




# Preserving interactions among moving objects in surveillance video synopsis

Namitha K<sup>1</sup>  · Athi Narayanan<sup>1</sup>

Received: 1 October 2019 / Revised: 10 July 2020 / Accepted: 29 July 2020 /  
Published online: 27 August 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Video synopsis is an effective solution for fast browsing and retrieval of long surveillance videos. It aims to shorten long video sequences into its equivalent compact video representation by rearranging the video events in the temporal domain and/or spatial domain. Conventional video synopsis methods focus on reducing the collisions between tubes and maintaining their chronological order, which may alter the original interactions between tubes due to improper tube rearrangement. In this paper, we present an approach to preserve the relationships among tubes (tracks of moving objects) of the original video in the synopsis video. First, a recursive tube-grouping algorithm is proposed to determine the behavior interactions among tubes in a video and group the related tubes together to form tube sets. Second, to preserve the discovered relationships, a spatio-temporal cube voting algorithm is proposed. This cube voting method optimally rearranges the tube sets in the synopsis video, minimizing false collisions between tubes. Third, a method to estimate the duration of the synopsis video is proposed based on an entropy measure of tube collisions. The extensive experimental results demonstrate that the proposed video synopsis framework condenses videos by preserving the original tube interactions and reducing false tube collisions.

**Keywords** Video synopsis · Surveillance · Interaction · Tube grouping · Tube rearrangement

## 1 Introduction

With the growing demand for security solutions in public and private sectors, surveillance cameras play a huge role in our day-to-day lives. The enormous amount of videos captured by these cameras are increasing explosively, creating challenges in its effective retrieval and review. In most cases, surveillance footage contains highly redundant data with only lim-

---

✉ Namitha K  
namitha.amrita@gmail.com

Athi Narayanan  
mail2athi@gmail.com

<sup>1</sup> Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India

ited useful information. Therefore, it is time-consuming to browse through such massive video data for inspecting an event of interest. Several techniques such as video fast forward [17], video skimming [30], video montage [21], video condensation [25] and video summarization [8, 12, 26] have been proposed to address the aforementioned challenge. However, these methods lose the dynamic aspect of video or have apparent stitching seams in the summary. To address these drawbacks, an object-based technique, video synopsis [38, 39, 41] has been proposed in the literature.

Video synopsis provides a compact video representation of long surveillance videos where multiple objects are displayed simultaneously, irrespective of their time of occurrence. The fundamental processing block for video synopsis methods is a *tube*, which is a sequence of positions of a moving object in 3D space-time volume. The extracted tubes in the original video are rearranged along the temporal axis to reduce the spatio-temporal redundancies. The new optimal start-times for the shifted tubes are generally determined by minimizing a well-defined energy function [22, 23]. To generate summarized videos with high condensation ratio, Rav-Acha et al. [41] proposed an object-based approach that shifts moving objects in the temporal domain. Pritch et al. [38, 39] extended this method for endless video streams and proposed a two-phase synopsis framework. Following the classical framework [39, 41] in video synopsis, several other approaches have been proposed by the research community.

Due to lack of unified qualitative standards to determine the effectiveness of synopsis video, the evaluation of results depends on the input video content and application. However, some of the standards mentioned in [2, 39] that should be satisfied by a synopsis video are as follows: 1) The length of the synopsis video should be lesser than the original video and it should preserve maximum activities of the original video; 2) Collisions between moving objects should be minimized as far as possible; 3) The temporal order between objects should be preserved as much as possible in the synopsis video.

In any video sequence, the spatio-temporal collisions between moving objects can be considered as some sort of behavioral interactions among them. For example, the overlap of body parts of multiple persons in a video may indicate people walking together, normal conversation, fight or personal attack. The collisions between moving objects in the original video, which indicates important behavior interactions, are generally termed as *true collisions*. Collisions that occur in the synopsis video due to the temporal and spatial rearrangement of object tubes, but which does not exist in the original video, are termed as *false collisions*.

