Applied Intelligence (2022) 52:3954–3975
https://doi.org/10.1007/s10489-021-02636-4

Interactive visualization-based surveillance video synopsis

K. Namitha[1] · **Athi Narayanan**[2] · **M. Geetha**[1]

Accepted: 21 June 2021 / Published online: 14 July 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Video synopsis is an effective technique for the efficient analysis of long videos in a short time. To generate a compact video, multiple tracks of moving objects, which we call as tubes are displayed simultaneously by rearranging them along the time axis. Contemporaneous video synopsis approaches focus on collision avoidance, or preservation of chronological order among tubes. However, generation of an adaptive personalized user-oriented synopsis video congruent to users' preferences has yet not been thoroughly experimented. This paper propounds a framework for personalized visualization of synopsis video, integrating pertinent object attributes such as color, type, size, speed, travel path and direction towards generation of synopsis video for precise inference of user needs. The framework motivates users to interactively define queries for creation of the targeted synopsis. User queries are classified into visual-queries, temporal-queries, spatial-queries, and spatio-temporal queries concomitant with the visual and spatio-temporal attributes. Tubes relevant to a user-query are selected, and grouped according to original behavioral interactions followed by their rearrangement, to generate synopsis video with fewer false collisions. To evaluate the proffered technique, two evaluation metrics are proposed and extensive experiments of publicly available surveillance videos are conducted. The experimental results demonstrate the propriety and usability of the newer approach.

Keywords Visualization · Surveillance · User-interaction · Object attributes · Video synopsis

1 Introduction

With the unsatiated user interests in multimedia applications, deployment of surveillance cameras has become commonplace around the globe. However, efficient browsing and review of massive recorded video data for detection and selection of events of interests, pose an exigent, tedious task. Surveillance videos can be effectively used by reducing their duration significantly. Towards this end, several video shortening approaches like video abstraction [1], condensation [2], summarization [3, 4] and video synopsis [5–7] have been proposed. In recent times video synopsis has been a popular technique, which generates a compact representation of a long surveillance video, by the simultaneous display of multiple events that may have occurred at different times in the original video. Besides condensing the video content, video synopsis serves as an index to the original video, for detailed tracking and analysis of events, in diverse surveillance applications. A sequence of spatio-temporal positions of a moving object, termed as a tube, is the fundamental unit in video synopsis methods. The terms "objects" and "tubes" are used interchangeably throughout this paper.

Video surveillance systems that utilize video synopsis, support users with an expert system to analyze videos for operational decision-making in various sectors of society such as law enforcement, crime investigation, incident analysis, critical infrastructure security, transportation, health-care and education, property and retail security, among others. In the past decade, researchers have proposed numerous approaches for the generation of a synopsis video [8] focusing either on reducing collisions among moving

✉ K. Namitha
namitha.amrita@gmail.com

Athi Narayanan
mail2athi@gmail.com

M. Geetha
geetham@am.amrita.edu

[1] Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India

[2] Kimball Electronics (India), Technopark Campus, Thiruvananthapuram, India

objects [9–12], optimal rearrangement of events [13–16], or varying camera topology [17, 18]. Rav-Acha et al. [5] proposed object-based video synopsis in 2006. Later, Pritch et al. [6, 7] extended the method of [5] into a framework in which the moving objects of a video are re-arranged, along the temporal domain, achieving a high condensation ratio. Following the classical framework in [7], Nie et al. [19] proposed an optimization method that relocates objects in both temporal and spatial domains, by expanding the moving space of objects in a video. Shifting the tubes temporally or spatially may ensue in their collisions with each other in the synopsis video. Li et al. [9] proposed an approach to reduce collision in synopsis video by scaling down the size of colliding objects. Another method to minimize collisions was proposed by Nie et al. [12] in which the speed of objects are varied together with the size reduction. However, an ideal synopsis video should be one that is in tune with users' preferences while pursuing a tradeoff between commonly used standards [20, 23] such as preserving as many events as possible, reducing collisions and maintaining temporal order between moving objects.

Although video synopsis has emerged as a pioneering technology in the field of video analysis, there are tricky challenges like generating a synopsis video in align with the preferences of users. Therefore, there is a need for video synopsis approaches that rely on personalized and customized tube abstraction methods.

During the review of a surveillance video, a viewer perceives its contents with more meaningful visual and motion cues, such as object color, type, size, speed, tracks, and interactions. Hence, we need to consider these object features, in order to infer the users' needs, to facilitate human perception, for the generation of user-specific synopsis of a video. Toward this end, this paper proposes a visualization framework for video synopsis, which enables users to interactively select tubes of interest in a video, group them based on behavioral interactions, and optimally rearrange the tubes to generate customized video synopsis.

The proposed visualization framework enables users like security personnel to review hours of surveillance videos within minutes, inspecting objects of interest by their attributes, and accelerating target-based filtering based on various search combinations. The interactive user interface of proposed framework assists end-users to analyze videos for fast and operational decision-making in various sectors of society like public and private surveillance, law enforcement, transportation, retail stores and the like. Some among the numerous applications of the proposed framework are the following: supports investigation or surveillance team to pinpoint objects by type, speed, interaction, and more for tracking and inspecting suspects; identifies activity hot-spots in a retail store, on the highway, or in a parking area, quantifies the type of visitors or vehicles, and discover their movement patterns using filter by direction, regions of interest (ROI), traffic flow, and object type. Moreover, the visualization framework enables end-users of various public and private sectors to generate custom synopsis videos of people, vehicles, or other objects of interest by adjusting the tolerance levels to refine search.

The proffered framework supports users with a user-friendly interface to interactively submit queries, making use of visual and motion cues. For example, a user may specify object attributes like color, size, or select a particular region of interest in a video, to view events occurred there [desired location], or to view objects that traversed through a specified path of interest. Hence, we have categorized the user-defined queries into visual-queries, temporal-queries and spatial-queries, pertinent to the fixed visual object attributes, spatial and temporal characteristics of the video. Methods are presented hereunder to group the related tubes for the preservation of behavioral relationships among them, and determination of the optimal placements of the tube groups in a synopsis video. Our contributions and focus of this paper are in the modules of interactive tube-selection, tube-grouping, and tube-group rearrangement. Hence, for the pre-processing/post-processing steps of synopsis generation like tracking and stitching of tubes, we have employed state-of-the-art methods.

To summarize, the main contributions of this paper are as follows:

- A visualization framework for the generation of user-oriented video synopsis is proposed. Users are afforded a user-friendly interface, to interactively select customized combinations of analytic features for the abstraction of tubes, thus, generating targeted synopses.
- Based on several attributes of an object such as size, shape, path, color, behavioral interactions, and motion, we propose four main classifications of user-defined queries, namely, i) visual, ii) temporal, iii) spatial and iv) spatio-temporal.
- A personalized tube-grouping procedure is presented to discover the relationships among tubes by grouping the interacting tubes that are relevant to a user's query. Further, the optimal positions for these tube groups in synopsis video are determined by a space-time cube representation-based method that aims to minimize collisions between tubes.
- Two novel evaluation metrics: *False Overlapping Area (FOA)*, *Non-Preserved Interactions (NPI)* are proposed and extensive experiments are conducted to validate the effectiveness of the proposed approach using surveillance videos.

The remainder of this paper is organized as follows. Section 2 presents selected related works on video synopsis and visualization systems. Section 3 describes the proposed

approach in detail. In Section 4, the proposed evaluation metrics are described. Experimental results are presented in Section 5. We conclude the paper in Section 6.

2 Related work

This section reviews the related work of video synopsis, notably user-oriented synopsis video generations.

2.1 Object-based video synopsis

In contrast to the aforementioned visualization systems, video synopsis approaches are object-based that benefit from shifting moving objects along the temporal axis. Rav-Acha et al. [5] and Pritch et al. [6, 7] introduced object-based synopsis methods, which focus on the rearrangement of objects in the time domain. They termed the sequence of occurrences of the same object, across multiple video frames, as a tube. These tubes are then shifted along the time axis, which effectively reduce spatial and temporal redundancies. To avert the collisions that may occur due to the rearrangement of tubes, Nie et al. [19] presented a method to shift objects, spatially and temporally by expanding the moving space of objects in the video. Zhu et al. [21] augmented and accelerated the pioneering work in [7] by formulating the tube rearrangement as a step-wise optimization problem, and implementing it using Graphic Processing Unit (GPU). To reduce collisions between objects, Li et al. [9] proposed an optimization method that scales down the size of colliding objects. He et al. [10] reformulated tube rearrangement for online video synopsis using a potential collision graph that computes collision relationship between tubes in advance. To reduce the collision artifacts in offline video synopsis, He et al. [11] determines the potential collisions that may happen in the synopsis video by formulating it as a graph coloring problem. An approach for generating synopsis videos by grouping tracklets of moving objects, utilizing their spatio-temporal relations is presented in [22]. Li et al. [23] proposed a framework that discovers the relationships between moving objects and can be applied to generate synopsis for scenes with crowdedness.

Recently, Ra et al. [13] proposed an algorithm to accelerate the tube rearrangement optimization by making use of fast Fourier transform and parallel processing. To reduce the optimization time during tube rearrangement in video synopsis, Ghatak et al. [15] presented a hybrid energy minimization scheme using Simulated Annealing and Teaching Learning-based optimization methods. Ruan et al. [14] proposed a dynamic graph coloring problem for the tube rearrangement in online video synopsis, where the tube relationships are modeled using a dynamic graph. A framework to reduce collisions between objects by changing the speed, scaling the size and shifting of objects, together with a Metropolis sampling algorithm is presented in [12]. Moussa et al. [16] proposed a particle swarm optimization-based approach for solving energy minimization in video synopsis to minimize collisions between tubes and maintain their temporal order. Furthermore, some video synopsis approaches focus on multi-camera topology, activity clustering, and target-based that integrate user-inputs.

2.1.1 User-oriented video synopsis methods

Unlike numerous studies investigating the generation of synopsis videos, only a few have dealt with the facility to incorporate user-preferences, though they limit to specific types of user queries. Pritch et al. [6] proposed a two-phase method to generate synopsis of videos from webcam or surveillance cameras. Display of objects of interest in the synopsis were selected from the user specified time period. However, user-query defines only the desired temporal interval, while simultaneous display of multiple random activities during that period may create a confusing synopsis video. Subsequently, similar activities are clustered using appearance features and motion features in a synopsis method proposed by Pritch et al. [20]. Thus, the users are enabled to view synopsis of preferred object types using appearance-based clusters and objects with similar motion path using motion-based clusters. Nevertheless, objects with similar trajectories but varying speed may be assigned to different clusters.

To overcome the aforementioned problem in [20], a synopsis approach was proposed by Chou et al. [24] that used longest common subsequence algorithm to group trajectories with similar motion and dissimilar speed/length in the same cluster. In this method, the starting and ending locations of all trajectories are grouped to obtain the number of coherent-events. Furthermore, users are enabled to specify the number of event groups, and view synopsis of multiple events with similar trajectories. To generate a synopsis of abnormal objects in a video, Lin et al. [25] presented an abnormality detection approach that utilizes a patch-based method and blob optimization process. In [26], an event-based video synopsis method was proposed that generates synopses of similar kinematic events by clustering trajectories. Another clustering-based video synopsis method was proposed by Ahmed et al. [27] in which synopsis videos are generated based on a few user-queries. Their method adapts users' synopsis preferences, based on three object classes, similar trajectories, objects with similar starting or ending locations and combinations of these interests. All the aforementioned user-oriented approaches generate targeted synopsis videos limited to specific applications such as a synopsis of events

within a temporal period of interest, synopsis based on three object types (i.e. car, bike, pedestrian), a few trajectory-based synopses including similar activities or kinematic events, abnormal objects, coherent events and movements between key-regions.

