Content and task-based view selection from multiple video streams

Fahad Daniyal · **Murtaza Taj** · **Andrea Cavallaro**

Received: date / Accepted: date

Abstract We present a content-aware multi-camera selection technique that uses object- and frame-level features. First objects are detected using a color-based change detector. Next trajectory information for each object is generated using multi-frame graph matching. Finally, multiple features including size and location are used to generate an object score. At frame-level, we consider total activity, event score, number of objects and cumulative object score. These features are used to generate score information using a multivariate Gaussian distribution. The algorithm. The best view is selected using a Dynamic Bayesian Network (DBN), which utilizes camera network information. DBN employs previous view information to select the current view thus increasing resilience to frequent switching. The performance of the proposed approach is demonstrated on three multi-camera setups with semi-overlapping fields of view: a basketball game, an indoor airport surveillance scenario and a synthetic outdoor pedestrian dataset. We compare the proposed view selection approach with a maximum score based camera selection criterion and demonstrate a significant decrease in camera flickering. The performance of the proposed approach is also validated through subjective testing.

Keywords Content scoring · information ranking · feature analysis · camera selection · content analysis · autonomous video production

1 Introduction

Multi-camera settings are becoming increasingly common in scenarios ranging from sports to surveillance and smart meeting rooms. An important task is the quantification of view quality to help select a single camera or a subset of cameras for optimal observability. The

F. Daniyal
Multimedia and Vision Group - Queen Mary University of London, United Kingdom, E1 4NS, UK
Tel.: +44-20-7882-5259
E-mail: fahad.daniyal@elec.qmul.ac.uk

M. Taj
E-mail: murtaza.taj@elec.qmul.ac.uk

A. Cavallaro
E-mail: andrea.cavallaro@elec.qmul.ac.uk

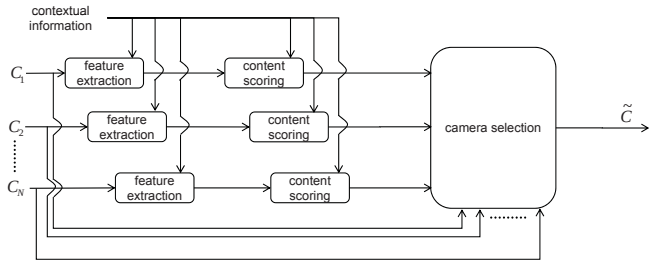


Fig. 1 Block diagram of the proposed approach, composed of three main blocks i.e. feature extraction, content scoring and camera selection.

ranking of each view can assist in applications such as summary production, highlight generation, and action replay. Although extensive work has been done in autonomous content production, camera selection and scheduling for site or target monitoring (Table 1), not much literature is available on the quantification of the quality of a view. View quality is dependent on the task at hand and on the features that one aims to observe [5]. Hence the overall view selection problem splits into three sub-problems (Fig. 1): (i) to analyse the features of each frame in each video stream; (ii) to assign a score to each camera that is dependent on context and scene content; (iii) to select the camera based on the calculated score and contextual information. The selection should be done such that frequent and short switches are avoided.

The contributions of this paper are twofold. First we introduce a novel quality measure based on the combination of local and global features, which is used to assign a *visibility score* to each view over time. The assignment of this score is based on a Gaussian observation model which can be used to select a single camera at any given time. The second contribution is the use of scene-centric state modeling. This modeling is based on a Dynamic Bayesian Networks (DBN) to integrate camera network information as a prior for selection. DBN allows us to achieve an optimal number of camera switches by enforcing temporal smoothing.

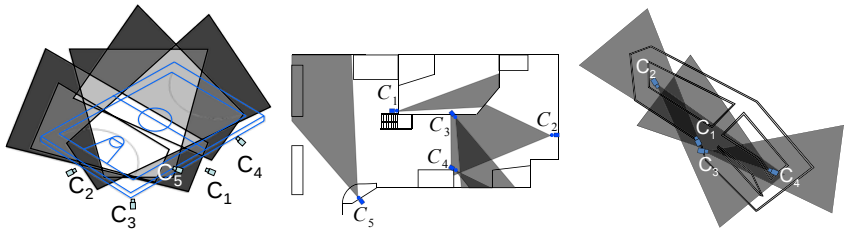
The organization of the paper is as follow. Section 2 discusses the state-of-the-art in content ranking and camera selection. In Sec. 3 we define the problem of content scoring and best view selection. Section 4 describes the features extraction procedure. Section 5 deals with event detection. Section 6 describes the proposed algorithms for feature merging and content scoring. Methods employed for view selection are discussed in Sec. 7. In Sec. 8 we discuss the experimental results. Finally, in Sec. 9 we draw conclusion.

2 Related work

The state-of-the-art for best-view selection can be divided into two parts, namely methods on content analysis and ranking and methods on camera selection and view planning across cameras.

2.1 Content ranking

Content ranking involves the extraction of features and their ranking across time and across cameras. The choice of these features and the ranking criterion are driven by the task at



(a) (b) (c)

Fig. 2 Configuration of the cameras in (a) sports (basketball court); (b) surveillance (airport [22]) and (c) a simulated road scenario.

Table 1 State of the art on best-view selection (FS:resilience to frequent camera switching; FOV: field of view; KTSP: Kinetic Traveling Salesman Problem; POMDP: Partially Observable Markov Decision Process)

Ref.	Features	Method	FS.
[27]	Camera pose & configuration, task constraints	Volumetric intersection	No
[10]	Object size, pose & orientation	Dynamic Programming	Yes
[5]	Object size, deadline, orientation, events	Feature visibility & weighting	No
[9]	Object position, appearance & pose	Heuristic-based greedy programming	No
[15]	Event probability and priority	Event ranking	No
[12]	Target and camera location	Automatic determination of FOV lines	No
[2]	Tracking predictions and appearance	KTSP with deadlines	Yes
[4]	Target location and velocity	Greedy scheduling	No
[18]	Object detections	Weighted round robin	No
[6]	Path observations, activity patterns	Incremental learning	No
[13]	Tracking, occlusion	Markov Chains	Yes
[23]	Target appearance	Weighted fusion of features	No
[19]	Estimation entropy	POMDP	Yes

hand and the network configuration. In [23] target appearance is used as a ranking measure. The weighted average of each color channel in the segmented pixels is used to select the most appropriate set of sensors. The goodness of the proposed approach is demonstrated in terms of detection and tracking error through the selected sensor. Although this approach is general enough for wide variety of scenarios, being based only on low-level features it results in higher ranks whenever there is a change in the scene that does not necessarily correspond to an interesting activity.

The view angle and the distance between target and sensor are other features used to determine the quality of a view obtained from multiple overlapping sensors. In [21] frontal images of each person are captured as they enter the scene. This work lacks a formulation for multiple targets and is applicable only to a single pedestrian. Moreover the view angle estimation draws heavily on the 3D knowledge of the scene (both in terms of camera and the target), which may not be always available. In [3], visibility factor is associated to an occlusion metric and hence a camera configuration that minimizes occlusion is chosen. A similar method is proposed in [27], where the visibility is measured in terms of observability of features of interest. Features of interest in the environment are required to simultaneously

be visible, inside the field of view, in focus, and magnified as per the specification of the task at hand. Features extracted from the track information are fused to obtain a global score to mark regions as interesting on the basis of the quality of view.