## 1.1 Motivation and overview

Most of the existing methods [27, 32, 39] in video synopsis take either collision avoidance or chronological order preservation into consideration during the generation of synopsis. In order to reduce false collisions between tubes or to increase the condensation rate, the related tubes may be shifted to inappropriate temporal locations in the synopsis video, which may alter the original behavior interactions between them. Moreover, the synopsis video depends highly on the tracking and segmentation results of moving objects. Several tracking errors are possible due to occlusion and object cross scenes in the video, where the tubes involved will be tracked as new tubes after such scenarios. Furthermore, a single tube can be segmented into several tubes due to over-segmentation. Hence, preserving the original object interactions become a challenge during synopsis generation while trying to achieve a trade-off among maintaining all activities, reducing collisions and preserving temporal order

between tubes. In this paper, we aim to discover and preserve the relationships between tubes by grouping all related tubes using a recursive tube-grouping method.

During synopsis generation, conventional video synopsis approaches [13, 33, 39] carry out the rearrangement of tubes immediately after extracting tubes from video. The tube rearrangement process computes the new optimal start-time labels to tubes for positioning them in synopsis video. However, inapt spatial and temporal rearrangement of tubes in a synopsis video may generate false interaction perceptions. In view of this problem, we can observe that the underlying reason for false tube collisions is the simultaneous overlap of 3D trajectories of tubes in the spatio-temporal domain. Therefore, to reduce false collisions, we have to prevent the sharing of same spatial location at the same time (spatio-temporal cube) in the synopsis video by multiple tubes that do not overlap in the original video. To this end, we propose a spatio-temporal cube voting algorithm by considering the synopsis video as a three-dimensional volume  $V$ , created using several space-time cubes. This spatio-temporal cube voting approach optimally rearranges the interacting tube sets, which are identified, and grouped using the proposed recursive tube-grouping method.

In most of the conventional video synopsis methods [14, 27, 39], the length of synopsis video is either user-defined or randomly determined. Nevertheless, the length of the synopsis video has to be computed by taking the video content into consideration, rather than determined by the user who is not much aware about the actual object density in the video. Hence, we propose a method to determine the length of a synopsis video using an entropy measure of tube collisions in original video.

## 1.2 Contributions

To summarize, the three main contributions of our work are as follows:

- A *recursive tube-grouping algorithm* is proposed that identifies and binds interacting tubes together to form tube sets for preserving the strong behavior interactions among tubes, utilizing their spatial and temporal proximity.
- A *spatio-temporal cube voting algorithm* is presented for optimal positioning of tube sets in the 3D space-time volume representation of synopsis video, to maintain tube interactions discovered by our tube-grouping approach. This method aims to reduce false tube collisions by minimizing the number of tubes that share each space-time cube in the 3D representation using a method of cubes voting to possible temporal locations of tube sets.
- A *length estimation method* for synopsis video is proposed using an entropy-based measure of tube collisions and duration of the longest tube set.

The contribution of this paper is integrating the three above components together into a single framework for the generation of synopsis video, preserving object interactions while minimizing false collisions.

The rest of the paper is organized as follows. Section 2 outlines the related works. Section 3 details the proposed approach. Section 4 presents the experimental results and Section 5 concludes the paper.

## 2 Related work

This section reviews the related existing works in video synopsis. The various state-of-the-art methods for synopsis generation can be roughly categorized based on different aspects

such as optimization framework, camera topology, and trajectory analysis. Table 1 gives a comprehensive list of the existing approaches in each of the different video synopsis category. In the following, we discuss the relevant works under each classification of video synopsis.

## 2.1 Off-line optimization frameworks

An off-line optimization method that improves condensation ratio and utilization of spatial domain by shifting objects both spatially and temporally is presented by Nie et al. [33]. Additionally, the moving space of the input video is expanded to include the shifted objects using a multilevel patch relocation (MPR) method. Li et al. [27] proposed a two-stage optimization framework that aims to reduce object collisions by decreasing the size of objects during a collision. To determine the collision relationship between tubes, He et al. [15] formulated the tube rearrangement as a graph coloring problem. A comprehensive framework, which extracts tubes and video clips for handling crowdedness of video is proposed in [28]. The proposed approach aims to discover the relationships between tubes and group them accordingly. Further, a greedy optimization method rearranges the groups to form condensed video. To reduce the computational time of energy minimization problems in video synopsis, Ghatak et al. [13] presented a hybrid optimization approach using simulated annealing and teaching learning-based optimization algorithms. A synopsis method to reduce collisions by varying the speeds of moving objects, concurrent with size scaling is proposed in [32]. Ahmed et al. [2] presented an approach, which takes a few types of user queries into consideration for generating synopsis video. An hybrid method using simulated annealing and jaya algorithms for minimizing energy in synopsis generation is proposed in [14]. Although the off-line optimization techniques produce substantially best synopsis results, they are time-consuming and requires large memory.