In contrast to the above mentioned approaches, the method proposed in this paper focuses on user-oriented synopsis generation, by utilizing most of the visual, temporal and spatial attributes of an object, to define a user-query. The proposed visualization framework employs an interface to interactively define queries by selecting arbitrary combinations of different object attributes such as size, type, color, travel path, and direction along with ROI, traffic flow and interacting objects in a video. Furthermore, we use a recursive personalized grouping method to preserve interactions by identifying and grouping the related tubes that are relevant to a query. The optimal temporal locations for these tube groups are then determined by a space-time cube representation-based approach.

3 Proposed methodology

This section explains the proposed framework of personalized user-oriented video synopsis generation. Initially, an input video is preprocessed by detection and tracking multiple moving objects to generate tubes. Then, the extracted tubes are classified using deep learning methods. When a user submits a query via graphical user interface (GUI), the subset of tubes relevant to the user queries are selectively retrieved from the whole set of extracted tubes. To preserve the behavioral interactions among those selected tubes, the query-relevant tubes are grouped, utilizing their spatio-temporal proximity. In the end, the tube groups are stitched to the generated background based on the temporal locations determined by a cube representation-based tube group rearrangement approach. Figure 1 presents the proposed framework. For reference, Table 1 summarizes the definitions of key notations used in this paper.

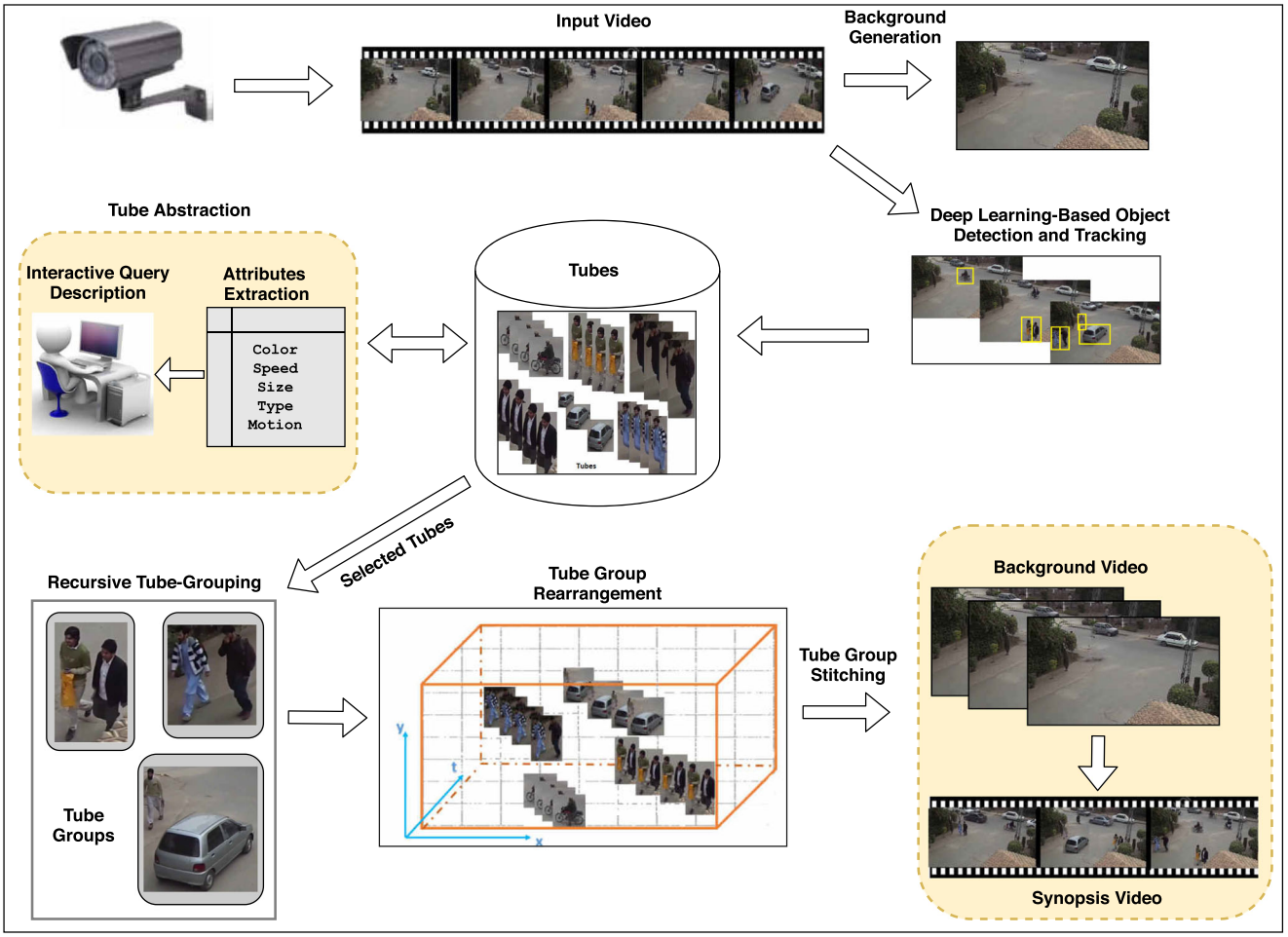

Fig. 1 The framework of the proposed user-oriented video synopsis approach

Table 1 Notation definitions

Notation	Definition
T	a tube
ξ	trajectory information of a tube
φ	set of visual attributes of a tube
N	number of tubes in the input video
Γ	whole set of N tubes
Q	number of tubes relevant to a user-query
Ψ	abstracted set of Q tubes
G	a tube group
M	number of tube groups
F	total number of frames in a synopsis video
V	spatio-temporal volume representation of synopsis video
P_i	the i^{th} temporal location of synopsis video

3.1 Background estimation and tube extraction

The primary steps in the preprocessing of video are estimation of video background, followed by extraction of tubes. Surveillance videos usually have static backgrounds. In this work, a temporal median is applied over the neighboring frames of each video frame to estimate the background. Considering slight variations in the illumination, background is computed for a duration of every 60 seconds of the video.

Given a video, extraction of the tubes is performed by detection and tracking [28] of objects in motion. Many classical algorithms like Gaussian Mixture Model (GMM), and Aggregated Channel Features (ACF) can be used for detection and extraction of foreground. State-of-the-art object detectors focus mainly on detecting small objects [29], detecting vehicles in traffic systems [30], sequential image processing-based [31], fusion-based [32], localizing moving regions [33] along with others. In recent years, deep learning methods [34] have been outperforming classical techniques. In the work propounded in this paper, a recent and popular deep learning-based detector YOLOv3 (You Look Only Once v3) [35], applied for detection of objects. Making detections of objects at 3 different scales, the upgraded version of YOLO [36] called YOLOv3, detects even small objects with a speed of 30 frames per second along with high detection accuracy. Taking a video frame as an input, YOLOv3 outputs bounding boxes, object class and detection confidence for each object detected in the frame. For example, each detection is associated with:

$$d = [c_o, b_x, b_y, b_w, b_h, pc] \tag{1}$$

where (b_x, b_y) represents the center of bounding box, b_w and b_h denotes its width and height, respectively. c_o denotes the class of detected object and pc represents the confidence in percentage with which the detection is performed. These detection results are used as input for further tracking of multiple objects in video.

The next step of tube extraction is object tracking, which associates the detections corresponding to an object in one frame to the same object across other video frames. This paper adopted a simple online and real-time tracking (SORT) algorithm, integrated with a deep association metric (Deep-SORT) [37] for tracking multiple objects. Incorporation of appearance information in the tracking methods of SORT, Deep-SORT, track objects even in the presence of prolonged occlusions.

Detection results from YOLOv3 are conveyed to the Deep-SORT algorithm as input. Then, a Hungarian algorithm is used for associating the object detections to existing tracks. To accomplish that, each object is defined with a state vector as

$$[b_x, b_y, b_r, b_h, \hat{b_x}, \hat{b_y}, \hat{b_r}, \hat{b_h}] \tag{2}$$

where (b_x, b_y) are the center coordinates of bounding box, b_r and b_h are the aspect ratio and height of the bounding box, respectively. $\hat{b_x}, \hat{b_y}, \hat{b_r}, \hat{b_h}$ represent the respective velocities. A Kalman filter is used on every bounding box to predict the aforementioned state of an object. Deep-SORT introduced an appearance feature, which is extracted for each detected bounding box using a pretrained convolutional neural network (CNN). Object tracking generates 2-D trajectories of objects which can be integrated with object attributes for further visualization and query processing.

3.2 Tube abstraction for content analysis

Tube abstraction is a critical process given the vast amount of tubes generated for each video. Since prior knowledge of user's query is unavailable, tube extraction step produces tubes for each moving object in a video, and not all tubes are necessarily associated with interesting events. Therefore, we take the analytic attributes of an object, such as color, type, size, speed, motion, and interaction into consideration while selecting the tubes relevant to a user's query, termed as tube abstraction. To exploit the visual and motion characteristics of video, we classify user queries into visual-queries, temporal-queries, spatial-queries and, spatio-temporal queries. The proposed framework facilitates a compact and query-driven interface (see Fig. 2) that displays original video and object attributes to design a query. This gives the user the ability to choose attributes by themselves, or in conjunction with other attributes based on the context and requirements for interactively generating synopsis video. A few examples of such queries can be "red vehicles", "big white vehicles moving at high speed", "pedestrians with blue colored dress moving towards the right part of the video". Figure 3

Fig. 2 User interface for the proposed interactive visualization and query creation by utilizing visual and motion attributes displayed on the screen. The user can select the required video and temporal period to shorten the video. **A** Here, the users are supported to select attributes of interest like object's type, size, speed, color, and direction of travel to define a query. **B** The user can draw ROI (to be included or excluded) and travel path on the video frame, click on *interacting objects* to filter objects accordingly. Object trajectory or area of activity can be selected to view the traffic flow in the video. **C** Execute user-defined query and generate synopsis video

illustrates an example of querying and customized synopsis generation for each of the four query types, using different attributes of an object. Consequently, we define each extracted tube as a set

$$T = \{\xi, \varphi\} \tag{3}$$

where ξ is the trajectory information and φ is the visual attribute set of tube.

$$\xi = (x_1, y_1, w_1, h_1), ..., (x_n, y_n, w_n, h_n) \tag{4}$$

where (x, y) represents object location in each frame with (w, h) as width and height of the object respectively.

$$\varphi = \varphi_1, \varphi_2, ..., \varphi_i \tag{5}$$

where φ_i $(1 \leq i \leq n)$ represents an attribute such as object color, size etc. The set of all N extracted tubes in a video can be represented as

$$\Gamma = \{T_1, T_2, ...T_N\} \tag{6}$$

Further, Γ is abstracted to create its subset of relevant tubes Ψ, in response to a user-defined query:

$$\Psi = \{T_1, T_2, ...T_Q\},\ Q \leq N \tag{7}$$

where Q is the number of tubes relevant to a user-query.