In [7] the rank for each sensor is estimated as a weighted sum of individual features of each target. The combination of features (blob area, detection of face, its size and the direction of motion of the target) is used to rank individual object features. Since this approach does not incorporate high-level scene analysis (e.g., event detection), it is very likely that the highest rank may be given to the sensor with the largest number of targets, whereas sensors with fewer targets but containing interesting or abnormal behaviors may be ignored. Similarly the problem of content ranking is considered in [5] as an observation on multiple features. Ranking is performed on the basis of the features and events associated to each object. The overall approach in this work is deadline driven and based on a constant direction motion model. The work in [15] uses event information as a cue to select the best view where the event recognition probability and the information about its importance are used as selection criterion.

In [16] a dual camera system is proposed for indoor scenarios with walking people. Targets are repeatedly zoomed in to acquire facial images using a supervised learning approach driven by skin, motion and detectability of features. Priors such as size (height) and the motion path of the target are set to narrow the choice in selection of targets in the scene. However this work lacks a formulation for target scheduling and its performance degrades in crowded scenes. In [2, 18], target features such as gaze direction and motion dynamics are used to compute the minimum time that the target will remain in the monitored area (deadline). This deadline can be used to assign weights to active cameras in order to decide which sensor can attend to the target with minimum adjustment cost [18].

In [29] the occupancy map of the objects is constructed and used as shape approximation. Several features are then extracted from the occupancy map (distance covered, speed, direction, distance from the camera center, visibility and face visibility). The construction of the occupancy map reduces the amount of noise due to spurious detections. A similar voxel representation is used for different body parts in [9] to determine pose of the target in order to obtain its probability of visibility.

2.2 Camera selection

Camera selection takes into account physical constraints such as the scheduling interval, orientation speed (in case of an active camera) and location of sensors in the network. In [2] Time Dependent Orienteering (TDO), target motion, position, target birth and deadline are used to trigger a PTZ camera that capture targets in the scene. To minimize the number of switches a scheduling interval is used. The cost of the system is associated to the number of targets not captured. The scheduling strategy used is Kinetic Traveling Salesperson Problem with deadlines. A schedule to observe targets is chosen which minimizes the path cost in term of TDO. This work does not consider target occlusions and does not provide any formulation for the prediction of the best time intervals to capture images. In [10] a cost function is proposed that depends on the view quality measures using features such as object size, pose and orientation. This cost function also includes a penalizing factor to avoid frequent switches.

In [31] a single person is tracked by an active camera and when there is more then one person in the view of the static camera, the active camera focuses on the closest target. The performance of the system degrades in case of crowded scenes as the camera switches from

target to target. In [17], a surveillance system is proposed, comprising of passive cameras with wide field-of-view and an active camera which automatically captures and labels high-resolution videos of pedestrians. A Weighted Round Robin technique is used for scheduling each target that enters the monitored area. The approach is scalable both in terms of number of cameras and targets.

The work in [4] uses greedy scheduling policies to observe people where targets are treated as network packets and a routing approach based on techniques such as First Come First Served (FCFS), Earliest Deadline First (EDF) and Current Minloss Throughput Optimal (CMTO). However these approaches do not include the transition cost for the camera that is associated with target swaps. A system for automatically acquiring high-resolution images by steering a pan-tilt-zoom (PTZ) camera is described in [20]. The system uses calibrated master cameras to steer slave cameras. However, in case of multiple targets, the slave PTZ cameras focuses on the target which was detected first in the scene and then based on the arrival time of the targets subsequent scheduling of targets is done. In [8] a ceiling mounted omni-directional camera provides input for a PTZ camera mounted at head height to capture facial images of the targets moving in the scene. No formulation is provided to tackle lost objects and for cluttered scenes there is significat degradation in the performance of the system. In addition there needs to be accurate calibration between the master and slave cameras.

The work in [1] concentrates on active tracking: a simple behavior (a policy) with a finite state machine is defined in order to give some form of continuity when the currently tracked target is changed. Authors in [19] propose the use of Partially Observable Markov Decision Processes (POMDP) to estimate the state of the target at any time and select a camera configuration so that estimation error in detecting the state of the target is minimized. Scheduling interval is used to observe targets for a duration of time. However they do not take into account target interactions with the environment and no formulation is provided for multiple targets.

3 Problem formulation

Let a set of N cameras $C = \{C_1, \ldots, C_N\}$, with camera C_i at time t observe a set of $J_i(t)$ targets $O_i(t) = \{O_{i1}(t), \ldots, O_{iJ_i}(t)\}$. The problem of view selection consists in deciding the best view $\tilde{C}(t)$ at any time t such that features of interest are visible [27] and/or maximized [3]. Such a view is likely to contain information about the scene which is of most interest, given the site contextual information and camera network information. Let us define the set of features observed by each camera C_i as $\psi_i(t)$. Based on these features, a score $\rho_i(t) = f(\vartheta_i, \psi_i(t))$ is assigned to it, where ϑ_i is a set of parameters for camera C_i that encode the contextual information regarding the site. This score helps selecting camera $\tilde{C}(t) \in C$ at each time instant t. In order to avoid frequent switches and generate a pleasant view, let us consider the cameras as a set of states of the system. Then the problem is to find the most likely state based on a observations vector $\overline{\rho}(t) = (\rho_1(t), \ldots, \rho_N(t))$, where $\rho_i(t)$ is the score for camera C_i at time t.

The problem of camera selection can thus be regarded as a three-tier system (Fig. 1). In the first stage, the extraction of a feature set $\psi_i(t)$ at time t for each object in each camera C_i, $\forall\ i = 1, \ldots, N$ is performed. In the second stage, the features are used to generate a camera score $\overline{\rho}(t)$. In the final stage, a selection mechanism is constructed as a function of time t and the rank $\overline{\rho}(t)$ to select $\tilde{C}(t)$.

(a) (b) (c)

Fig. 3 Example of background learning result. Starting from he foreground objects are progressively removed: (a) frame 0; (b) frame 100; (c) frame 250.

(a) (b) (c) (d)

Fig. 4 Activity detection results for two cameras. (a,c): input images. (b,d): binary activity masks.

We use object as well as scene-centric features to represent the information being observed by each camera within the site. Initially, the amount of motion or activity $d_i(t)$ for C_i at time t is computed. Then objects of interest are detected and associated across frames. The size and the location of the object are considered as features of interest. The size feature $s_{ij}(t)$ used in this work is a linear function of width and height of object (see Sec. 4.3). The site is divided into regions based on their importance. Each object is assigned a location score $\lambda_{ij}(t)$. In addition, events of interest within the field of view of each camera are detected. Event detection is done using both low and high level features.

In the following sections we provide a detailed description of the methodology involved in the extraction of these features.

4 Feature extraction

4.1 Amount of activity

The scalar value used to express the amount of activity is the number of foreground pixels normalized by the image size and is represented as $d_i(t) = \left|I_i^d(t)\right|$ for camera C_i at time t. The segmentation of an image into background and foreground is performed using a color change detector [25]. A reference image I_i^{ref} is first generated for each camera C_i using adaptive background learning (see Fig. 3). Let $I_i(t)$ be an input image from camera C_i at time t, then the difference image $I_i^d(t)$ at time t is calculated as $I_i^d(t) = \left|I_i^{ref}(t) - I_i(t)\right|$. $I_i^d(x,y,t)$, a pixel at location (x,y) at time t in $I_i^d(t)$, is classified as foreground or background based on dynamic thresholding. Sample results for the activity detection are shown in Fig. 4.