## 2.2 On-line optimization frameworks

To speed up the optimization process and obtain synopsis videos in real-time, Feng et al. [10] converted the conventional two-phase off-line video synopsis process into a single phase on-line framework. In [19], a MAP estimation-based approach is presented to determine the start-time labels of objects using an online synopsis table. Zhu et al. [51] further extended the method in [19] by transforming tube rearrangement problem into a stepwise optimization problem, which uses graphics processing unit (GPU) support. He et al. [16]

**Table 1** Classification of video synopsis state-of-the-art

Optimization Framework		Camera Topology	Trajectory-Based
Off-line	On-line		
Rav-Acha et al. [41], Pritch et al. [38, 39], Nie et al. [33], Li et al. [27, 28], He et al. [15], Ghatak et al. [13, 14], Nie et al. [32], Ahmed et al. [2]	Feng et al. [10], Huang et al. [19], Zhu et al. [51], He et al. [16], Ra et al. [40], Ruan et al. [43]	Zhu et al. [53], Mahapatra et al. [31], Hoshen and Peleg [18], Zhu et al. [52], Zhang et al. [50]	Lu et al. [29], Xu et al. [48], Chou et al. [7], Wang et al. [47], Pritch et al. [37]

proposed an online tube rearrangement method using a potential collision graph. In advance, the graph-based strategy determines the probable tube collisions that can occur in the synopsis video. To speed up video condensation when the number of tubes in the input video are large, Ra et al. [40] presented a parallelized tube rearrangement approach based on fast Fourier transform. Ruan et al. [43] proposed a dynamic graph coloring based tube rearrangement algorithm for streaming videos. The method models the relationships between tubes using a dynamic graph, which is updated progressively according to video streaming.