3.3 Visual-queries

These are user-defined queries created by utilizing the fixed visual attributes of a tube like color, type and size. For example, "pedestrians that traveled through the white pathway of the given video scene". In this work, we propose the following strategies for extracting various visual attributes.

3.3.1 Object color

Color of an object is one among the primary cues that draws human attention in a video as presented in [38–40] like a feature for content retrieval. In this work, we extract five dominant colors for representing each tube. The first step of feature extraction separates each input video frame into Red (R), Green (G) and Blue (B) component images. We define a standard Red Green Blue (RGB) color palette $S = s_i$, $1 \leq i \leq 255$, where s_i represents the RGB value of three channels color. Then, we compute the similarity between

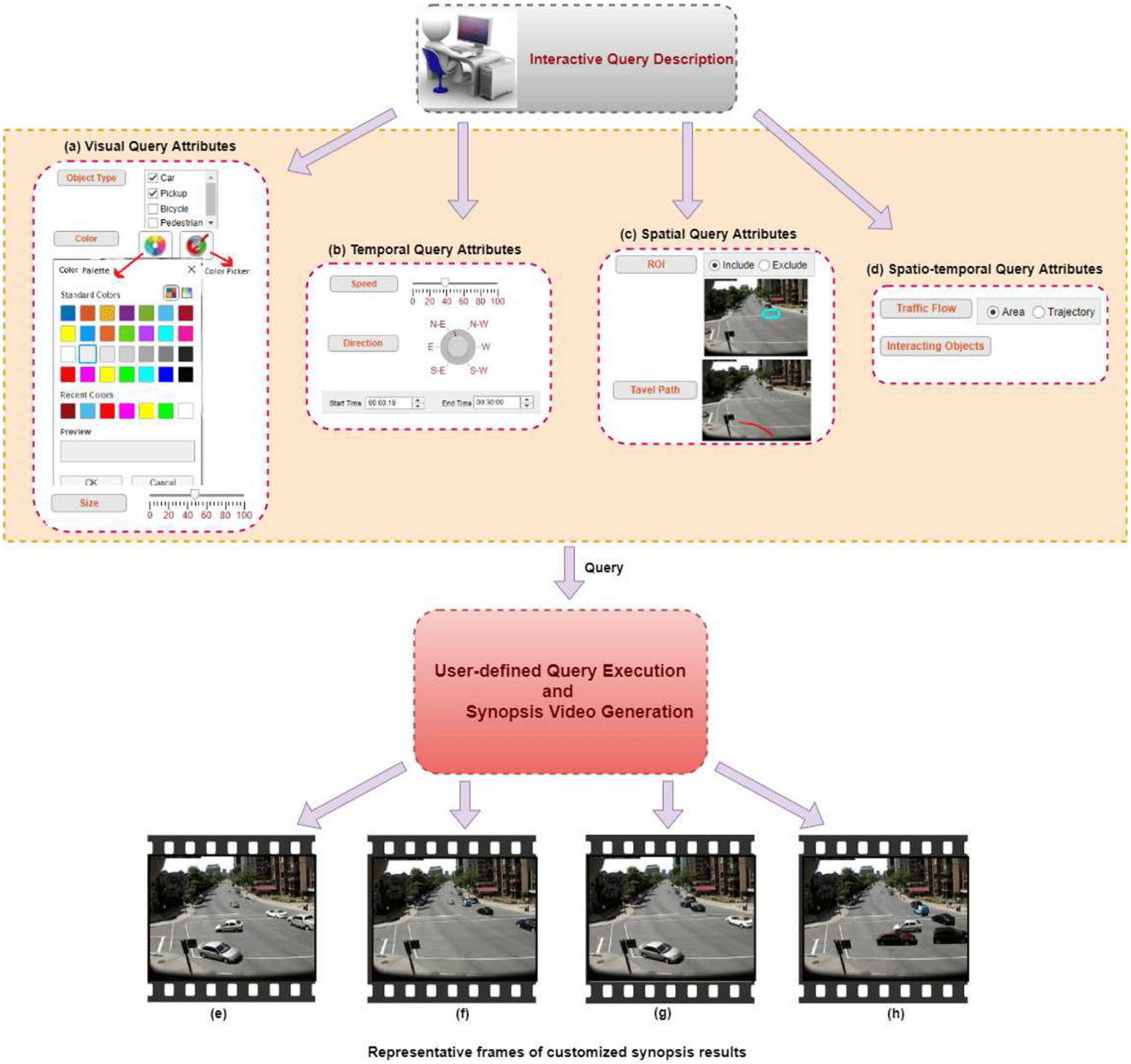

Fig. 3 Work-flow of interactive query creation and the corresponding personalized synopsis video generation. The user utilizes combinations of **a** visual, **b** temporal, **c** spatial, and **d** spatio-temporal attributes of an object to define a query. One representative frame from each synopsis result corresponding to four classes of illustrated queries are displayed in the figure. **e** Sample synopsis frame in response to (a) for medium-sized and off-white colored cars and pickups. **f** Sample synopsis frame in response to (b) for objects occurring in video between 00:00:10 and 00:30:00 time, moving in the North direction at an average speed. **g** Sample synopsis frame in response to (c) for objects traveling through the user-sketched path and ROI. **h** Sample synopsis frame in response to (d) for all interacting objects in the input video

pixels in each detected bounding box in a frame and S. The Euclidean distance similarity measure between the RGB value of each non-background pixel and each of the RGB color $s_i \in S$ is calculated. Each pixel of an object votes for its most representative color in S that has minimum distance with it. Next, a histogram of votes on the palette colors is created over the frame occurrences belonging to a single tube. Finally, we extract the first five colors as the dominant colors corresponding to the bins which receives the majority of votes.

3.3.2 Object size

Object size is often useful to differentiate large-sized vehicles such as trucks, buses from average (cars, SUVs) and small-sized vehicles (bike), and pedestrians from vehicle [39, 41]. Most traditional surveillance systems fail to utilize size of an object as a visual cue for producing user-defined video summaries, since multiple small adjacent objects are merged during motion segmentation and considered as a single object. In this work, the deep learning-based object detection and tracking method provides a precise bounding box (BB) for each moving object in a frame. Consequently, we get dimensions of the bounding box in pixels, with $Area(BB)$ yielding the approximate size. We extract three aggregate size measures for each tube: $min(Area(BB))$, $max(Area(BB))$ and $mean(Area(BB))$ over the entire track of an object. Taking all occurrences of an object across its appearing frames into account, alleviates the scaling issue.

3.3.3 Object type

In several surveillance applications, determining the type of object is critical [38]. Object-category based queries supports a user to view only those objects in any desired classes such as 'pedestrians', 'cars' in a synopsis. We have used YOLOv3 framework for detection and labeling of moving objects. YOLO divides each video frame into several $n \times n$ grids on which classification and localization are applied. Each grid predicts bounding boxes and class conditional probabilities that represents the likeness of detected object to belong to a particular class. YOLOv3 uses a much deeper classification network Darknet-53 with 53 convolutional layers for feature extraction, compared to the previous versions. Avoiding the softmax function used in YOLO, YOLOv3 uses logistic classifiers for multi-label classification, which has reduced the computation complexity. Softmax function assumes classes to be mutually exclusive, where one object cannot belong to multiple classes. However, this assumption may not work with classes like 'Pedestrian' and 'Woman' in a dataset. With logistic regression, YOLOv3 assigns classes with scores greater than a threshold to the object inside bounding box.

3.4 Temporal-queries

A temporal query utilizes the motion attributes of objects such as direction of travel, speed, and temporal ROI in a video. For example, "synopsis of past two hours of surveillance", "cars moving in the east-west direction", "high speed vehicles".

3.4.1 Activity speed

An object moving with high speed or low speed in a video captures human attention. Hence, users may prefer to view synopsis displaying objects with varying speeds such as fast, slow or average speed. To determine speed attribute, we use the statistics of detected bounding boxes in each video frame using method similar to [41]. A bounding box from YOLO detector is represented as

$$BB = \{b_x, b_y, b_w, b_h\} \tag{8}$$

where (b_x, b_y) represents the center of bounding box, b_w and b_h denotes its width and height, respectively. Once we have the location of an object at each frame, it is straightforward to calculate its speed. We measure the speed of an object using the change in its displacement with respect to each frame. The displacement is computed as the difference between centroids of the bounding boxes of an object in consecutive frames. Thus, three main speed variations, namely fast, mean and slow speed are extracted across bounding boxes of each tube.

3.4.2 Activity direction

The information about direction of motion [40] is implicitly present in the object trajectories ξ, obtained from tracker. We compute motion direction for each tube utilizing the centroid coordinates $b_{(x,y)}$, of the bounding boxes through its frames of occurrences. The direction of object can be obtained by calculating the difference between centroids of consecutive frames. However, the difference will be negligible unless the object is traveling at high speed. To empathize on larger object movements, we compute the centroid difference as

$$\Delta_x = b_x^i - b_x^k \tag{9}$$

$$\Delta_y = b_y^i - b_y^k \tag{10}$$

$$if|\Delta_x| > 0, \text{direction} = \text{'East'}; else, \text{direction} = \text{'West'} \tag{11}$$

$$if|\Delta_y| > 0, \text{direction} = \text{'South'}; else, \text{direction} = \text{'North'} \tag{12}$$

where b^i is the centroid of a tube T in i^{th} frame and b^k is centroid of T at k frames behind i^{th} frame. Δ_x and Δ_y are changes in the centroid locations along x and y axes, respectively. For a pixel-distance threshold p, if $abs(\Delta) > p$, then we define the motion direction. If $sign(\Delta_x)$ is positive, then the object direction is towards 'East (Right)'. The object will be moving towards 'West (Left)', when $sign(\Delta_x)$ is negative (see (10)). Similarly, the direction of motion is considered towards 'South (Down)', when $sign(\Delta_y)$ is positive and towards 'North (Up)' if negative (see Eq. (11)). However, if there is a significant difference in both x and y axes, then we define it as

object movement in directions such as 'South-East' (SE) or 'North-West' (NW).

3.4.3 Occurrence-time of events

Each tube extracted in the preprocessing phase is associated with a start-time t^s and end-time t^e, as given in (13). These time stamps are utilized when a user prefer to view synopsis of events occurred within the temporal ROI specified by them.

$$T = (t^s, t^e) \tag{13}$$

In this work, we pay attention to time-sensitive nature of events - generate synopsis with events occurred i) before a user specified time, ii) after a user specified time, iii) during a user-defined time period.

3.5 Spatial-queries

For a given video, spatial queries specify a spatial ROI utilizing the location coordinates of tubes. The user interface in the proposed framework enables users to select the spatial region in a scene by drawing on a frame with a mouse. For example, "cars that travel through the given spatial region of the road", where the ROI is specified in spatial coordinates.

3.5.1 Activity path

Tube trajectories extracted by object tracker are characterized by a sequence spatial location with source and destination points represented as

$$\xi = \{(x_s, y_s), ..., (x_d, y_d)\} \tag{14}$$

The proposed visualization framework enables users to select an activity path in the video either by interactively sketching the motion path as shown in Fig. 4a and b, or by specifying the source and destination spatial coordinates of the path. The motion path specifies a spatial ROI, thereby enabling objects moving in that path alone to be displayed in the synopsis video. However, retrieving trajectories that exactly match the user-defined motion path may not always be feasible. Therefore, instead of a single trajectory, we retrieve trajectories of all objects moving within a distance threshold ϵ from the sketched activity path.