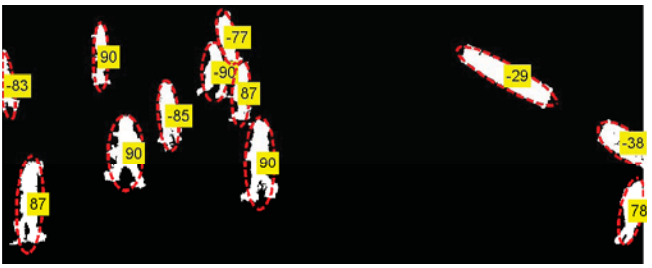


Fig. 5 Sample image for object detection. Objects not meeting the criterion for size and orientation are classified as noise or spurious detections.

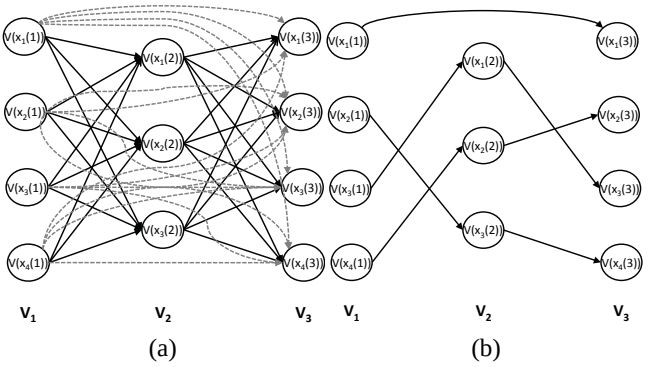


Fig. 6 Example of digraph G for 3 frames motion correspondence: (a) the full graph; (b) a possible maximum path cover.

4.2 Object detection and tracking

Contextual information about the site is exploited after the activity detection to classify foreground objects as real targets or spurious objects. This contextual information includes the expected width, height and orientation of the target given its location. When observing humans, in most cases a person is upright and therefore we only consider detections with an upright major axis (see Fig. 5). However this detection module can be replaced with other approaches that incorporate target modeling [11].

Next, data association links different instances of the same object over time (target tracking). Target tracking is done using a graph based approach that uses multiple object features to establish correspondence. Let us assume that we detect $M_i(t)$ candidate targets in camera C_i at time t, where $M_i(t) \neq J_i(t)$. Let this set of $M_i(t)$ detections be represented as $\mathbf{X_i}(t) = \{X_{ij}(t)\}_{j=1\ldots M_i(t)}$ at time t and $v(\mathbf{x}_a(t)) \in V(t)$ be the set of vertices representing the detected targets at time t. Each $v(\mathbf{x}_a(t))$ belongs to $G = (V,E)$, a bi-partitioned digraph (i.e. a directional graph), such as the one reported in Fig. 6 (a). The candidate correspondences at different observation times are described by the gain g associated to the edges $e(v(\mathbf{x}_a(t)), v(\mathbf{x}_b(\bar{t}))) \in E$ that link the vertices such that $t \neq \bar{t}$.

The gain g between two vertices is computed using the information in $\mathbf{X_i}(t)$, where the elements of the set $\mathbf{X_i}(t)$ are the vectors $\mathbf{x_a}(t)$ defining $\mathbf{x}$, the 4D state of the object $\mathbf{x} = (x, y, \dot{x}, \dot{y}, w, h)$. Here (x,y) is the center of mass of the object, $(\dot{x},\dot{y})$ are the vertical and

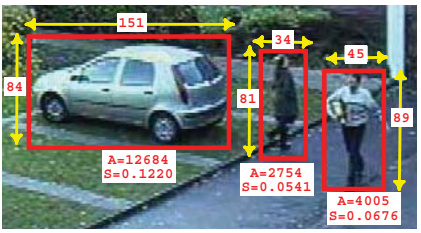


Fig. 7 Object sizes (s) for multiple objects and multiple classes. Using the size measure compared to an area measure reduces the inter- and intra-class size gap across objects.

horizontal velocity components in the image plane and (w,h) are the width and height of the bounding box. The velocity is computed based on the backward correspondences of the nodes. If a node has no backward correspondence, then $(\dot{x},\dot{y})$ are set to 0. The best set of tracks is computed by finding the maximum weighted path cover of G (see [25] for more details).

4.3 Object size

To consider the visibility of the object $O_{ij}(t)$ at time t we take into account the *size* of the target. However conventional measures for size (e.g., the blob area, or the area of the bounding box itself) give higher values to objects closer to camera and thus these objects are regarded as *interesting*, which may not always be the case. Authors in [31, 21] normalize the blob area by the distance between the object and the camera. However this requires 3D information about site and the camera, which may not always be available. Moreover when we consider a scenario with multiple object-classes (Fig. 7), bigger objects (car) will always be ranked higher than pedestrians. To overcome this shortcoming we rescale the object area with half of its perimeter thus converting the square area measure to a linear measure, i.e.,

$$s_{ij}(t) = \frac{1}{A_i}.\frac{w_{ij}(t)\times h_{ij}(t)}{w_{ij}(t)+h_{ij}(t)}, \tag{1}$$

where A_i is the imaging area of the camera C_i, w_{ij} and h_{ij} refer to the width and height of the j^{th} object respectively when viewed from C_i. As an example, consider multiple object-classes as shown in Fig. 7; the pixel area of the car ($A_c = 12684$) compared to pedestrians ($A_{p1} = 2754$, $A_{p2} = 4005$) is considerably larger ($A_c/A_{p1} = 4.6051$, $A_c/A_{p2} = 3.167$, $A_{p1}/A_{p2} = 0.6876$). As a result frames containing the car will always have a higher score than those containing pedestrians alone. In contrast to this, using the size feature introduced in Equation 1 this difference is reduced ($s_c/s_{p1} = 2.25$, $s_c/s_{p2} = 1.804$, $s_{p1}/s_{p2} = 0.8003$).

4.4 Object location

An observed site can be divided into regions of varying importance based on the task at hand. In a sports scenario these could be the regions near the goal or basket. In surveillance scenarios these can be the entry and exit zones of the site. Objects in these regions would be of higher significance than objects elsewhere. To this end, the monitored site is divided into K non-overlapping regions on a common reference plane. We take the ground plane (π)

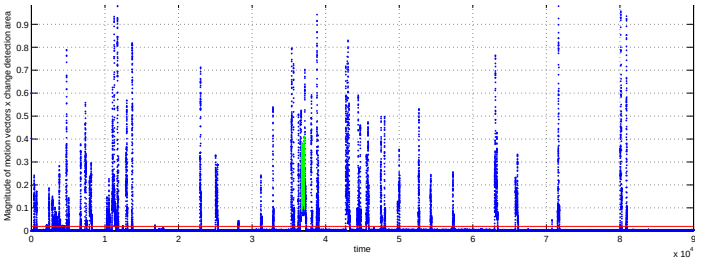


Fig. 8 Sample motion vector magnitudes in foreground regions. The magnitudes below the threshold line are due to noise whereas the highlighted patch indicates the interval occupied by the event.

as the reference plane. The motivation for using a common plane for assignment of score is to assign scores to region based on their significance in the site rather then in the image alone. Each region is assigned a *region score* $\gamma_k \in [0,1]$ where $\gamma_k \to 1$ represents the region of higher significance. The detections from the image plane are transferred to the ground plane. The image-plane to ground-plane (π) projection can be estimated by applying the homography matrix $H_{i\pi}$.

$$O_{ij}^{\pi}(t) = H_{i\pi} O_{ij}(t), \tag{2}$$

where $O_{ij}^{\pi}(t)$ is the ground-plane projection of the j^{th} object when observed from i^{th} camera. Each object is assigned a region score $\gamma_{ij}(t)$ based on its location in the scene at time t.