### 2.3 Camera topology

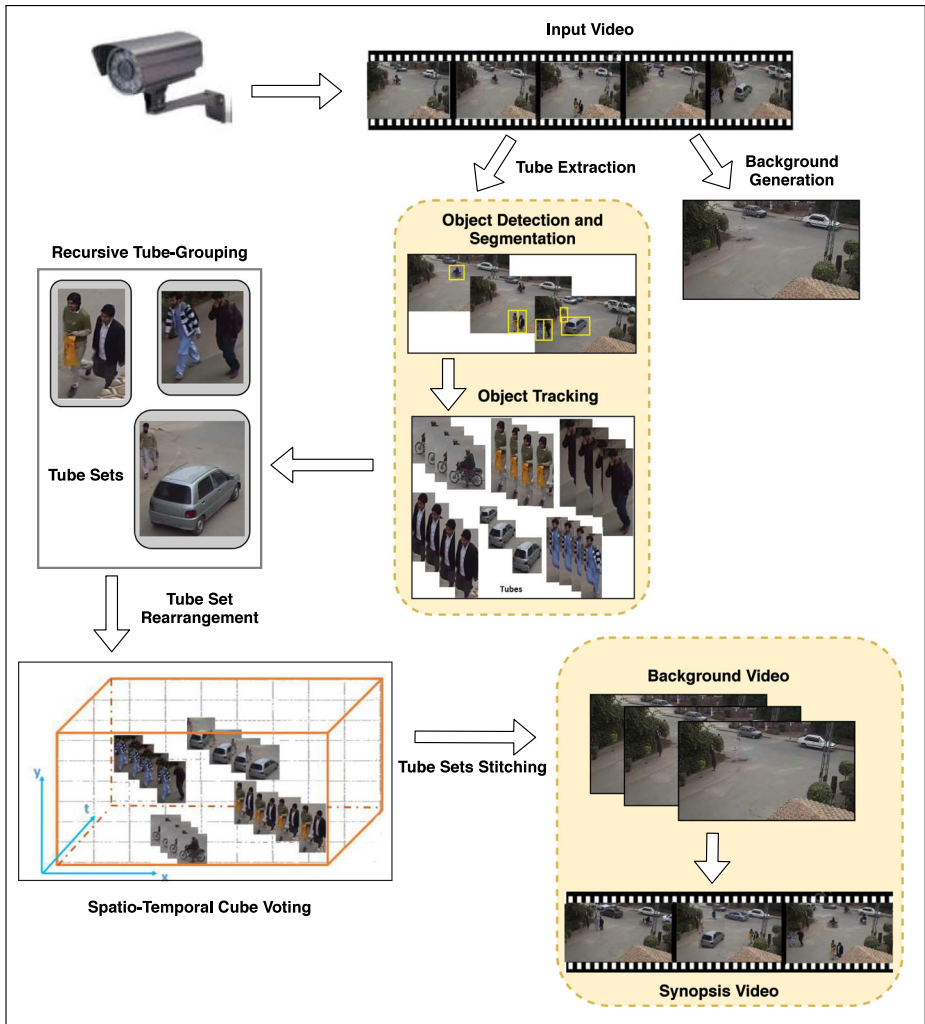
Most conventional video synopsis approaches process videos with single camera views. To improve the efficiency of synopsis results, video synopsis methods that work with multi-camera views are presented. Zhu et al. [53] proposed a synopsis method for multiple camera environment with overlapping views. Objects and backgrounds from multicamera cameras are concatenated while generating video synopsis. Similar to [53], Mahapatra et al. [31] presented another multi-camera approach that extracts objects in the field-of-view of different cameras, prioritize and associate them using homography and trajectory matching. Hoshen and Peleg [18] proposed a master-slave camera hierarchy with multiple slave cameras associated with a master camera. Live videos are captured by the master camera and objects from different slave camera streams are identified without aggregating them. Zhu et al. [52] proposed a joint video synopsis approach that preserves chronological order of objects globally, among different camera views. An optimization approach is proposed in [50] based on graph cuts and dynamic programming to present synopsis results from multiple cameras in an understandable way to the users.

### 2.4 Trajectory-based

The trajectories of moving objects are spatio-temporal tubes that provide an insight into the motion proximity and interaction among objects. Trajectory analysis of 3D object tubes is applied for optimal tube extraction [29], trajectory clustering [7, 37, 47] and combining multiple trajectories [48]. Clustering of trajectories based on a similarity measurement as the longest common subsequence is proposed in [7], motion similarity and appearance distance in [37], event-based trajectory kinematics descriptors [47]. A survey on the various synopsis methods is presented in [20] and [3].

## 3 The proposed work

In this section, the proposed framework is described in detail. Figure 1 illustrates the main procedures of our proposed framework which primarily consists of four steps. Given a video, the foremost step is preprocessing by detecting and segmenting [1, 9, 34, 45] the moving objects, followed by multiple objects tracking [10, 24, 44], collectively known as tube extraction. Secondly, the tubes which are highly interactive are then identified and grouped together to form tube sets using the proposed recursive tube-grouping algorithm. To start with tube set positioning, the duration of the synopsis video is determined using an entropy-based measure of tube collisions and length of the longest interacting tube set. Subsequently, the start-time labels of tube sets are determined using the proposed spatio-temporal cube voting algorithm, where the relative temporal relationships between tubes in each tube set are maintained. As the final step, all tube sets are stitched into the background video to form