3.5.2 Activity region of interest (ROI)

With this attribute, the proposed framework generates video synopsis either by including or excluding tubes that are active within the regions of interest. In this work, the users are allowed to interactively select a spatial ROI within a video frame. Any polygonal shape such as a rectangle, square, or circle can be used to represent a ROI. The

Fig. 4 Sample spatial query **a** Activity path of interest (shown in red color) drawn on a video frame by a user. **b** Sample frame from the corresponding synopsis video. **c** ROI (see the rectangle) marked to include tubes in the synopsis video. **d** Sample frame from the corresponding result

proposed framework employs a rectangular shape to define a ROI. Only those tubes whose trajectories intersect with the ROI are filtered to generate synopsis video. In an 'inclusive' mode, only the objects movements within the ROI are taken into account. An 'exclusive' mode selects all trajectories other than those within the ROI for synopsis generation. Figure 4c and d illustrates an example of including ROI in the synopsis result.

3.6 Spatio-temporal queries

These are the user queries that utilizes both spatial and temporal characteristics of a video. An example is when a user prefers to view synopsis of highly active areas such as "most selling products in a supermarket" or synopsis of "all interacting tubes" in spatio-temporal domains.

3.6.1 Behavioral interaction

Conventional video synopsis methods tend to minimize tube collisions or preserve the temporal order among tubes. Spatial collision between tubes is a kind of interaction between them [42]. Spatial tube-collisions that are present in the input video are true collisions. Collisions which does not exist in the original video, but exist in the synopsis video are false collisions. Preservation of tube-interactions in a synopsis video is also an important aspect to be reckoned with, which is unaccounted for in the extant synopsis approaches. In a video sequence, an interaction may represent a conversation between people, accidents, fights, theft or personal attack. Hence, the proffered framework enables users to create a query for the exclusive display of interacting objects, in the synopsis.

We have proposed a method to discover the interacting tubes in a video, utilizing their spatio-temporal proximity and group them together to preserve the interactions. A recursive grouping approach determines the related tubes for each tube T_i. Any two tubes T_i and T_j are assigned to the same group, if they are temporally and spatially closer to each other. The spatio-temporal proximity is computed as

$$D_{ST} = \begin{cases} exp(\dfrac{-min_{f \in T_i \cap T_j}\{dst_f(T_i, T_j)\}}{avg(\omega_{T_i}, \omega_{T_j})}), & \text{if } T_i \cap T_j \neq \Phi \\ \infty, & \text{otherwise} \end{cases} \tag{15}$$

where $dst_f(T_i, T_j)$ represents the Euclidean distance between the centroid coordinates of tubes T_i and T_j at every commonly shared frame f. ω_{T_i} and ω_{T_j} are the areas of tubes T_i and T_j, respectively. The interacting tube groups can be determined using Algorithm 1. Let G_i is a group and tube $T_i \in G_i$. T_j will be assigned to the same group of T_i, when $min(dst_f(T_i, T_j)) \leq D_t$ and $T_i \cap T_j \neq \Phi$, where D_t is a maximum spatial distance for interaction grouping, determined empirically in our experiments. $T_i \cap T_j$ represents the temporal intersection between T_i and T_j. Further, the above procedure is recursively performed to determine the associated tubes of T_j. If a tube T_k is not interacting with any other tubes, then, a new group is created and T_k is added to this new group. The aforementioned recursive tube-grouping method is repeated for each extracted tube until all the tubes are assigned with a group label. Thus, the set of relevant tubes Ψ can be represented in terms of interaction groups as

$$\Psi = \{G_1, G_2 ... G_M\}, M \leq Q \tag{16}$$

where M is the number of tube groups relevant to a user-query.

Algorithm 1 Recursive grouping algorithm.

Input: Set of relevant tubes to a query $\Psi = \{T_1, T_2, ...T_Q\}$
Output: Set Ψ with tube group labels, $\forall\ T_i \in \Psi$
Initialization: $\delta = \Phi$
for each tube T_i in Ψ **do**
  if $T_i \notin \delta$ **then**
    Create a new tube group G_i
    $T_i.tubegroup = G_i$
    $\delta = \delta \cup T_i$
    $\mathcal{V}ISIT\ (T_i)$
  end if
end for
function $\mathcal{V}ISIT\ \{T_i\}$
$T_j.visited = false, \forall\ T_j \notin \delta$
for each frame f in which T_i appears **do**
  for each tube $T_{j(j \neq i)}$ in f **do**
    if $T_j \notin \delta$ and $T_j.visited = false$ **then**
      $T_j.visited = true$
      Compute $D_{ST}(T_i, T_j)$
      if $D_{ST} < D_t$ **then**
        $T_j.tubegroup = G_i$
        $\delta = \delta \cup T_j$
        $\mathcal{V}ISIT\ (T_j)$
      end if
    end if
  end for
end for
end function

Advantages Figure 5 depicts the advantages of the proposed tube-grouping approach. Figure 5 exhibits the preservation of interactions between tubes by the proposed approach, regardless of their temporal order in comparison

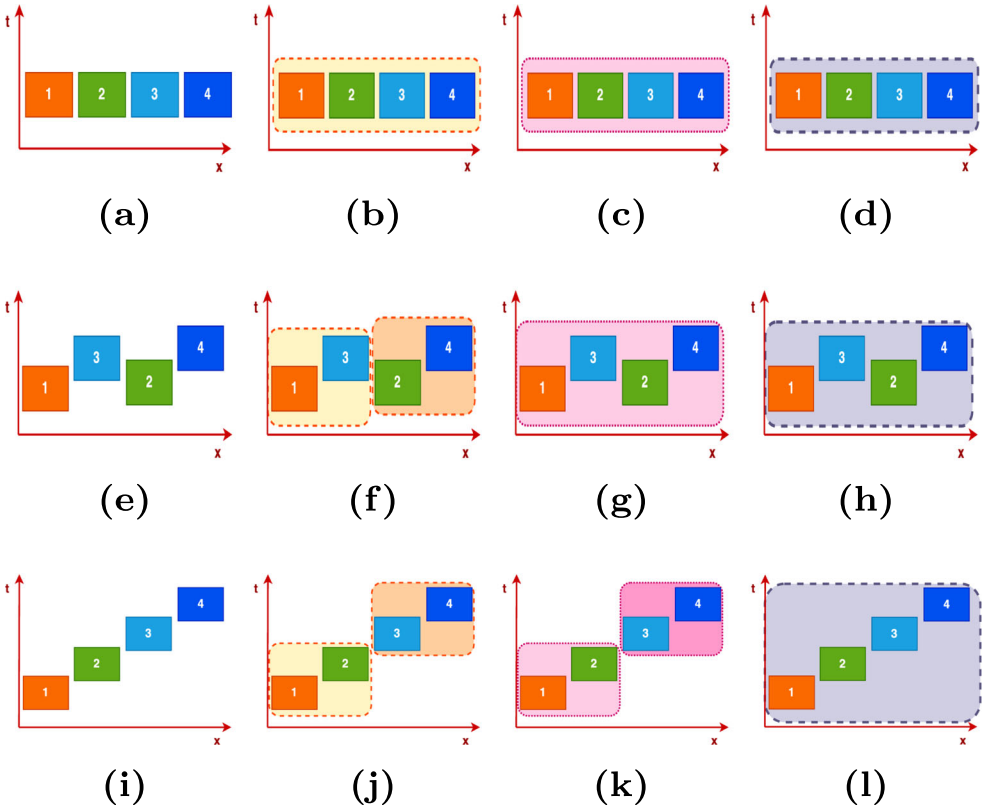

Fig. 5 The first column shows representative scenarios with tube interactions. The second, third and fourth columns illustrate grouping of tubes by [22, 23] and proposed tube-grouping approach, respectively

with the tube aggregation methods of [22] and [23]. Figure 5 represents different scenarios that have tubes with strong spatio-temporal relationships. Ahmed et al. [22] and Li et al. [23] have proposed methods that process each tube sequentially and bind the associated tubes together. In Fig. 5a, four tubes with same start-times and high spatial proximities are depicted. The grouping methods in [22, 23] as well as our proposed grouping method associates the related tubes 1 to 4 into an undivided group as presented in Fig. 5b, c, and d, respectively. Another scenario shown in Fig. 5e illustrates the similar appearing times of tubes 1,2,3, and 4 in a different sequence. The group-partition algorithm in [23] is unidirectional based on the sequential order of tube appearances. Accordingly, as shown in Fig. 5f, algorithm in [23] aggregates tubes 1, 3 into one group, and tubes 2, 4 into another group. However, both the grouping method in [22] and proposed recursive grouping method binds all interacting tubes (1 to 4) into a single group as presented in Fig. 5g and h, respectively. Figure 5i represents another tube interaction scenario. In view of the fact that the spatial distance between tubes 1 and 3 are high, the group-partition algorithm in [23] partitions the interacting set of tubes 1 to 4 into a group of 1 and 2, and another group of 3 and 4 as shown in Fig. 5j. Considering that both the temporal as well as the spatial distance between tubes 1 and 3, and tubes 1 and 4 are high, the grouping method in [22] splits the interacting tube group of 1 to 4 into groups of 1 and 2, and 3 and 4 as shown in Fig. 5k. However, the recursive approach of the proposed grouping method preserves the mutual interactions by aggregating tubes 1,2,3 and 4 into a single tube-group as shown in Fig. 5l.

3.6.2 Traffic flow

In surveillance videos that have highly redundant information, highlighting the scenes with greatest activity is particularly relevant. For example, the video monitoring person-in-charge at a retail shop may prefer to watch the product sections where most people visit, whereas a library-in-charge monitor may like to know which books have more check-outs (readers). In this work, we support users to quickly visualize the traffic observed for a specified time period by the generation of a heat map.

3.6.3 Composite attributes

To generate queries like “show large vehicles that traveled from east to west at a high speed” or “show white cars moved through an user-defined ROI at a specified time”, the users may have to select multiple attributes mentioned in the above sections. The proposed framework enables users to combine two or more attributes into a single query for the display of objects of interest, in the synopsis video.

3.7 Video synopsis generation

Upon creation and execution of user-defined queries, the relevant tubes (Ψ) are filtered from the whole set (Γ). To identify and preserve the true tube interactions in Ψ, if any, we apply the aforementioned recursive tube-grouping method to create tube groups. Next, we have to find optimal positions for these groups in the synopsis such that the false collisions among tube groups are minimized.

3.7.1 Tube group arrangement

We have proposed a space-time cube representation-based tube group placement method to optimally arrange groups, $\Psi \subseteq \Gamma$ in the synopsis. A set of tube groups relevant to a query:

$$\Psi = \{G_1, G_2...G_M\} \tag{17}$$

In this placement method, we consider a video as space-time volume $V(x, y, f)$, created using 3D cubes. (x, y) represents the spatial coordinates of a pixel at frame f, where $x = 1, .., W$; $y = 1, ..H$; $f = 1, .., F$. (W, H) represents the frame size and F denotes the total frames initialized for synopsis video. When a tube group is positioned at a temporal location in V, it may extend across several cubes spatially. There is a set of allowable temporal locations for each tube group. Positioning of a group elsewhere may exceed the synopsis length. Therefore, our objective is to determine an optimal start-time for each $G_i \in \Psi$ so that several tubes passing through the same cube in V at the same time is minimized. To record the tube groups and corresponding start-time labels with which a cube gets covered, we initialize each cube using a (tube group G_i, position P_i) pair matrix of size $M \times F$.

