the proposed system can capture better quality images for intelligent surveillance.

The captured images by the proposed system can be further used for advanced 3D feature computation. In the experiment, the 3D gazes were computed from the captured face images [13]. Figure 19 shows the two images captured simultaneously by the active cameras while the object was in the overlap area of the third and fourth cells. The gazes of the right eye in the two images captured by two cell-camera units simultaneously are $[-0.333\ -0.100\ 0.937]^T$ and $[-0.282\ -0.106\ 0.955]^T$. And the gazes of the left eye were computed as $[-0.2857\ -0.0357\ 0.9576]^T$ and $[-0.3929\ -0.0714\ 0.9168]^T$. The 3D gazes are drawn in the standard front face images in Fig. 19(c), where the green and red arrows are for the results computed from the first and second images, respectively (they are almost overlapped due to very tiny angles in-between). The two computed gazes should be ideally identical, as they were computed from the same people at the same time. And the angle between gazes computed from the left eye is 2.8 degrees. And the gaze angle of the right eye is 3.6 degrees. The results demonstrate that the proposed system can capture good-quality video images, which are sufficient and effective for 3D gaze computation.

Fig. 16 (**a**) the background modeling with GMM; (**b**) motion segmentation from background subtraction; (**c**) accurate face detection from motion and appearance features

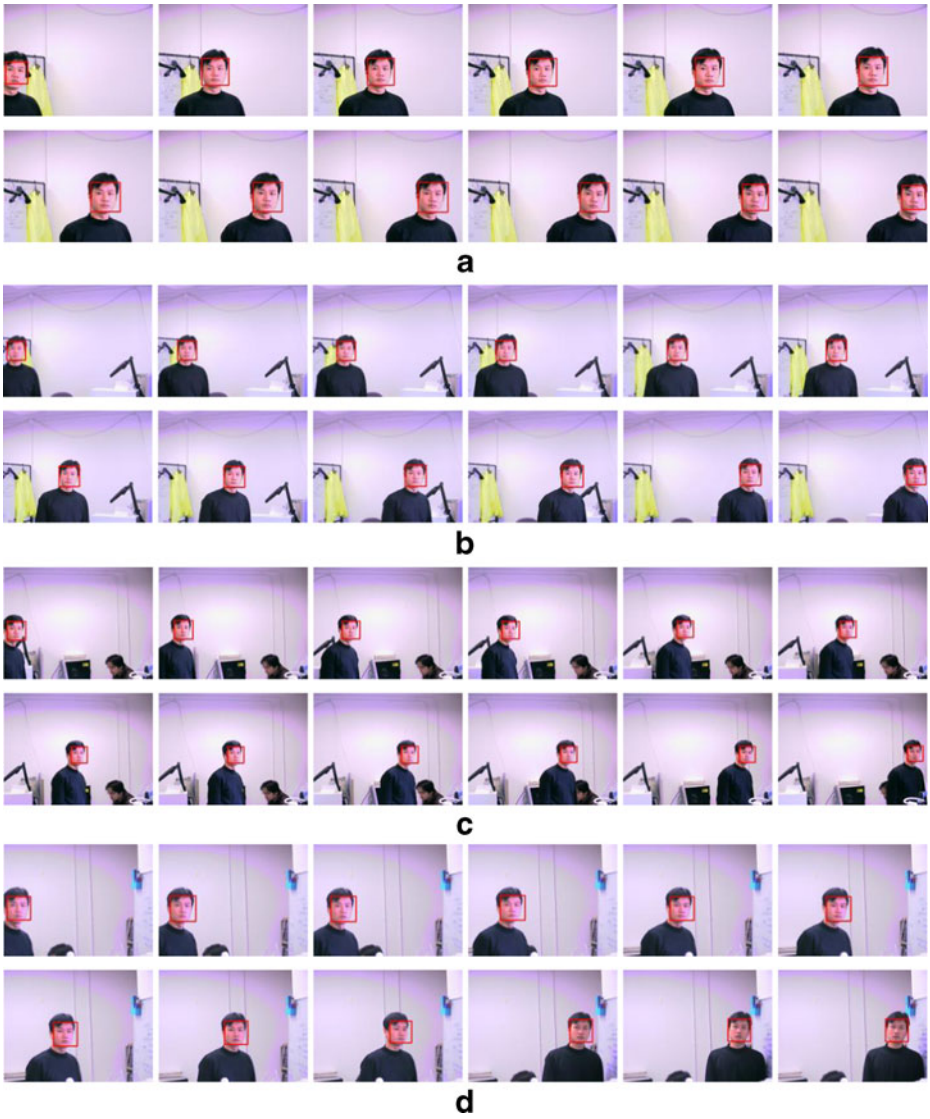

Fig. 17 Video images captured by active cameras in the four cells (from (**a**) ~ (**d**)), when a person walked through the surveillance space from the first cell to the fourth one with face detected and tracked