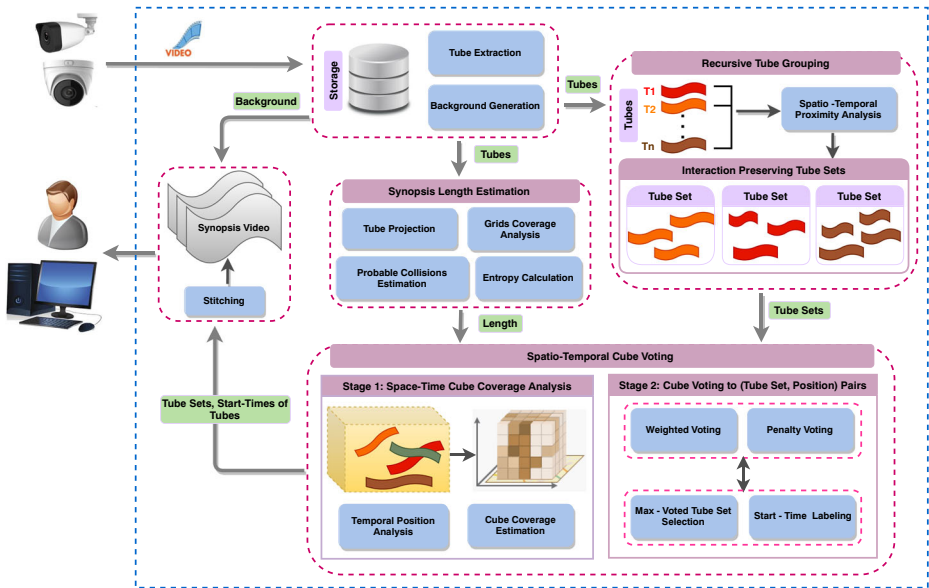


**Fig. 1** An overview of the proposed video synopsis framework. In the beginning, tubes are extracted from the original video. Then, the interacting tubes are grouped to form tube sets by our recursive tube-grouping algorithm. A spatio-temporal cube voting algorithm rearranges the interacting tube sets, assigning new start-times to tubes. Finally, the tube sets are stitched into the background to generate a synopsis video

synopsis video according to the start-times computed in the cube voting step. The system architecture with a high level overview of the above mentioned proposed components and other pre-processing/post-processing functional parts is presented in Fig. 2.

### 3.1 Tube extraction

A preprocessing step in synopsis generation is tube extraction. Given a video, the extraction of tubes begins with the detection and segmentation of moving objects. Generally, surveillance cameras are static so that the illumination variation in the background will be slow. Hence, a background subtraction based on Gaussian mixture models [42] is employed



**Fig. 2** Overall system architecture of the proposed video synopsis system consisting of three main components: Recursive tube grouping, Spatio-temporal cube voting, Synopsis length estimation, and other functional parts

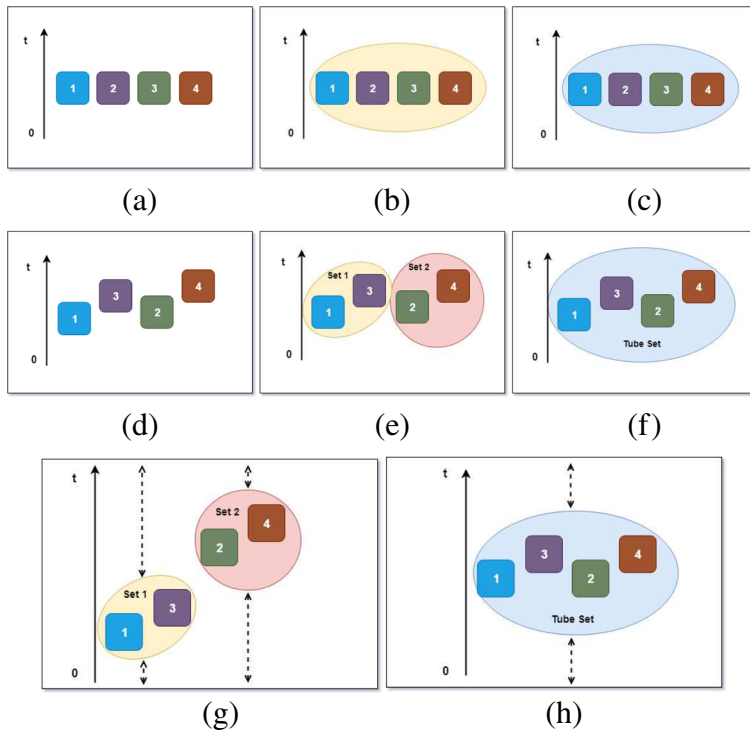
to detect the foreground pixels. Morphological operations are applied to eliminate noise from the resulting foreground mask. Further, a multiple-objects tracking algorithm based on Kalman filter is implemented to associate the detections of the same object across frames generating object tubes. In this paper, we have paid more attention on the grouping and optimal rearrangement of tubes rather than tube extraction.