The placement method is comprised of two stages: cube coverage and voting. During stage I, we discover the cubes that are covered by the placement of each group at each of its possible locations. The matrix cell (G_i, P_i) of a cube C_k is set as '1', when positioning of G_i at start location P_i covers C_k given that G_i is not allocated with any start-time label. In the next stage, each cube C_k in V votes to (G_i, P_i) pairs if G_i can cover C_k when positioned at P_i. C_k distributes votes equally among all such covering pairs. Similarly, each (G_i, P_i) may receive votes from multiple cubes which gets covered by this pair.

During stage II, cubes that have already been covered due to the final placement of some groups still participate in further voting by inducing a penalty vote. Penalty voting reduces false collisions by minimizing the sharing of same cube by multiple (G_i, P_i) pairs. Finally, the (G_i, P_i) pair that receives maximum vote is selected for the placement in synopsis since such a solution ensures minimal sharing of space-time cubes. To maintain interactions between tubes that are identified by tube-grouping method, the relative time interval between start-times of tubes in a group is preserved, while positioning the group as a whole in the synopsis. The voting stage is repeated until all tubes groups are allocated with a start-time label. The detailed process is described in Algorithm 2.

Algorithm 2 Tube group rearrangement algorithm.

Input: Set of tube groups $\Psi = \{G_1, G_2...G_M\}$
Output: Set of start-time labels for tube groups, $\zeta = \{l_1, l_2, ..l_M\}$; Set of tube groups A, assigned with start-time labels
Initialization:
$\zeta \leftarrow \Phi$
Tube group-position pair matrix $Z_{m \times f} \leftarrow 0$, $\forall\, m \in \{1, ..M\}$, $f \in \{1, ..F\}$
$C_k \leftarrow Z_{m \times f}$, $\forall$ cube $C_k \in V$
$Vote(G_i, P_j) \leftarrow 0$, $\forall\, i \in \{1, ..M\}$, $j \in \{1, ..F\}$
for each tube group G_i in $\{\Psi - A\}$ **do**
    $max = F - length(G_i) + 1$
    for each possible position $\{P_i\}_{i=1}^{max}$ of G_i **do**
        for each cube C_k in V **do**
            if C_k is covered by G_i placed at P_i **then**
                $C_k\{Z(G_i, P_i)\} \leftarrow 1$
            end if
        end for
    end for
end for
while $A! = \Psi$ **do**
    for each cube C_k in V **do**
        Retrieve all (G_i, P_i) pairs which can cover C_k, where $G_i \notin A$
        if C_k is not covered by any $G_i \in A$ **then**
            $\alpha = \frac{1}{\#\{G_i, P_i\} pairs}$
            $Vote(G_i, P_i) \leftarrow Vote(G_i, P_i) + \alpha$
        else
            β = number of times C_k was previously covered
            $Vote(G_i, P_i) \leftarrow Vote(G_i, P_i) - \beta$
        end if
    end for
    $(G_i, P_i) \leftarrow \arg\max_{(G_i, P_i), G_i \notin A} Vote$
    $l_i = P_i$
    $\zeta = \zeta \cup l_i$
    $A = A \cup G_i$
end while

Advantages In earlier methods of video synopsis [7, 9, 12, 15, 19, 21], the tube rearrangement is formulated as an energy minimization problem [43, 44] and solved by minimizing energy functions. Minimization of energy functions

require iterative computation of pairwise tube activity, collision and temporal costs. In addition to redundant costs computation, the aforementioned optimization processes does not guarantee to preserve tube interactions. However, the proposed cube representation-based placement method aims to preserve the relative tube interactions by maintaining the true collisions and minimizing false tube collisions.

3.7.2 Tube group stitching

In this final step, the synopsis video is generated by stitching the tube groups onto the estimated background. A background video of synopsis video length is created for stitching the objects. As mentioned in Section 3.1, a background frame is created for each video frame using a temporal median over its neighboring frames. To generate a background video, a uniform temporal sampling is applied to these background images. Then, each tube group is stitched to the background video at optimal locations to generate synopsis video. Poisson image editing [45] is employed to stitch the tube groups into the background video, which is widely used in many video synopsis methods [6, 7, 9, 19].

4 Proposed evaluation metrics

Though there are no unified standards to measure the quality of synopsis videos, most of the state-of-the-art methods refer to three main standards: incorporate all activities, reduce the collision among objects, and maintain the temporal order of objects as far as possible. On the basis of these standards, video synopsis methods are generally evaluated using the following metrics [46]: frame compact ratio (CR), frame condensation ratio (FR), overlap ratio (OR), temporal disorder ratio (TD), time consumption, and visual quality. CR measures the degree of object density in each frame. Higher the CR, more compact is the synopsis video. FR measures the ratio of the number of frames in synopsis to the input video. A higher condensation of video is indicated by a smaller FR. OR computes the collision degree of tubes, it should be a smaller value. TD measures the number of tubes that are temporally disordered. A greater violation of temporal order results in a higher TD. Time consumption measures the execution time taken by a method to generate a synopsis video, evaluated mostly in seconds. Apart from the aforementioned quantitative performance metrics, visual quality is a qualitative metric to measure the visual pleasantness of synopsis results. Subjective feedbacks from users are employed by some synopsis methods [19] to compare the results.

As presented in [14], two main factors should be considered during the generation of synopsis videos: reducing collisions, and preserving interactions between tubes when rearranging them. To preserve strong spatio-temporal interactions, true collisions also have to be maintained as such in the synopsis video. In addition, false tube collisions need to be reduced in the synopsis, concerning the first key factor. However, none of the aforementioned conventional metrics consider false collisions or original interactions that are not preserved in synopsis, during the evaluation of video synopsis methods.

Therefore, we propose two metrics to measure false tube collisions and tube interactions of original video that are not preserved in the synopsis: False Overlapping Area (FOA) and Non-Preserved Interactions (NPI).

1) *False Overlapping Area (FOA)*: The existing metrics such as OR and collisions (C) [9], which are used to measure overlapping of objects consider the total collisions between tubes that incorporate both true and false collisions in the synopsis video. Whereas the proposed FOA metric takes only false collisions into account since original collisions in the synopsis need not be considered as newly generated during tube rearrangement. The original definition of OR from [11] is given as

$$OR = \frac{1}{w.h.T_s} \sum_{t=1}^{T_s} \sum_{x=1}^{w} \sum_{y=1}^{h} 1 \quad \{\text{if } p(x, y, t) \in \text{ the collision foreground}\} \tag{18}$$

where the collision foreground represents the overlapping area of tubes in the synopsis video, T_s is the synopsis length, $p(x, y, t)$ denotes a pixel at t^{th} frame of synopsis, w and h are the width and height of frame, respectively. The collision (C) metric in [9] is defined as the sum of all overlapping areas in the synopsis video. The proposed metric FOA indicates the false collision degree of tubes in the synopsis video. *FOA* between tubes a and b is computed as:

$$FOA(a, b) = \begin{cases} OA_s(a, b) - OA_o(a, b), & \text{if } t_a^s - t_b^s = t_a^{\hat{s}} - t_b^{\hat{s}} \\ OA_s(a, b), & \text{otherwise} \end{cases} \tag{19}$$

where $OA_o(a, b)$ denotes the sum of overlapping areas between a and b tubes in the original video and $OA_s(a, b)$ denotes the sum of overlapping areas in the synopsis. t^s and $t^{\hat{s}}$ denote the starting times of tubes in the original and synopsis video, respectively. Smaller the FOA, fewer the synopsis false collisions.

2) *Non-Preserved Interactions (NPI)*: NPI measures the degree to which the original interactions between tubes is

destroyed or not preserved in the synopsis video. NPI is defined as follows:

$$NPI = \frac{N_{IV}}{N_{IO}} \tag{20}$$

where N_{IV} is the number of tube pairs where interaction is violated in synopsis video and N_{IO} is the total number of interacting tube pairs in the original video. The count of original tube interactions are determined using the recursive grouping algorithm. The number of original interactions preserved in synopsis are computed by comparing the relative intervals between start-times of related tubes in original and synopsis video. A smaller NPI indicates lesser degree of modified true tube interactions in synopsis. A synopsis video with every true interactions maintained as such in the original video will lead to a NPI score of 0.

5 Experiments and results

We have conducted several experiments to evaluate the performance of the proposed approach. All experiments were conducted in MATLAB R2019a on an Intel Core i7-7560U CPU running at 3.80 GHz processor with 32 GB memory.

5.1 Dataset

To carry out the experiments, we have used 8 publicly available surveillance videos with diverse scenes and object-interactions. The experimental videos were carefully selected that depict diverse, real-world surveillance scenarios, such as busy intersection road-traffic, subway station, crossroads, urban traffic with vehicles and pedestrians, atrium with moving people and streets with multi-person interactions. The characteristics of these videos are summarized in Table 2.

5.2 Evaluation metrics

The response accuracy on user's query is evaluated with respect to three quantitative metrics: Precision, Recall and F-score. In addition to the proposed metrics: false overlapping area (FOA), and non-preserved interactions (NPI), the performance evaluation of the proposed approach in comparison to the state-of-the-art-methods is carried out based on six conventional metrics [46] such as frame compact ratio (CR), overlap ratio (OR), temporal disorder (TD), running time (RT), visual quality, and subjective evaluation. The proposed metrics and first four conventional metrics are employed for quantitative evaluations, whereas the visual quality and subjective feedbacks are used for qualitative evaluation. The frame condensation ratio (FR) was not compared as same length was set for all methods, in order to make a fair comparison.

5.3 Parameter Analysis

To effectively determine the value of parameter D_t in the recursive grouping method, experiments are conducted on 8 test videos. D_t is the upper bound of distance between two tubes within which an interaction is defined. Figure 6 shows the performance curve of NPI metric versus the change of D_t. It can be seen from Fig. 6 that the NPI score gradually increases beyond a certain value of D_t, indicating a higher number of non-preserved interactions in the synopsis video. Similarly, a lower value of D_t will cause the grouping of interacting tubes into different groups. Therefore we set D_t to 50 in the later experiments.