5 Event detection

We use a combination of low- and high-level features for event detection. The low-level features include motion vectors and activity detection (Section 4.1), whereas the high-level feature are based on the output of an object detector. These features include the size and the location of the target. In case of the airport dataset [22] this was a pedestrian detector [30], and for basketball and simulated pedestrian scenarios (Fig. 2(a) and 2(c), respectively) we use the output of the object detector and tracker described in Section 4.2.

The motion vectors are computed by applying block matching using different window sizes based on the camera perspective. We use rectangular blocks instead of square block as the target objects, i.e. pedestrians, form upright rectangular bounding boxes. The three different block size used were 2×4, 4×8 and 8×16 with a shift of 1 pixel and a search window of 14×14 pixels. Figure 8 shows the magnitudes of motion vectors. The peaks in the signal indicate activity intervals where there are some objects in the scene. Due to perspective, the object sizes vary across the scene and does the the magnitude of the associated motion vectors. This magnitude is normalized by dividing with the average magnitude, over non-event intervals of a square block of the scene (Fig. 9). The normalizing factor is further smoothed by applying a mean filter (see [26] for further details).

List of events specific to each dataset is generated and an event score, based on their significance is assigned to each event. In this work, we apply thresholds on object- and frame-level features to identify these events. However model-based approaches can be applied depending upon the event of interest [24]. To this end, for the basketball dataset, we consider the event of *attempt on basket* as a global event. For this we consider the region in the vicinity of the basket and if the overall magnitude of vectors is higher than a threshold, this is considered to be an *attempt on basket* (see Fig. 10). The second class of events for this

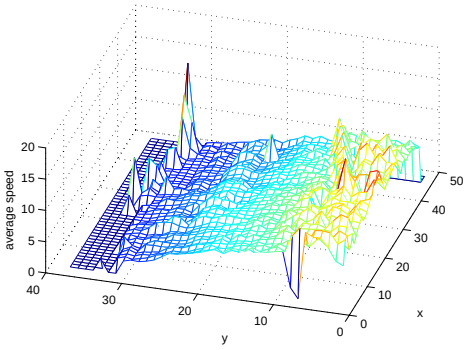


Fig. 9 Sample normalization factor to compensate for changes in perspective due to object size, computed for each 16×16 region of the image.

Fig. 10 An example frame for *attempt on basket* event.

data set is the *high activity event*. This event is directly related to the amount of motion vectors in the scene and the amount of activity $d_i(t)$ associated to the frame. The event considered for the pedestrian dataset was the event of an object being on the road (marked by green lines for visualization in Fig. 13) for a duration longer then β time instances (*pedestrian-on-road* event). For this we consider the location score $\gamma_{ij}(t)$ associated to each object O_{ij} being observed from camera C_i at time t. In these set of experiments we use a value of $\beta \geq 15$ frames. In the airport surveillance dataset we performed the detection of three events namely *person runs* (total magnitude of motion vectors normalized with average magnitude of motion vectors within a region), *elevator no entry* (when an object does not enter the elevator with elevator door open) and *opposing flow* (a person walking in a direction opposite to the allowed direction).

Assume that there are L_s possible events which can happen for a multi-camera setup and this set is represented as $\Gamma = \{\lambda_1, \ldots, \lambda_{L_s}\}$. Based on the importance of each event, it is assigned a score $\theta^l|_{l=1,\ldots,L_s}$. If a set $\lambda_i^l(t)_{l=1,\ldots,L_i} \in \Gamma$ of L_i events are detected in camera C_i at time t, the total event score $\Theta_i(t)$ for each camera at time t is given as $\Theta_i(t) = \sum_l^{L_i} \theta_i^l$, where θ_i^l for $l = 1, \ldots, L_i$ is the list of event scores seen by camera C_i at time t.

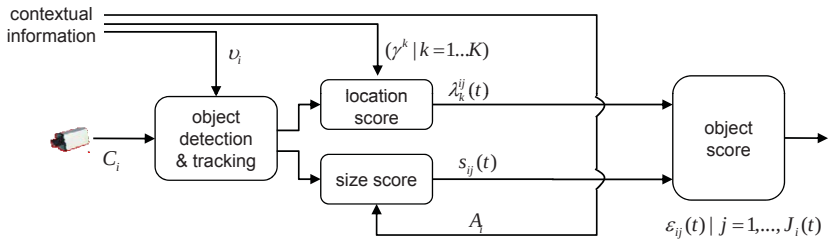


Fig. 11 Block diagram showing for the calculation of the object visibility score.

(a) (b) (c)

Fig. 12 Object score example: (a) object score in a region of lower interest. Increase in score due to (b) change in size (b) entered in the region of high interest.

(a) (b) (c) (d)

Fig. 13 Example scores for an object as it (a) approaches region of high interest, (b) is in the region of interest, (c) moves away from the camera, (d) just before it leaves the region of interest.

6 Content Scoring

The content of the scene is ranked in two steps: In the first step all objects are ranked independently in each camera view. In the next step a combination of frame-level features and object score is used to find the frame-level score for each camera.

6.1 Object Score

The object score $\varepsilon_{ij}(t)$ for each object O_{ij} is calculated by scaling the size feature associated with its location score

$$\varepsilon_{ij}(t) = s_{ij}(t)\gamma_{ij}(t). \tag{3}$$

The pipeline used for the extraction of object level features and assigning score to each object is highlighted in Fig. 11. This score is indicative of the importance of an object within the scene. Fig. 12(a-c) shows the development of the score of a single object as it moves from a region of low interest to a region of high interest (Fig. 17). The score takes values of 0.24

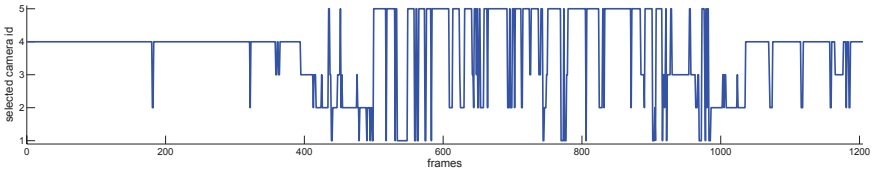


Fig. 14 Sample output of maximum-score-based view selection for the basketball scenario with 5 cameras as shown in Fig. 2.

(Fig. 12(a)), 0.26 (Fig. 12(b)) and 0.73 (Fig. 12(c)). The slight increase in object rank in Fig. 12(b) is due to the increase in the size of the detected player. A significant increase in the object score is observed (see Fig. 12(c)) when the object enters the region of higher significance. Similarly, Fig. 13 ((a)-(d)) shows the increase in the score of a target from 0.092 (Fig. 13(a)) to 0.799 (Fig. 13(b)) as the pedestrian steps on the road. This rise in the score of the object is due to the pedestrian stepping on the region of high interest. However as the target moves farther away from the camera (Fig. 13 (c),(d)) the object score further decreases to 0.304 and 0.273.

Information about objects within a camera view is provided by its local features. To evaluate the importance of the camera view itself, these local features need to be combined with the global features. This generates a frame level scoring at each time t which is discussed in the next section. The cumulative object score $E_i(t)$ at time t, for all $J_i(t)$ objects in the view of camera C_i, is calculated as $E_i(t) = \sum_{j=1}^{J_i(t)} \varepsilon_{ij}(t)$.