Table 3 Performance evaluation and comparison

The performance	The proposed system	Single-camera system	Two-camera system
Facial Resolution (pixel)	120×90	70×50	80×55
Surveillance Space (m)	4.1	2.8	4.1
Image calibrated?	Yes	Yes	Yes
Object coverage?	Yes	Yes	Yes

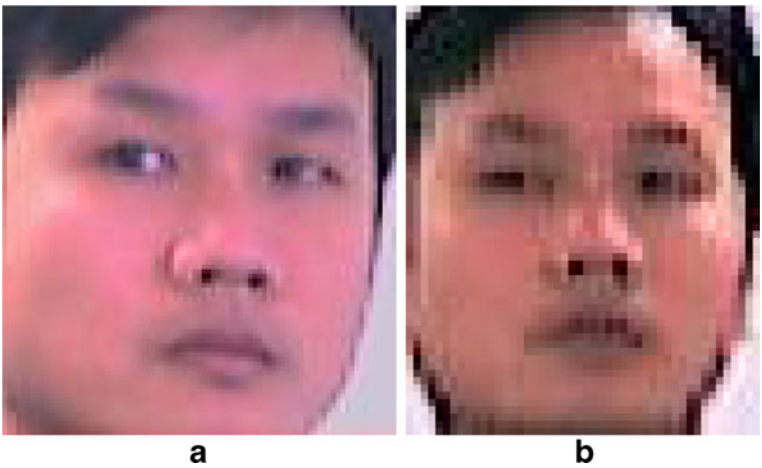

Fig. 18 Face images captured by the proposed system (**a**) and by using single fixed camera (**b**), respectively

5 Conclusions and recommendations

We've proposed a framework of cell-based surveillance for advanced 3D feature extraction. We addressed the problem of cell-based visual surveillance with multiple active cameras for a path-like space, which is partitioned into cell sequence. We proposed the conception of camera scan speed map (CSSM) that is particularly effective to tackle the mechanical delay problem for active camera system design. We formulated the camera layout, cell partition, and camera control for N (≥2) active cameras into an optimization problem by maximizing the mean object image resolution while meeting various constraints such as system mechanical delay, full visual coverage, minimum object resolution, etc. The optimization problem is solved by a full search method. The cell-based calibration is applied for system calibration to effectively solve the active camera online calibration problem. We proposed the object detection by combing appearance and motion features, and switch and control the multiple active cameras across different cells to track the object for high quality image capture. The proposed system and algorithms have been validated in a real surveillance experiment with satisfactory results presented.

Future work based on the cell-based conception proposed in this paper is recommended as follows. First, the proposed system and algorithms will be applied for advanced road traffic

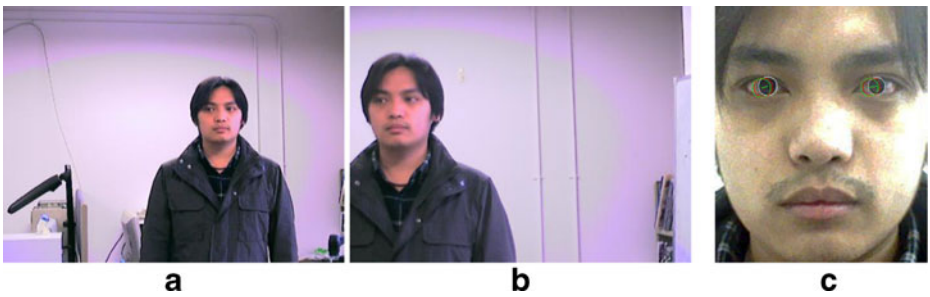

Fig. 19 3D Gaze computation results (**c**) from the face images (**a**), (**b**) simultaneously captured in the 3rd and 4th cells by different active cameras

surveillance. Especially, we are working forward for high-resolution imaging and advanced 3D feature extraction for vehicle or pedestrians of interest on the road. As the computation objective is different from that in the proposed paper, the objective function need to be modified to incorporate more performance parameters. However, the cell-based conception and the camera switching and control scheme are similar with those in the proposed paper. Second, the cell-based conception will be applied to more general and complex surveillance space in addition to the proposed path-like one, such as a square space. In the cell-based framework, the cell partition and camera switching and control scheme may need to be modified to fulfill different surveillance tasks. And we believe that the proposed cell-based surveillance system suggests a novel way to enhance visual surveillance.

Acknowledgments The work presented in this paper was sponsored by grants from National Natural Science Foundation of China (NSFC) (No. 51208168), Tianjin Natural Science Foundation (No. 13JCYBJC37700), the Youth Top-Notch Talent Plan of Hebei Province, China, and the Grant-in-Aid for Scientific Research Program (No. 10049) from the Japan Society for the Promotion of Science (JSPS).