5.4 Quantitative results

5.4.1 Evaluation of tube abstraction

We evaluate the efficacy of different query classes with varied user-defined object attributes in terms of quantitative evaluation metrics like Precision, Recall and F-score. The

Table 2 Characteristics of test videos: resolution, total number of frames, frames per second (fps), total number of tubes

Video ID	Video title	Resolution	Number of frames	fps	Number of tubes
V1	M-30 [47]	800 × 480	7520	25	253
V2	Urban1 [47]	600 × 360	23435	25	226
V3	Town-Centre [28]	1920 × 1080	7500	25	231
V4	Sherbrooke [48]	800 × 600	1001	7	20
V5	i-Lids [49]	480 × 360	4470	30	120
V6	Car-Traffic [19]	610 × 480	4710	30	15
V7	ThreePastShop [50]	384 × 288	1650	25	9
V8	Atrium [48]	800 × 600	4540	7	52

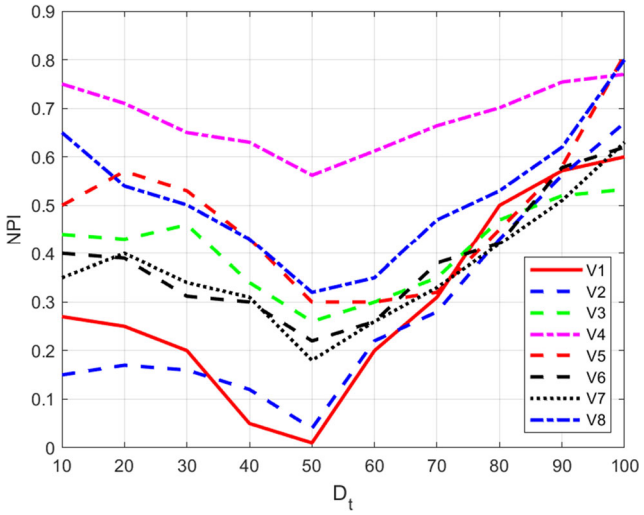

Fig. 6 Performance of NPI influenced by parameter D_t on 8 different videos

filtered subset of tubes, Ψ, in response to various visual, temporal and spatial queries, were compared against the annotated ground truths. We computed the Precision, Recall and F-scores for 10 queries each under different query classes on test videos. The average of these performance measures is presented in Table 3. Since F-score provides a balance between Precision and Recall, it is usually used as the main criterion for accuracy evaluation. From the results in Table 3, we can see that the responses generated by the proposed approach are satisfactory with a highest F-score as 0.97 and lowest as 0.75. In the view of Precision and Recall, the proposed method achieves a high to low score in the range of 1 to 0.77 and 1 to 0.60, respectively.

As mentioned in Section 2.1.1, a few synopsis methods [20, 24–27] generate targeted synopsis, albeit they address only one or two kinds of queries. The query types proposed in [24, 26] and [27] are compared against our proposed query classes in terms of response accuracy on user's queries, as depicted in Table 4. The results are average accuracy on 8 test videos. The query classes in the proposed approach exhibits higher average accuracy values compared to its corresponding similar query types in [24, 26] and [27].

Table 4 Comparison of response accuracy on different user-defined queries

Query class	Attribute	Method	Average accuracy
Visual	Type	[27]	0.89
		Our	0.92
Spatial	Path	[24]	0.81
		[27]	0.83
		Our	0.88
Temporal	Direction	[26]	0.85
		Our	0.89
Temporal	Speed	[26]	0.93
		Our	0.96

Figure 7 presents the response of proposed approach for user's query, "traffic flow in video V8". Figure 7a, b, and c illustrates sample video frames from the time period of interest (00:02:20 (hh:mm:ss) to 00:04:40 (hh:mm:ss)) of video. The corresponding motion trajectories are displayed in Fig. 7d. Figure 7e presents the heat map showing highly active and inactive regions for the entire duration of video.

5.4.2 Comparison with the State-of-the-art methods

To evaluate the performance of proposed query-based video synopsis generation approach, we compare synopsis videos generated using the proposed method and those using four state-of-the-art methods in video synopsis. We mainly compare with a classical approach proposed in [7], a method of scaling down the size of colliding objects to minimize collisions [9], a query-based method for traffic monitoring [27], and an object-based approach for reducing collisions [16]. To provide a fair comparison between these disparate query-based and non-query-based methods, we generate synopsis videos by the proposed method and aforementioned approaches [7, 9, 16, 27] using the relevant subset of tubes (Ψ) abstracted according to a given user-defined query.

Table 5 presents the quantitative evaluation results of the CR, OR, FOA, NPI and TD metrics on 8 test videos. The results are average performance on 8 videos, assessed

Table 3 Quantitative analysis of different query classes and tube attributes

Query Class	Attribute	Precision	Recall	F-score
Visual	Color	0.77	1.0	0.85
Visual	Size	0.91	0.77	0.83
Visual	Type	1.0	0.93	0.97
Temporal	Speed	1.0	0.60	0.75
Temporal	Direction	0.82	0.93	0.88
Spatial	Path	0.90	0.75	0.82
Spatial	ROI	0.94	0.97	0.96

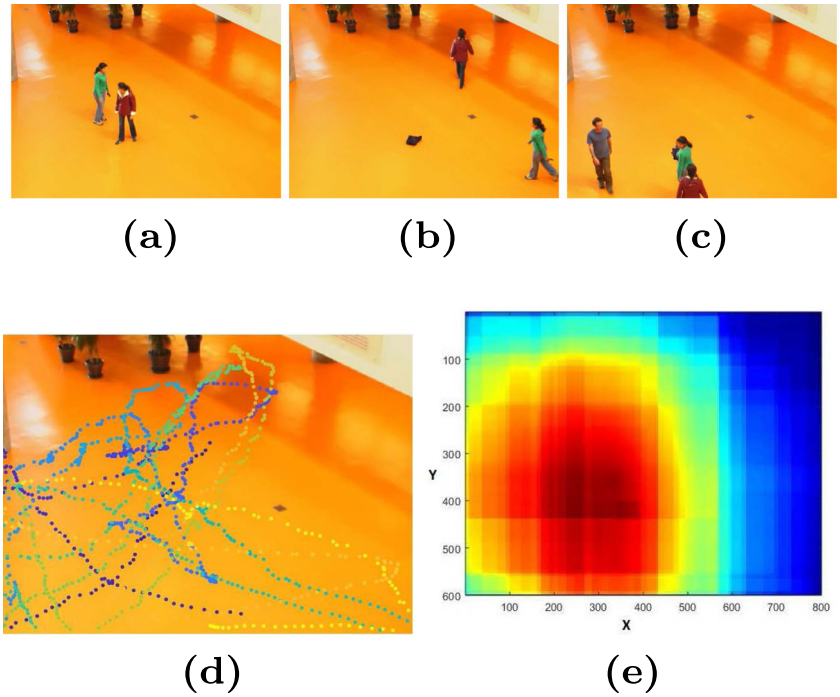

Fig. 7 Traffic flow of video V8. **a**-**c** Representative frames from the temporal period (2:20, 4:40). **d** Trajectories of activities during the temporal period (2:20, 4:40). **e** Heat map of activities for the whole video

across different query classes. The comparison results show that the approach, put forth in this study, achieves best performance in terms of CR, OR, FOA and NPI among the five methods. The proposed approach obtained remarkably less FOA than the state-of-the-art methods [7, 9, 16, 27], even for those that do not show significant differences in the OR scores. From this, it is evident that the proposed approach generates only a few new collisions in the synopsis video, which are not there in the input video. Thus, false collisions contribute very less towards the total collisions (OR) in synopsis generated by our approach, while OR obtained by the other methods is mainly due to false collisions rather than original collisions that are maintained in the synopsis.

Similarly, the proposed approach preserved all original spatio-temporal interactions among tubes in the synopsis, achieving NPI scores of 0, for every test video. We can also see that the method noted in [9] obtained an NPI score of 1 for all videos. This is due to the fact that the approach in [9] scales down the sizes of colliding objects in the synopsis video and relocates them to optimal temporal locations, which may alter their spatio-temporal relationships. The methods proposed in [16, 27] and [7] preserve tube relationships with NPI values in the range of 0 - 0.44, 0 - 0.58 and 0 - 0.71, respectively.

Additionally, the results of comparison show that the method of [9] obtains the lowest TD values, versus the approach-under-study achieved the second-lowest TD, overall. This is because the proposed method focus on preserving the relative temporal order of tubes within each group, and reducing the false collisions. Meanwhile, the optimal temporal locations determined for the placement of interacting groups may alter the chronological order among tube groups. However, the main objective of video synopsis is to determine optimal tube rearrangement for generating a condense video with or without destroying the chronological sequence.

The comparison of average runtime (in seconds) for the test videos are depicted in Fig. 8. It can be seen from Fig. 8 that the proposed approach takes less running time than that of [7, 9, 27] and [16]. From Table 5, it is evident that the proposed approach creates less false collisions, and preserves original spatio-temporal tube interactions noticeably better than the other four methods, with lower TD and larger CR values, even for varying number of tubes in the original video.

5.5 Qualitative results

5.5.1 Visual comparison

We further present sample results for visual comparison in Figs. 9 and 10. False collisions and unlikeable visual effects in the synopsis are represented using red ellipses. Interactions in the original video that are maintained in the synopsis video are highlighted with green ellipses. Blue ellipses represent the original interactions that are modified during synopsis generation. More than one observations corresponding to same object, which are displayed simultaneously in the synopsis are denoted by

Table 5 Quantitative comparisons between previous methods [7, 9, 16, 27] and proposed approach across different queries

Query Class	Attribute	Method	CR	OR	FOA	NPI	TD
Visual	Color	[7]	0.0893	0.171	3.09×10^6	0.67	1.562
		[9]	0.0902	0.146	5.94×10^6	1.0	0.189
		[27]	0.0897	0.157	3.63×10^6	0.31	1.491
		[16]	0.0882	0.189	4.12×10^6	0.54	1.422
		Our	0.0934	0.125	2.67×10^6	0.0	1.131
Visual	Size	[7]	0.0471	0.262	7.18×10^5	0.56	3.141
		[9]	0.0508	0.115	6.76×10^5	1.0	1.782
		[27]	0.0488	0.183	5.21×10^5	0.35	2.623
		[16]	0.0492	0.205	6.63×10^5	0.28	2.916
		Our	0.0514	0.086	3.24×10^5	0.0	2.152
Visual	Type	[7]	0.0272	0.163	3.67×10^5	0.45	2.785
		[9]	0.3016	0.137	5.88×10^5	1.0	0.831
		[27]	0.0298	0.153	3.06×10^5	0.32	2.053
		[16]	0.0262	0.198	2.63×10^5	0.39	2.601
		Our	0.0321	0.131	2.12×10^5	0.0	1.937
Temporal	Speed	[7]	0.0353	0.223	6.25×10^5	0.28	1.732
		[9]	0.0433	0.125	4.92×10^5	1.0	0.566
		[27]	0.0372	0.146	4.84×10^5	0.24	1.467
		[16]	0.0399	0.179	5.23×10^5	0.31	1.582
		Our	0.0417	0.104	3.58×10^5	0.0	1.381
Temporal	Direction	[7]	0.0511	0.311	6.88×10^6	0.58	2.571
		[9]	0.0519	0.236	7.71×10^6	1.0	1.353
		[27]	0.0474	0.322	6.25×10^6	0.44	2.604
		[16]	0.0509	0.318	6.29×10^6	0.46	2.456
		Our	0.0543	0.217	5.96×10^6	0.0	2.322
Spatial	Path	[7]	0.1012	0.178	4.05×10^6	0.71	1.899
		[9]	0.1223	0.131	3.26×10^6	1.0	1.052
		[27]	0.1201	0.164	3.83×10^6	0.21	1.731
		[16]	0.1181	0.171	4.03×10^6	0.35	1.752
		Our	0.1263	0.129	2.31×10^6	0.0	1.193
Spatial	ROI	[7]	0.0236	0.183	2.62×10^6	0.45	7.534
		[9]	0.0301	0.118	3.58×10^6	1.0	2.929
		[27]	0.0284	0.161	3.02×10^6	0.40	7.402
		[16]	0.0252	0.127	2.05×10^6	0.56	6.561
		Our	0.0299	0.093	1.95×10^6	0.0	6.732
Spatio-Temporal	Interaction	[7]	0.1192	0.133	4.21×10^6	0.67	2.756
		[9]	0.1635	0.095	4.92×10^6	1.0	0.634
		[27]	0.1083	0.106	2.56×10^5	0.31	2.055
		[16]	0.1201	0.119	4.52×10^6	0.58	2.510
		Our	0.1721	0.069	1.24×10^5	0.0	1.541

The results are average performance on 8 test videos

yellow ellipses. The false positives in synopsis results are represented using green rectangles.