6.2 Frame scoring

The frame rank is computed at each time t using the *amount of activity*, *number of objects*, *scene-centric events* and the *accumulated object score* in the frame. The change $d_i(t) = \left|I_i^d(t)\right|$ observed in the frame at time t is used as a cue for the amount of activity. The activity level of the object in the near field of the camera is more as compared to the ones in the far field due to the perspective view. To cater for this we take into account the number of objects $J_i(t)$ in the view of a camera C_i. To this end a feature vector $\psi_i(t)$ is constructed as

$$\psi_i(t) = (J_i(t), d_i(t), E_i(t), \Theta_i(t)). \tag{4}$$

We model the score $\rho_i(t)$ for each camera C_i at each time t as a continuous multivariate distribution $\mathcal{N}(\mu_i, \Sigma_i, \psi_i(t))$ with mean μ_i and covariance Σ_i. Figure 14 shows an example output for the basketball scenario for the best-view selected using the maximum frame score based selection criteria: $\tilde{C}(t) = \text{argmax}_i\,[\rho_i(t)|i = 1 \ldots 5]$. It can be seen that there are frequent switches, some of which are of very short intervals. Adding constraints such as minimum viewing/scheduling period [2, 10] may cause loss of information. This problem can be solved by considering the knowledge of the previously selected view and the prior knowledge about the scene as discussed in the next section.

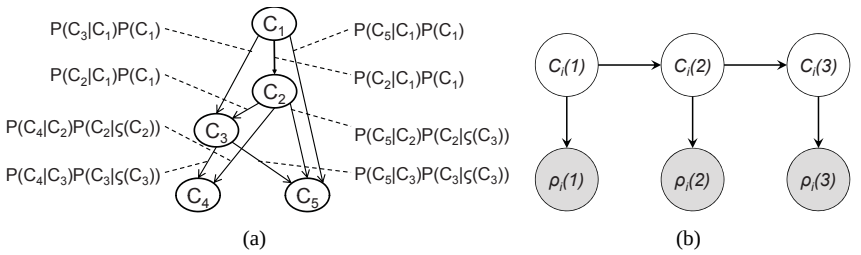


Fig. 15 Sample Bayes network (a) defining the adjacency between cameras where C_i is the i^{th} state and $\zeta(.)$ are the parent nodes for C_i; (b) DBN unrolled for $T = 3$.

7 View selection

Frequent switches (Fig. 14) between cameras are undesirable as they make the content of the video difficult to understand. These switches can be avoided by considering the past states and prior knowledge about the sensor network. This knowledge (the state priors $P(C_i)$ and the state transitions $P(\tilde{C}(t) = C_i|\tilde{C}(t-1) = C_j))$ is computed based on the activity in each camera and their placements. This state modeling can be done using Dynamic Bayesian Networks (DBNs), a generalized form of Hidden Markov Model (HMM), where both the hidden states and the observations can be defined as a set of (discrete or continuous) random variables. Furthermore, DBN are dynamic acyclic graphs (DAGs) where each node is connected to another node using a directed arc/edge. This feature of DBN makes it more suitable for use in view selection where switching between cameras is dependent on their placement in the network instead of in an arbitrary order. This constraint can be easily represented as a DAG which allows to convert the heuristics about the network structure into a probabilistic algorithm. A sample DAG is shown in Fig. 15(a) where each edge represents the probability of going from one stage to another. In DBN, the joint state is computed as

$$P(C_i(t)) = \prod_{l=t-T}^{t} \prod_{j=1}^{N_i^{\zeta}} P(C_j(l)|\zeta(C_j(l))), \tag{5}$$

where $\zeta(.)$ are the parent nodes of $C_i(t)$ and T is the number of time slices (number of past observations to take into account) and N_i^{ζ} is the number of parent nodes for camera C_i. For example, Fig. 15(b) shows an unrolled version of DBN for a sequence of time slices $T = 3$, the joint probability distribution is given as

$$P(C_i(1:T), \rho_i(1:T)) = P(C_i(1))P(\rho(1)|C_i(1))\prod_{t=2}^{T} P(C_i(t)|C_i(t-1))P(\rho_i(t)|C_i(t)). \tag{6}$$

Note that in Fig. 15(b) the observations are also nodes of the graph. Furthermore, each node only depends upon information up to current time and there is no dependency on future information, thus preserving the acyclic condition of the DBN. The effect of camera-camera transition probability $P(C_i(t)|C_i(t-1))$ is also evident from Fig. 15(b) and Eq. 6. The network structure consisting of $N+1$ nodes (an additional node for the absorption stage) using the adjacency matrix (Eq.(9)), where N is the number of cameras in the sensor network. This matrix is created using the camera configuration thus defining the possible state transitions (e.g Fig. 15(a)). Each element of the adjacency matrix provides the transition

probability of selecting a view given the current view. Formally, the probability of observing state C_i given the current state is C_j can be computed as

$$P(C_i(t)|C_j(t-1)) = P(C_i(t)|C_j(t-1))P(C_i(t)|\zeta(C_j(t-1))). \tag{7}$$

In this computation we assume the model to be first order Markov and the transition and observation functions are time-invariant i.e. $P(C_i(t)|C_j(1:t-1)) = P(C_i(t)|C_j(t-1))$. To facilitate transitions after the system is in the absorbing state, a auxiliary node is introduced which allows transition to the parent nodes. The scores ρ_i for each camera C_i are used as observation for the Bayesian network. The observation is an N-dimensional feature vector defined as $\bar{\rho}(t) = \{\rho_1(t), \cdots, \rho_N(t)\}$. The states are modeled using binomial distribution. The choice of binomial distribution is because in this distribution each trial results in exactly one of some fixed finite number k of possible outcomes, with probabilities $(p_1, \cdots, p_k)$ such that $\sum_{i=1}^{k} p_i = 1$, and there are n_{tr} independent trials. The parameter learning for each node is based on Expectation Maximization [14] algorithm. The training is performed for each state up to n iterations or until the change in log likelihood is less then a threshold $\eta = 10^{-100}$. The likelihood for each state is calculated by applying marginalization. The final camera selection $\tilde{C}(t)$ is then computed as

$$\tilde{C}(t) = \underset{i}{\operatorname{argmax}} \left[\Upsilon(\rho_i(t-T:t)|P(C_i(t)))P(C_i(t))\right], \tag{8}$$

where $\Upsilon(.)$ is assumed to be Gaussian and $P(C_i)$ is the prior on the state.

8 Results

To evaluate the performance of the proposed approach we show two different types of experiments. The first set of experiments regards the content scoring and camera selection. The second experiment focuses on the evaluation of the system in terms of selection of the best-view while minimizing the number of view switches. In both sets of experiments we demonstrate the effectiveness of the proposed approach by comparing it with manually generated ground-truth. The ground-truth was done manually by 11 non-professional users. Each user was asked to select a camera view at each time instant. Then the view selected by majority of the users was chosen as the best-view at that time instant.

8.1 Experimental setup

We demonstrate the performance of the proposed approach on three scenarios, namely a basketball game, an indoor airport surveillance and an outdoor scene. The basketball game (Fig. 2(a)) is monitored by 5 cameras with partially overlapping fields of view. The data consists of a total of 17960 frames, out of which 500 were used for training the DBN and the remaining were used for testing. Contextual information is composed of the expected size and orientation of the players in each camera view. The regions of the video outside the field are not considered, and any features of interest observed in these regions are ignored. The camera network configuration is encoded in the form of Bayes net as shown in Fig. 15(a). The indoor airport surveillance dataset was also acquired using 5 partially overlapping cameras (Fig. 2(b)). The section of data on which we demonstrate the proposed approach contains 180006 frames for each camera, out of which 1000 were used for training the DBN and remaining for testing the algorithm. Contextual information in this case is composed of

the normalized magnitude of the motion vectors after applying temporal smoothing for each camera view and information about the site. The outdoor scene is composed of synthetic data generated using [28] and consisting of a 4-view set-up with semi-overlapping cameras (see Fig. 2(c)). These video streams consist of 1816 frames per camera, out of which 100×4 were used for training the DBN and the remaining for testing. Contextual information in this case included the location of the road which was used to generate a *pedestrian-on-road* event. The adjacency matrices define the transition probabilities of going from one state to another (selecting a camera, given the selected camera at previous instant). The adjacency matrices for these datasets, A_B for basketball, A_A for airport surveillance, and A_H for the outdoor dataset, are defined in Eq. 9

$$A_B = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, A_A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, A_H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

In A_A (adjacency matrix for airport dataset) we do not enforce any camera transition order as we always want to select the view with the highest score. For each of these setups the location importance is assigned manually from the contextual information about the site. The normalization of motion vectors is performed over a 16×16 block of the scene.