References

1. Bellotto N, Benfold B, Harland H, Nagel HH, Pirlo N, Reid I, Sommerlade E, Zhao C (2013) Cognitive visual tracking and camera control. Comput Vis Image Underst 116(3):457–471
2. Bradski GR (1998) Computer vision face tracking for use in a perceptual user interface, Intel Technology Journal, Q2
3. Chen KW, Lin CW, Chiu TH, Chen YY, Hung YP (2011) Multi-resolution design for large-scale and high-resolution monitoring. IEEE T Multimed 13(6):1256–1268
4. De D, Ray S, Konar A, Chatterjee A (2005) An evolutionary SPDE breeding–based hybrid particle swarm optimizer: application in coordination of robot ants for camera coverage area optimization, PReMI, pp 413–416
5. Erdem U, Sclaroff S (2006) Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. Comput Vis Image Underst 103(3):156–169
6. Hartley R, Zisserman A (2004) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge
7. Hu W, Tan T, Wang L, Maybank S (2004) A survey on visual surveillance of object motion and behaviors. IEEE T SMC C 34(3):334–352
8. Loy C, Xiang T, Gong S (2011) Detecting and discriminating behavioral anomalies. Pattern Recogn 44(1): 117–132
9. Matsuyama T, Ukita N (2002) Real-time multi-target tracking by a cooperative distributed vision system, Proceedings of the IEEE, 90(7):1136–1150
10. Morris BT, Trivedi MM (2008) A survey of vision-based trajectory learning and analysis for surveillance, IEEE Transactions on Circuits and Systems for Video Technology, 18(8):1114–1127
11. Saini M, Atrey PK, Mehrotra S, Kankanhalli M (2012) W3-privacy: understanding what, when, and where inference channels in multi-camera surveillance video, Multimedia Tools and Applications
12. Sankaranarayanan K, Davis W (2011) Object association across PTZ cameras using logistic MIL, IEEE Conf. CVPR, pp. 3433–3440
13. Shi Q, Nobuhara S, Matsuyama T (2012) 3D face reconstruction and gaze estimation from multi-view video using symmetry prior. IPSJ T Comput Vis Appl 4:149–160
14. Stauffer C, Grimson WEL (1999) Adaptive background mixture models for real-time tracking, IEEE Conf. CVPR, pp.246–252
15. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features, IEEE Conf. CVPR, pp.511–518
16. Wang XG (2013) Intelligent multi-camera video surveillance: a review. Pattern Recogn Lett 34(1):3–19
17. Yamaguchi T, Yoshimoto H, Matsuyama T (2010) Cell-based 3D video capture method with active cameras, in Image and Geometry Proc. for 3-D Cinematography, pp. 171–191, Springer
18. Yamaguchi T, Yoshimoto H, Nobuhara S, Matsuyama T (2010) Cell-based 3D video capture of a freely-moving object using multi-viewpoint active cameras. IPSJ T Comput Vis Appl 2(8):169–184

19. Yonetani R, Kawashima H, Hirayama T, Matsuyama T (2010) Gaze probing: event-based estimation of objects being focused on. ICPR, pp 101–104
20. Zhang Z (2000) A flexible new technique for camera calibration. IEEE T Pattern Anal Mach Intell 22(11): 1330–1334
21. Zheng WS, Gong SG, Xiang T (2011) Person re-identification by probabilistic relative distance comparison. CVPR, pp 649–656

Zhaozheng Hu received the Bachelor and Ph.D. degrees from Xi'an Jiaotong University, China, in 2002 and 2007, respectively, both in information and communication engineering. During his PhD period, he visited the computer vision lab at the Chinese University of Hong Kong from Nov., 2004 to Jun., 2005. From Jul., 2007 to Aug., 2009, he worked as a Post-doc Fellow in Georgia Institute of Technology, U.S.A. From Jul., 2010 to Jul., 2012, he was with the visual information processing lab in Kyoto University, Japan, under a JSPS Fellowship program. He is currently a professor in Wuhan University of Technology, Wuhan, China. His research topics mainly focus on visual geometry, stereo vision, intelligent transportation system, intelligent surveillance system, etc.

Takashi Matsuyama received the BEng, MEng, and DEng degrees in electrical engineering from Kyoto University, Japan, in 1974, 1976, and 1980, respectively. Currently, he is working as a professor in the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University. His research interests include knowledge-based image understanding, computer vision, 3D video, human-computer interaction, and smart energy management. He has written more than 100 papers and books, including two research monographs: A Structural Analysis of Complex Aerial Photographs (Plenum, 1980) and SIGMA: A Knowledge-Based Aerial Image Understanding System (Plenum, 1990). He won 10 best paper awards from

Japanese and international academic societies including the Marr Prize at ICCV '95. He is on the editorial board of the Pattern Recognition Journal. He was awarded fellowships from the International Association for Pattern Recognition, the Information Processing Society of Japan, and the Institute of Electronics, Information, and Communication Engineers Japan. He is a member of the IEEE.

Shohei Nobuhara received his B.Sc. in Engineering, M.Sc. and Ph.D. in Informatics from Kyoto University, Japan, in 2000, 2002, and 2005 respectively. From 2005 to 2007, he was a postdoctoral researcher at Kyoto University. Since 2007, he has been a research associate at Kyoto University. His research interests include computer vision and 3D video. He is a member of IPSJ, IEICE, and IEEE.