The results in last row of Fig. 9 and last column of Fig. 10, demonstrate the effectiveness of the proposed approach that generates synopsis with almost no unpleasant visual effects and false collisions. Compared to the video synopsis methods in [7, 9, 27], and [16], our proposed framework preserves the spatial and temporal interactions among related tubes by maintaining their relative temporal order in synopsis video.

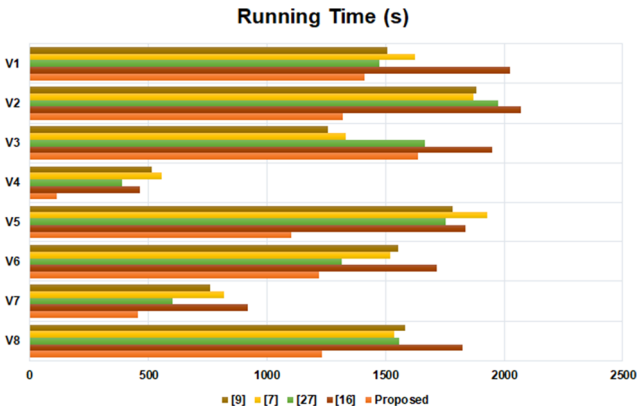

Fig. 8 Comparison of running times (in seconds) between state-of-the-art video synopsis methods [7, 9, 16, 27] and proposed approach on 8 test videos

5.5.2 Subjective evaluation

For an additional evaluation of the usability and effectiveness of the proffered approach, a subjective user study was conducted among 20 participants, aged from 25 to 45 years. They were asked to watch the original videos first, and then, the corresponding synopsis videos. The participants were requested to provide their ratings on four criteria, specified in Table 6. For the first three questions, the participants were asked to rate on a scale of 1 (worst) to 5 (best). In question four, the participants were asked to provide a score of 1 for any one of the most satisfied synopsis video and a score of 0, if not satisfied with any synopsis videos. The average results of subjective feedback and related statistics of overall synopsis are illustrated in Fig. 11. We can see that the proposed approach performed better in assuring a pleasant, interaction-persevered synopsis video.

5.6 Complexity analysis

The computational complexity of proposed visualization framework mainly depends on Algorithm 1 and Algorithm 2. The time complexity of the recursive grouping method (Algorithm 1) in the proposed work is $O(N^2F)$, where N and F denote the number of tubes and frames in the

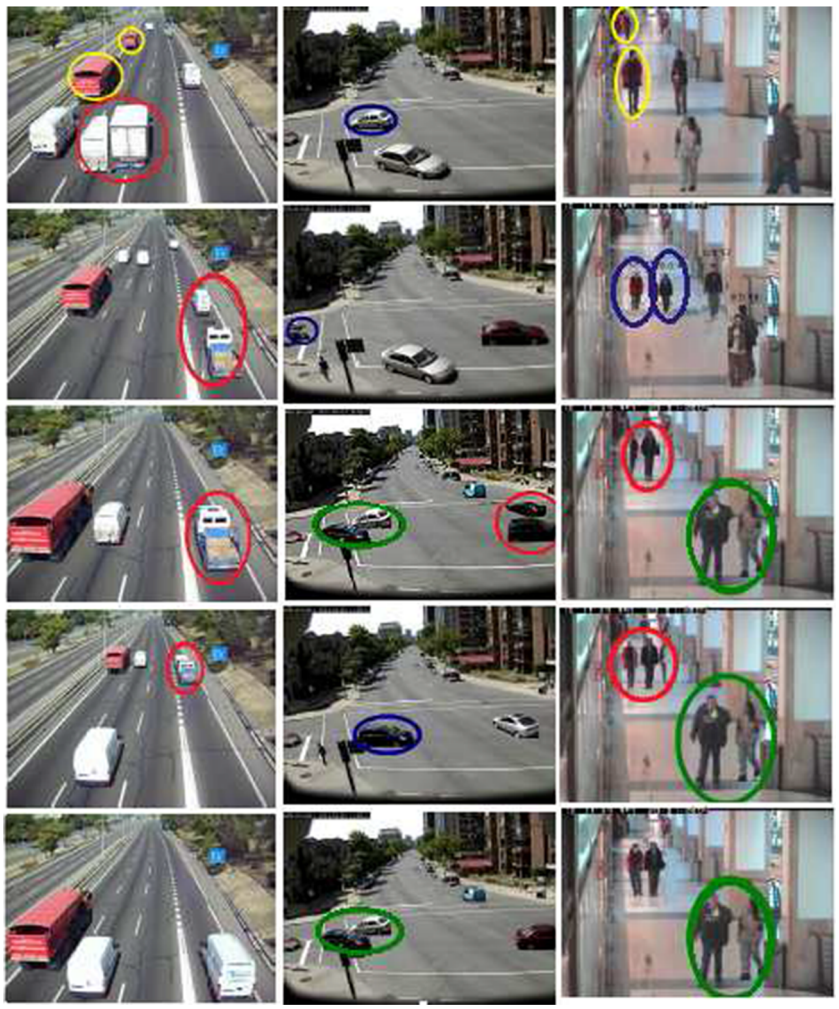

Fig. 9 Visual comparison of synopsis results. From top to bottom: 1st, 2nd, 3rd, 4th and 5th rows corresponds to sample frames from results using [7, 9, 16, 27], and proposed approach, respectively. From left to right: 1st, 2nd and 3rd columns corresponds to results of video V1 in response to a visual query for "large vehicles", video V4 in response to a visual query for "cars and pedestrians", and video V7 in response to a spatio-temporal query for "interacting groups", respectively

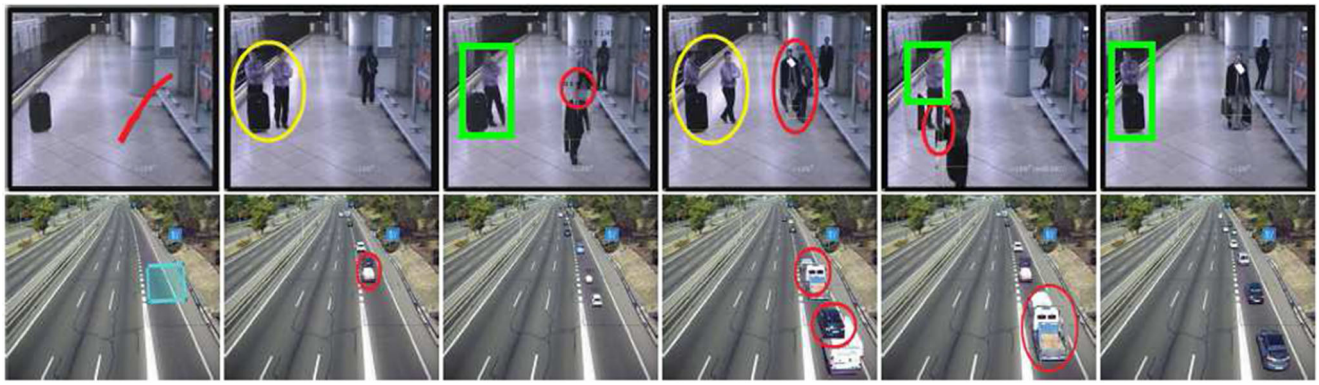

Fig. 10 Visual comparison of synopsis results in response to spatial queries. Top row: results of interested path of travel drawn on video V5 in 1st column. Bottom row: results of ROI marked on video V1 in 1st column. From left to right: 2nd, 3rd, 4th, 5th, and 6th columns corresponds to sample frames from results using [7, 9, 16, 27], and proposed approach, respectively

Table 6 Subjective evaluation questionnaire

No.	Question
1	Do you think the synopsis is "**pleasant**" to view?
2	Is this synopsis "**compact**" enough?
3	Do you think this synopsis "**preserves original behavioral interactions**" between tubes?
4	Which synopsis do you consider as "**overall satisfied**"?

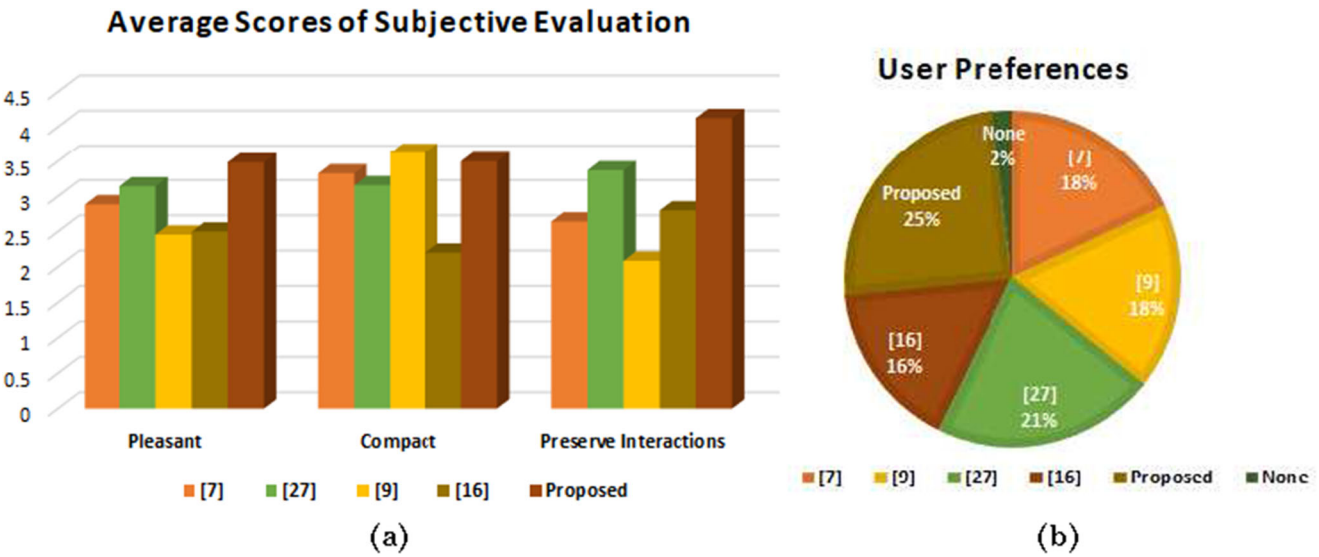

Fig. 11 **a** Average scores of subjective feedbacks for questions 1 to 3. **b** User preferences for overall satisfied synopsis

input video, respectively. The time complexity of proposed the tube group rearrangement method (Algorithm 2) is $O(MPC)$, where M denotes the number of tube groups, P denotes the feasible temporal positions of tube groups, and C represents the number of cubes in space-time synopsis volume. The time complexity of the proposed framework mainly depends on the number of tubes, relationships between tubes and synopsis length. Whereas the computational time of the off-line synopsis methods in [7] and [9] with complexities $O(T^N)$ and $O(T^N + X^N)$, respectively, will grow exponentially with an increase in the number of tubes, where T is the feasible temporal positions of tubes and X represents the search space of reduction coefficients. The proposed work computes the relationships between tubes in each tube group only once, while the methods in [7] and [9] compute the energy between tubes in each iteration of the optimization process. Hence, the total time complexity of the proposed work is lesser that of the time complexity reported in [7] and [9].