8.2 Content Scoring and View Selection

Figure. 16 shows the score, the DBN output, the ground truth and the selected state by the proposed approach on all 5 cameras of the first 1250 frames for each camera of the basketball dataset. It can be seen that in many cases (Fig. 16(c), frames 775 to 800 and Fig. 16(d), frames 0 to 400), contrary to score, the probability of the state computed by DBN is close to 1 (i.e. has higher confidence). This results in a reduced number of switches between views compared to the maximum score based view selection. The large number of peaks in the result of camera 2 (Fig. 16(b)) is caused by this camera's field of view that covers most of the scene and hence often observes some activity. The activities in the far field generate the peaks in this graph, whereas consistent higher probability is observed for this camera when most players are in the near field of the camera. The probabilities generated by DBN for the remaining four cameras (Fig. 16(a,c-d)) are higher in certain intervals only.

Figure 17 shows sample results of the proposed approach on the same dataset. Camera 1, 3 and 5 have fewer number of object in the region of interest (row 2). Camera 2 and camera 4 have almost an equal number of objects. However, camera 4 has a much higher activity level. This results in a much higher probability for camera 4 to be selected compared to the other cameras, as shown in Fig. 16. In Fig. 17, row 3, camera 1 has objects in the far field, whereas camera 4 has no objects. Camera 2, 3 and 5 all seem good candidate for selection and therefore none of them has a significantly high score (dashed line in Fig. 16). In this case, camera 3 is selected due to the accumulated temporal information (smoothness added by DBN). In row 4, all cameras except camera 4 can be selected as best view as at frame 940 the players are dispersing after an attack. Here the selected camera is camera 1 again due to the previous state in the state estimation of the DBN. To better visualize the performance of the proposed approach the input videos along with the results can be found at http://www.elec.qmul.ac.uk/staffinfo/andrea/view-selection.html

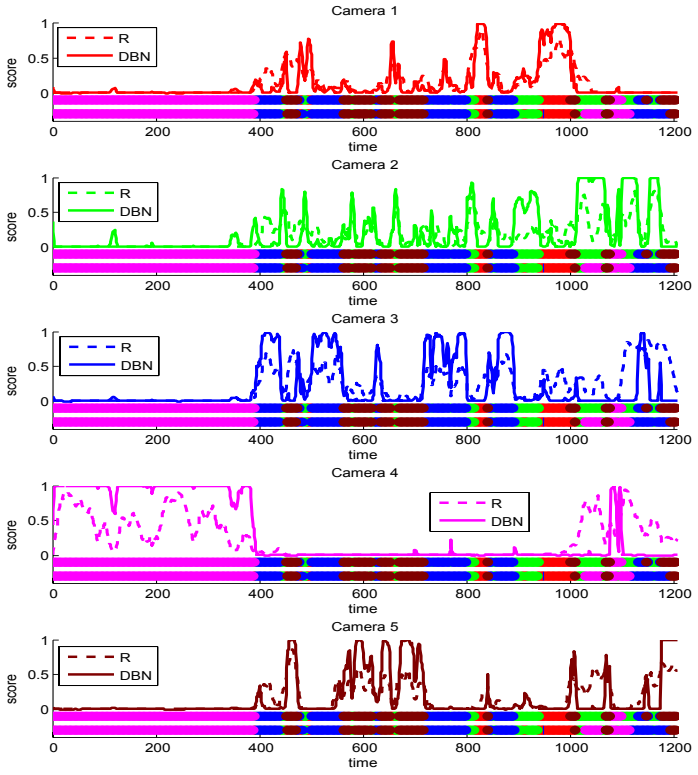


Fig. 16 Camera scores, DBN output, associated ground truth and generated output for the first 1205 frames of the Basketball data set (Ground truth key: red: camera 1, green: camera 2, blue: camera 3, magenta: camera 4, brown: camera 5) (Graph key: solid line: DBN output, dashed line: scores (R), GT: ground truth, AD: automated) ompared to R the DBN scores are closer to 1 and are consistent over several intervals which increase the reliability of selected view and reduce frequent short switches. The several high peaks in camera 2 are due to the fact that it observes the entire scene and hence there is always some activity.

Similarly, for the airport surveillance dataset (see Fig. 18), in row 2 camera 4 has no object, whereas there is activity in all the remaining cameras. Although the amount of change in camera 1 is high owing to a fast moving vehicle, however camera 2 is selected due to higher number of targets. However in row 3 camera 5 is selected, which is due to larger target size as compared to camera 2 which has more number of targets. In row 4 camera 5 is empty and camera 4 only contains 2 targets, however due to the target location (inside or close to the elevator), camera 4 is selected.

For the outdoor dataset (Fig. 19), camera 2 is selected due to a larger number of objects in its view (row 2) compared to the other views. In row 3, camera 3 and 4 have almost no objects and camera 1 and 2 observe the same two objects. However, camera 1 is selected due to its higher object scores (on the basis of the size of the objects). In row 4, camera 4 is selected, while ignoring other views. This is a result of the target being on the road, which is an area of higher interest.

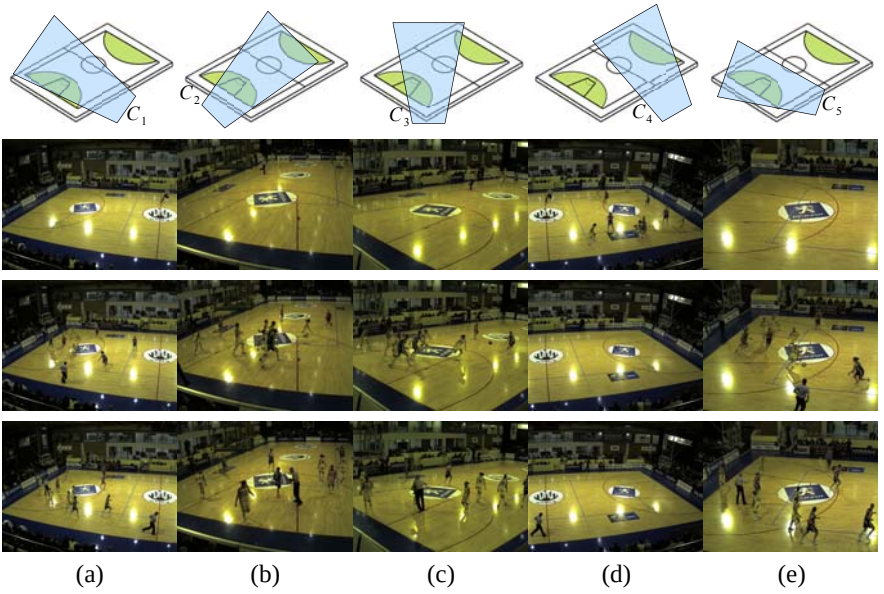


Fig. 17 Example frames from each camera and view selected by the proposed approach: (a) camera 1; (b) camera 2; (c) camera 3; (d) camera 4; and (e) camera 5. (Row 1) Layout of the scene showing camera field of view (blue) and regions of high interest (green). (Row 2) Frame 255 where camera 4 is selected; (Row 3) Frame 540 where camera 3 is selected and (Row 4) Frame 940 where camera 1 is selected.