### 3.2 Recursive tube-grouping algorithm

Though preserving original interactions between moving objects is an important aspect while condensing a video, most video synopsis methods focus on preserving the chronological order of moving objects or reducing their collisions. Hence, during tube rearrangement optimization, original interactions may get altered while reducing false collisions. We have proposed a recursive tube-grouping algorithm to identify the interacting tubes in a video, and group those with strong spatio-temporal interactions together in the same tube set.

#### 3.2.1 Advantages

The motivation of our proposed recursive tube-grouping approach is illustrated in Fig. 3. The illustration intuitively shows how our proposed approach preserves the behavior interactions of the tubes, irrespective of the order of their appearance, as compared to an existing unidirectional grouping method [28]. To discover the relationships between tubes for preserving interactions, a group-partition algorithm has been proposed by Li et al. [28], which sequentially process each tube and group the related tubes. However, this approach is unidirectional based on the incremental start-times of tubes and cannot ensure the aggregation of all related tubes into a single group. The group-partition algorithm in [28] process tubes sequentially according to their start-times. The tubes depicted in the various scenarios of



**Fig. 3** A schematic illustration of comparison between tube-grouping by [28] and our proposed recursive tube-grouping method. **a** Mutually interacting tubes 1 to 4 that are close in spatio-temporal domain of original video. **b** Tube-grouping by existing group-partition method [28]. **c** Tube-grouping by our proposed approach. **d** Mutually interacting tubes 1, 3 and 2, 4 with a similar time of occurrence appear in an alternate order in the original video. **e** Existing unidirectional tube-grouping method [28] process tubes sequentially and subsequently, aggregate the tubes into two sets. **f** Our recursive tube-grouping approach groups all related tubes into a single tube set. **g** During synopsis generation, the temporal shift of both sets may alter the original interactions between tubes 1 to 4. **h** Our approach preserves the behavior interactions between tubes even when the tube set is shifted along time axis in the synopsis video

Fig. 3, possess strong spatial proximity and temporal overlap, which indicates interaction between them. In Fig. 3a, four interacting tubes 1, 2, 3 and 4 that start simultaneously in the original video are presented. Both the existing unidirectional grouping method [28] and our proposed approach group the mutually interacting tubes 1 to 4 into a single set as shown in Fig. 3b and c, respectively. In another scenario represented by Figs. 3d–h, tubes 1 to 4 appear in an alternative order in the original video with nearly same start-times. In Fig. 3e, according to [28], tubes 1 and 3 forms one set and tubes 2 and 4 forms another set. Our recursive tube-grouping approach aggregates all related tubes 1-4 into a single tube set as presented in Fig. 3f. During synopsis generation, the two tube groups generated by the existing approach will be positioned somewhere along the time axis according to their duration and start-times as shown in Fig. 3g. Thus, the original interactions between tubes 1 to 4 are altered. However, by grouping all interacting tubes into a single set, our proposed approach preserves the behavior interactions even when the tube set is shifted temporally in the synopsis video as shown in Fig. 3h. The proposed recursive tube-grouping algorithm groups the interacting tubes in the original video to form a tube set. Hence, a set of tubes become the basic processing unit instead of a single tube in our proposed approach.



### 3.2.2 Implementation details

The detailed flow of our proposed tube-grouping process is shown in Algorithm 1. The recursive tube-grouping algorithm begins by initializing two logical sets  $Q$  and  $G$ . The set  $Q$  contains all tubes in the original video and  $G$  is the set of tubes that are assigned to any one of the tube sets created by our approach. The tubes in  $Q$  are processed one by one. If a tube  $T_i$  is not a member of  $G$ , a new tube set is created and  $T_i$  is assigned with the new tube set label. Then,  $T_i$  is added to the global set  $G$ . Next, the procedure *TUBEVISIT* is called to determine the related tubes for each newly fetched tube  $T_i$ . Each tube  $T_j$  in  $Q$ , but not included in  $G$ , is initialized as *not visited*. If there exists any tube  $T_j$  such that there is a temporal overlap between  $T_i$  and  $T_j$ , we measure the spatial proximity between them. If tube  $T_j$  is not assigned to any tube set and has not been visited before in any frame of  $T_i$ , the tube is immediately marked as *visited*. Then, the spatial proximity between  $T_i$  and  $T_j$  is defined as