6 Conclusion

In this paper, we presented an interactive visualization framework to generate user-oriented synopsis of surveillance videos. Using arbitrary combinations of visual, spatial and temporal attributes of tubes, users are allowed to create queries with the support of a user-friendly GUI. Tubes relevant to user's query are selected, and related tubes are grouped together, to preserve original interactions. False tube collisions are minimized by the optimal rearrangement of tube groups. The experimental results demonstrate the usability and effectiveness of our proposed approach in generating targeted synopsis videos.

Funding No funding was received for conducting this study.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Chamasemani FF, Affendey LS, Mustapha N, Khalid F (2018) Video abstraction using density-based clustering algorithm. Vis Comput 34(10):1299–1314
2. Nguyen HT, Jung SW, Won CS (2016) Order-preserving condensation of moving objects in surveillance videos. IEEE Trans Intell Transp Syst 17(9):2408–2418
3. Murugan AS, Devi KS, Sivaranjani A, Srinivasan P (2018) A study on various methods used for video summarization and moving object detection for video surveillance applications. Multimed Tools Appl 77(18):23,273–23,290
4. Elharrouss O, Almaadeed N, Al-Maadeed S, Bouridane A, Beghdadi A (2020) A combined multiple action recognition and summarization for surveillance video sequences. Appl Intell:1–23
5. Rav-Acha A, Pritch Y, Peleg S (2006) Making a long video short: Dynamic video synopsis. In: IEEE Computer society conference on computer vision and pattern recognition (CVPR'06), vol 1, pp 435–441
6. Pritch Y, Rav-Acha A, Gutman A, Peleg S (2007) Webcam synopsis: Peeking around the world. In: IEEE 11th International Conference on Computer Vision, pp 1–8
7. Pritch Y, Rav-Acha A, Peleg S (2008) Nonchronological video synopsis and indexing. IEEE Trans Pattern Anal Mach Intell 30(11):1971–1984
8. Namitha K, Narayanan A (2018) Video synopsis: State-of-the-art and research challenges. In: 2018 International conference on circuits and systems in digital enterprise technology (ICCSDET). IEEE, pp 1–10
9. Li X, Wang Z, Lu X (2015) Surveillance video synopsis via scaling down objects. IEEE Trans Image Process 25(2):740–755
10. He Y, Qu Z, Gao C, Sang N (2016) Fast online video synopsis based on potential collision graph. IEEE Signal Process Lett 24(1):22–26
11. He Y, Gao C, Sang N, Qu Z, Han J (2017) Graph coloring based surveillance video synopsis. Neurocomputing 225:64–79
12. Nie Y, Li Z, Zhang Z, Zhang Q, Ma T, Sun H (2019) Collision-free video synopsis incorporating object speed and size changes. IEEE Trans Image Process 29:1465–1478
13. Ra M, Kim WY (2018) Parallelized tube rearrangement algorithm for online video synopsis. IEEE Signal Process Lett 25(8):1186–1190
14. Ruan T, Wei S, Li J, Zhao Y (2019) Rearranging online tubes for streaming video synopsis: A dynamic graph coloring approach. IEEE Transactions on Image Processing
15. Ghatak S, Rup S, Majhi B, Swamy M (2019) An improved surveillance video synopsis framework: a HSATLBO optimization approach. Multimed Tools Appl:1–33
16. Moussa MM, Shoitan R (2020) Object-based video synopsis approach using particle swarm optimization. SIViP:1–8
17. Zhu J, Liao S, Li SZ (2016) Multicamera joint video synopsis. IEEE Trans Circ Syst Video Technol 26(6):1058–1069
18. Zhang Z, Nie Y, Sun H, Zhang Q, Lai Q, Li G, Xiao M (2019) Multi-view video synopsis via simultaneous object-shifting and view-switching optimization. IEEE Trans Image Process 29:971–985
19. Nie Y, Xiao C, Sun H, Li P (2012) Compact video synopsis via global spatiotemporal optimization. IEEE Trans Vis Comput Graph 19(10):1664–1676
20. Pritch Y, Ratovitch S, Hendel A, Peleg S (2009) Clustered synopsis of surveillance video. In: Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, pp 195–200
21. Zhu J, Feng S, Yi D, Liao S, Lei Z, Li SZ (2014) High-performance video condensation system. IEEE Trans Circ Syst Video Technol 25(7):1113–1124

22. Ahmed A, Kar S, Dogra DP, Patnaik R, Lee S, Choi H, Kim I (2017) Video synopsis generation using spatio-temporal groups. In: 2017 IEEE International conference on signal and image processing applications (ICSIPA). IEEE, pp 512–517
23. Li X, Wang Z, Lu X (2018) Video synopsis in complex situations. IEEE Trans Image Process 27(8):3798–3812
24. Chou CL, Lin CH, Chiang TH, Chen HT, Lee SY (2015) Coherent event-based surveillance video synopsis using trajectory clustering. In: IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp 1–6
25. Lin W, Zhang Y, Lu J, Zhou B, Wang J, Zhou Y (2015) Summarizing surveillance videos with local-patch-learning-based abnormality detection, blob sequence optimization, and type-based synopsis. Neurocomputing 155:84–98
26. Wang WC, Chung PC, Huang CR, Huang WY (2017) Event based surveillance video synopsis using trajectory kinematics descriptors. In: 2017 Fifteenth IAPR international conference on machine vision applications (MVA). IEEE, pp 250–253
27. Ahmed SA, Dogra DP, Kar S, Patnaik R, Lee SC, Choi H, Nam GP, Kim IJ (2019) Query-based video synopsis for intelligent traffic monitoring applications. IEEE Transactions on Intelligent Transportation Systems
28. Benfold B, Reid I (2011) Stable multi-target tracking in real-time surveillance video. In: CVPR, IEEE, pp 3457–3464
29. Pérez-Hernández F, Tabik S, Lamas A, Olmos R, Fujita H, Herrera F (2020) Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance. Knowl-Based Syst 194:105,590
30. Haritha H, Thangavel SK (2019) A modified deep learning architecture for vehicle detection in traffic monitoring system. Int J Comput Appl:1–10
31. Yu X, Ye X, Gao Q (2020) Infrared handprint image restoration algorithm based on apoptotic mechanism, vol 8
32. Liu X, Zhu X, Li M, Wang L, Zhu E, Liu T, Kloft M, Shen D, Yin J, Gao W (2019) Multiple kernel k k-means with incomplete kernels. IEEE Trans Pattern Anal Mach Intell 42(5):1191–1204
33. Aarthi R, Amudha J, Boomika K, Varrier A (2016) Detection of moving objects in surveillance video by integrating bottom-up approach with knowledge base. Procedia Comput Sci 78:160–164
34. Subbiah U, Kumar DK, Thangavel SK, Parameswaran L (2020) An extensive study and comparison of the various approaches to object detection using deep learning. In: 2020 International conference on smart electronics and communication (ICOSEC). IEEE, pp 183–194
35. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv
36. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
37. Wojke N, Bewley A, Paulus D (2017) Simple online and realtime tracking with a deep association metric. In: In: 2017 IEEE international conference on image processing (ICIP). IEEE, pp 3645–3649
38. Tang Y, Zhang H, Xu B (2015) Metadata organization and retrieval with attribute tree for large-scale traffic surveillance videos. In: International Conference on Big Data Computing and Communications. Springer, pp 434–443
39. Castañón G, Elgharib M, Saligrama V, Jodoin PM (2015) Retrieval in long-surveillance videos using user-described motion and object attributes. IEEE Trans Circ Syst Video Technol 26(12):2313–2327
40. Momin BF, Mujawar TM (2015) Vehicle detection and attribute based search of vehicles in video surveillance system. In: 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]. IEEE, pp 1–4
41. Thomas SS, Gupta S, Subramanian VK (2016) Perceptual synoptic view of pixel, object and semantic based attributes of video. J Vis Commun Image Represent 38:367–377
42. Kamat V (1993) A survey of techniques for simulation of dynamic collision detection and response. Comput Graph 17(4):379–385
43. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
44. Kolmogorov V, Zabih R (2002) What energy functions can be minimized via graph cuts? In: European conference on computer vision. Springer, pp 65–81
45. Pérez P, Gangnet M, Blake A (2003) Poisson image editing. ACM Trans Graph (TOG) 22(3):313–318
46. Baskurt KB, Samet R (2019) Video synopsis: a survey. Comput Vis Image Underst 181:26–38
47. Guerrero-Gomez-Olmedo R, Lopez-Sastre RJ, Maldonado-Bascon S, Fernandez-Caballero A (2013) Vehicle tracking by simultaneous detection and viewpoint estimation. In: IWINAC 2013, Part II, LNCS, vol 7931, pp 306–316
48. Jodoin JP, Bilodeau GA, Saunier N (2014) Urban tracker: Multiple object tracking in urban mixed traffic. In: IEEE Winter Conference on Applications of Computer Vision. IEEE, pp 885–892
49. Branch HOSD (2006) Imagery library for intelligent detection systems (i-lids). In: 2006 IET Conference on Crime and Security. IET, pp 445–448
50. Fisher R, Santos-Victor J, Crowley J (2003) Ec funded caviar project/ist 2001 37540 Available at http://homepages.inf.ed.ac.uk/rbf/CAVIAR/ (accessed 21 March 2020)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

K. Namitha received her B.Tech. degree in information technology and M.Tech. degree in computer science and engineering from Amrita Vishwa Vidyapeetham, India in 2008 and 2015, respectively. From 2008 - 2013, she was a research associate at Amrita Technology Business Incubator. She is currently pursuing her PhD in computer science and engineering at Amrita Vishwa Vidyapeetham, Amritapuri Campus, India. Her research interests include video processing, computer vision, and machine learning.

Athi Narayanan received a PhD in CSE from Amrita Vishwa Vidyapeetham, India in 2016. He was an Assistant Professor (SG) with the department of computer science and engineering at Amrita Vishwa Vidyapeetham, Amritapuri, India from 2016 - 2019 and is now a Lead Engineer of Computer Vision at Kimball Electronics, India. He has published two journal articles in IEEE Transactions and holds two US patents on head pose estimation. He has a world rank of 10 on MATLAB Cody. His team secured the second position in the 2019 IEEE CVPR NTIRE Image Colorization Challenge. His research interests include image processing, computer vision and biblical theology.

M. Geetha received a PhD in CSE from Amrita Vishwa Vidyapeetham, India in 2019. She is currently an assistant professor and vice chairperson with the department of computer science and engineering at Amrita Vishwa Vidyapeetham, Amritapuri, India. She has been with the department of computer science and engineering, Amritapuri campus since 2005. She has completed her masters from Amrita Vishwa Vidyapeetham and has a teaching experience of over 15 years. Her research interest includes the area of video analytics, machine learning, deep learning for edge devices and computer vision. She has funded project and patent in the area of video analytics.