8.3 Complexity

To compute the computational cost of the entire algorithm, we consider the three stages outlined in Fig. 1. Each stage further contains sub-processes as highlighted in Fig. 20. The stated percentage average times were calculated on an Intel 3.2 GHz Pentium dual core using non-optimized implementation. The detection and tracking was done using visual C++ and remaining modules were implemented using Matlab. It can be seen in Fig. 20 that the detection and tracking module contributes to 61.53% of the total computational cost. The frame ranking (5.95%) and the view selection (11.22%) on the other hand, takes only 17.15% of the total time. This shows that the bulk of the time (82.85%) is consumed in object detection and tracking and feature extraction (stage one of Fig. 1). This indicates that given an efficient implementation for detection, tracking and feature extraction the proposed view selection can be easily performed in short time duration.

8.4 Evaluation

To evaluate the effectiveness of the smoothing introduced by the proposed approach (DBN-BV) we compare it with a maximum score (MR) based approach. In this approach we introduce the selection interval τ and the decision is taken at the beginning of each selection interval. This results in reducing the number of switches as the change of view is only allowed after τ frames. Figure 21 shows the result where the number of switches reduces from 53 to 26 as τ increases from 1 to 20 when using MR. In case of DBN, the number of switches decreases from 24 to 20 only. This shows that by using DBN we can reduce the

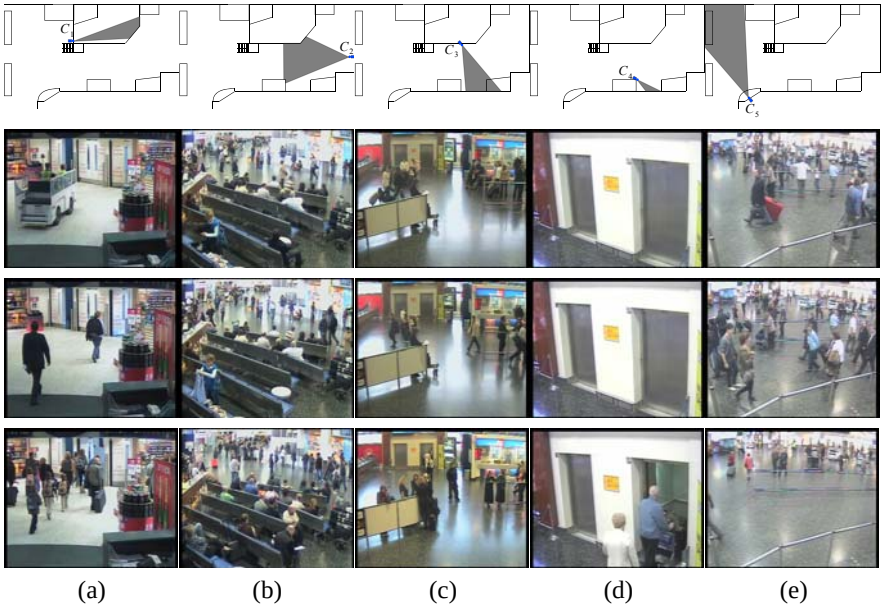


Fig. 18 Example frames from each camera and view selected by the proposed approach: (a) camera 1; (b) camera 2; (c) camera 3; (d) camera 4; and (e) camera 5. (Row 1) Layout of the scene showing camera field of view (blue). (Row 2) Frame 1740 where camera 2 is selected; (Row 3) Frame 2010 where camera 5 is selected and (Row 4) Frame 60680 where camera 4 is selected.

number of switches without introducing the additional parameter τ which may need to be adjusted based on the dynamics of the scene.

We also perform a subjective evaluation with a Turing Test on the sports dataset using 7 videos and 22 subjects (Fig. 22). Out of these 7 videos, 5 were generated using manual selection by different users ($U1-U5$), one video *(MR)* was generated by applying maximum score criteria on the overall frame score (shown in Fig. 14) and the last video was generated using the proposed approach *(DBN-BV)*. Each subject was asked to decide, for each video, *whether it is generated by manual selection or automatically*. It was observed that 73% of the subjects misidentified the video generated by the proposed approach as manually generated. Only 18% of the subject misidentified the MR video as manual, whereas 75% of the subjects were able to correctly identify the manually generated video.

9 Conclusions

We presented a best-view selection algorithm for multi-camera settings. The scoring of each camera view is based on the analysis of object and scene features. A multivariate Gaussian distribution model uses these features to assign scores to each view. To prevent frequent camera switching, camera selection is performed using a DBN with binomial distributions. We showed that using the DBN results in fewer short-term switches between cameras and demonstrated via subjective testing that this increases the likability of the generated video.

A further extension to this work is to use a multinomial distribution to model the continuous state space of an active camera. Also we would like to investigate the use of additional

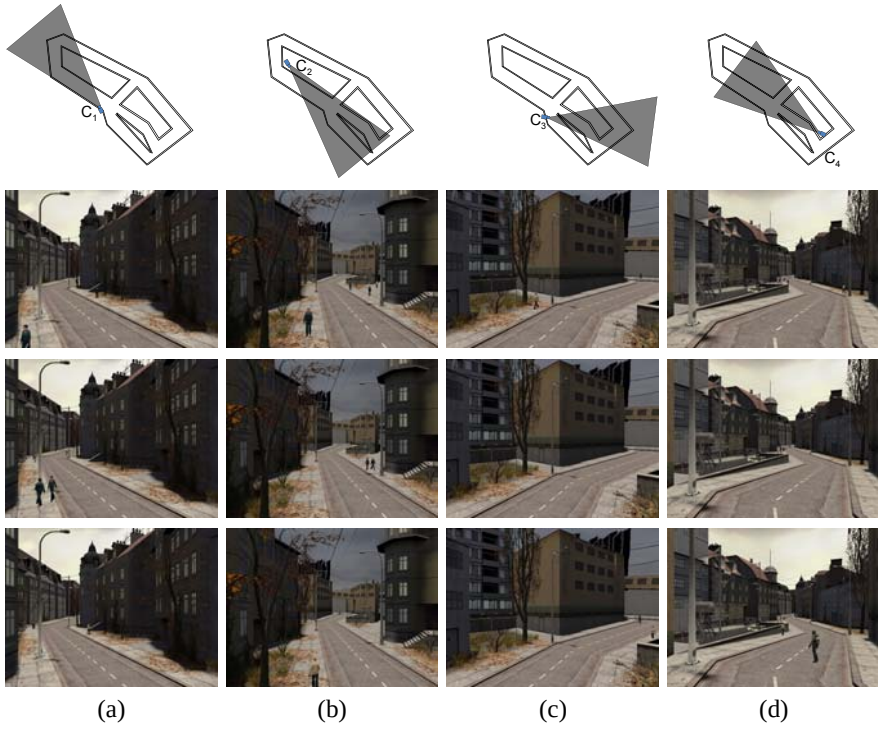


Fig. 19 Example frames from each camera and view selected by the proposed approach: (a) camera 1; (b) camera 2; (c) camera 3 and (e) camera 4. (Row 1) Layout of the scene showing camera field of view (blue). (Row 2) Frame 184 where camera 2 is selected; (Row 3) Frame 476 where camera 1 is selected and (Row 4) Frame 1627 where camera 4 is selected.

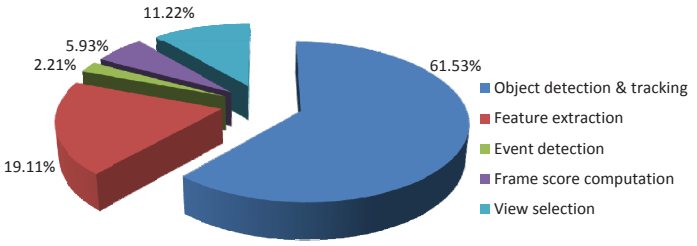


Fig. 20 Percentage average time for each module of the proposed approach.

features such as object tracking and motion models to enable the system to predict the next best view.

References

1. Batista J, Peixoto P, Araujo H (1998) Real-time active visual surveillance by integrating peripheral motion detection with foveated tracking. In: Proc. of IEEE Workshop on Visual Surveillance, pp 18–25

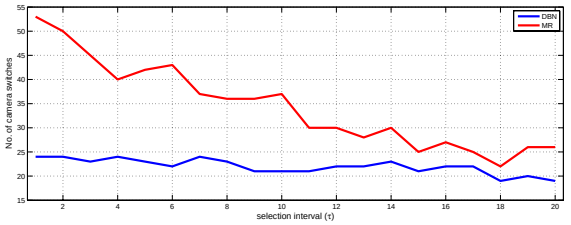


Fig. 21 Comparison of the proposed approach with a baseline (maximum score based selection) in terms of number of camera switches with the increase of the selection interval.

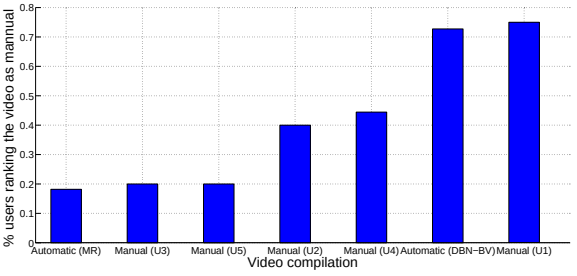


Fig. 22 Turing test carried out on the basketball video using 5 manual ($U1-U5$), one video using maximum score based criterion video *MR*, and one generated using the proposed approach *DBN-BV*.

2. Bimbo AD, Pernici F (2006) Towards on-line saccade planning for high-resolution image sensing. Pattern Recognition Letters 27(15):1826–1834
3. Chen X, Davis J (2008) An occlusion metric for selecting robust camera configurations. Machine Vision and Applications 19(4):217–222
4. Costello CJ, Diehl CP, Banerjee A, Fisher H (2004) Scheduling an active camera to observe people. In: Proc. of the ACM 2nd Int. Workshop on Video surveillance & sensor networks, pp 39–45
5. Daniyal F, Taj M, Cavallaro A (2008) Content-aware ranking of video segments. In: Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras, pp 1–9
6. Gilbert A, Bowden R (2006) Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In: Proc. of I9th European Conference on Computer Vision, Part II, pp 125–136
7. Goshorn R, Goshorn J, Goshorn D, Aghajan H (2007) Architecture for cluster-based automated surveillance network for detecting and tracking multiple persons. In: Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras
8. Greiffenhagen M, Ramesh V, Comaniciu D, Niemann H (2000) Statistical modeling and performance characterization of a real-time dual camera surveillance system. In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, pp 335–342
9. Gupta A, Mittal A, Davis LS (2007) Cost: An approach for camera selection and multi-object inference ordering in dynamic scenes. In: Proc. of IEEE Int. Conf. on Computer Vision, pp 1–8
10. Jiang H, Fels S, Little JJ (2008) Optimizing multiple object tracking and best view video synthesis. IEEE Trans on Multimedia 10(6):997–1012
11. Karlsson S, Taj M, Cavallaro A (2008) Detection and tracking of humans and faces. EURASIP Journal on Image and Video Processing 2008(1):1–9
12. Khan S, Shah M (2003) Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. IEEE Trans on Pattern Analysis and Machine Intelligence 25(10):1355–1360
13. Lien KC, Huang CL (2006) Multi-view-based cooperative tracking of multiple human objects in cluttered scenes. In: International Conference on Pattern Recognition, pp 1123–1126
14. Murphy K (2002) Dynamic bayesian networks: Representation, inference and learning. PhD thesis, Department of Computer Science, UC Berkeley

15. Park HS, Lim S, Min JK, Cho SB (2008) Optimal view selection and event retrieval in multi-camera office environment. In: Proc. of IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems, pp 106–110
16. Prince SJD, Elder JH, Hou Y, Sizinstev M (2005) Pre-attentive face detection for foveated wide-field surveillance. In: Proc. of IEEE Workshop on Application of Computer Vision, Vol. 1, pp 439–446
17. Qureshi FZ, Terzopoulos D (2005) Surveillance camera scheduling: a virtual vision approach. In: Proc. of the ACM Int. Workshop on Video surveillance & sensor networks, pp 131–140
18. Qureshi FZ, Terzopoulos D (2007) Surveillance in virtual reality: System design and multi-camera control. In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition
19. Rezaeian M (2007) Sensor scheduling for optimal observability using estimation entropy. In: IEEE Int. Workshop on Pervasive Computing and Communications, pp 307–312
20. Senior A, Hampapur A, Lu M (2005) Acquiring multi-scale images by pan-tilt-zoom control and automatic multi-camera calibration. In: Proc. of IEEE Workshop on Application of Computer Vision, Vol. 1, pp 433–438
21. Shen C, Zhang C, Fels S (2007) A multi-camera surveillance system that estimates quality-of-view measurement. In: Proc. of IEEE Int. Conf. on Image Processing, pp 193–196
22. Smeaton AF, Over P, Kraaij W (2006) Evaluation campaigns and trecvid. In: Proc. ACM Int. Workshop on Multimedia Information Retrieval, pp 321–330
23. Snidaro L, Niu R, Varshney P, Foresti G (2003) Automatic camera selection and fusion for outdoor surveillance under changing weather conditions. In: Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance, pp 364–369
24. Taj M, Cavallaro A (2008) Object and scene-centric activity detection using state occupancy duration modeling. In: Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance
25. Taj M, Maggio E, Cavallaro A (2006) Multi-feature graph-based object tracking. In: Proc. of Classification of Events, Activities and Relationships (CLEAR) Workshop, pp 190–199
26. Taj M, Daniyal F, Cavallaro A (2008) Event analysis on trecvid 2008 london gatwick dataset. In: Online Proc. of TREC Video Retrieval Workshop
27. Tarabanis K, Tsai R, Allen P (1995) The MVP sensor planning system for robotic vision tasks. IEEE Trans on Robotics and Automation 11(1):72–85
28. Taylor G, Chosak A, Brewer P (2007) OVVV: Using virtual worlds to design and evaluate surveillance systems. In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition
29. Tessens L, Morbee M, Lee H, Philips W, Aghajan H (2008) Principal view determination for camera selection in distributed smart camera networks. In: Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras, pp 1–10
30. Viola P, Jones M, Snow D (2003) Detecting pedestrians using patterns of motion and appearance. In: Proc. of IEEE Int. Conf. on Computer Vision, pp 734–741
31. Zhou X, Collins RT, Kanade T, Metes P (2003) A master-slave system to acquire biometric imagery of humans at distance. In: Proc. of ACM SIGMM Int. Workshop on Video surveillance, pp 113–120