$$D(T_i, T_j) = \begin{cases} \exp\left(\frac{-\min\{dist_f(T_i, T_j)\}}{\vartheta_f}\right), & \text{if } T_i^f \cap T_j^f \neq \Phi \\ \infty, & \text{otherwise} \end{cases} \quad (1)$$

---

**Algorithm 1** Recursive tube-grouping algorithm.

---

```

1: Input: A set  $Q$  with all  $M$  tubes in original video,  $Q = \{T_1, T_2, \dots, T_M\}$ 
2: Output: Set  $Q$  with  $T_i.tubeset$  labeled,  $\forall T_i \in Q$ 
3: Initialization:  $G = \Phi$ 
4: for each tube  $T_i$  in  $Q$  do
5:   if  $T_i \notin G$  then
6:     Create a new tube set  $g_i$ 
7:      $T_i.tubeset = g_i$ 
8:      $G = G \cup T_i$ 
9:     TUBEVISIT ( $T_i$ )
10:  end if
11: end for
12: function TUBEVISIT  $\{T_i\}$ 
13:  $T_j.visited = false, \forall T_j \notin G$ 
14: for each frame  $f$  in which  $T_i$  appears do
15:   for each tube  $T_{j(j \neq i)}$  appearing in  $f$  do
16:     if  $T_j \notin G$  and  $T_j.visited = false$  then
17:        $T_j.visited = true$ 
18:       Compute  $D(T_i, T_j)$ 
19:       if  $D < \theta_d$  then
20:          $T_j.tubeset = g_i$ 
21:          $G = G \cup T_j$ 
22:         TUBEVISIT ( $T_j$ )
23:       end if
24:     end if
25:   end for
26: end for
27: end function

```

---

where  $dist_f(T_i, T_j)$  is the Euclidean distance between the centroids of  $T_i$  and  $T_j$  at every commonly shared frame  $f$ .  $\vartheta_f$  represents the average size of objects  $T_i$  and  $T_j$  in frame  $f$ .  $T_i^f \cap T_j^f$  denotes the temporal overlaps between  $T_i$  and  $T_j$ . If  $T_i$  and  $T_j$  do not share any common frame,  $D(T_i, T_j)$  is infinity. If the minimum distance between tubes  $T_i$  and  $T_j$  is less than a spatio-temporal threshold  $\theta_d$ ,  $T_j$  will be assigned with the same tube set label of  $T_i$  and  $T_j$  is added to  $G$  also.

The interaction relationship between object tubes depends on the threshold parameter. Since the pixel distance  $dist_f$  does not represent the actual distance between objects, we need to consider the object size also while determining the threshold  $\theta_d$ . Moreover, most of the surveillance videos contain both pedestrians and vehicles where the interaction can occur between two pedestrians, two vehicles or a pedestrian and a vehicle. Therefore, taking either object height or width alone into consideration for determining threshold will not be sufficient. Hence, we adopt a dynamic threshold computing in our approach for which the area of temporally intersecting objects are considered which varies depending on the object size. The threshold is adaptively computed for every pair of tubes in each frame of the original video as

$$\theta_d = \exp\left(\frac{-\{A(T_i) * A(T_j)\}}{dist_f^2(T_i, T_j)}\right) \quad (2)$$

where  $dist_f(T_i, T_j)$  denotes the Euclidean distance between the centroids of  $T_i$  and  $T_j$ ,  $A(T_i)$  and  $A(T_j)$  represent the area of tubes  $T_i$  and  $T_j$  at every commonly shared frame  $f$ .

Further, procedure *TUBEVISIT* is recursively called to find the related tubes of  $T_j$ . Thus, every tube related to  $T_i$ , but not a member of any tube set, are recursively called and processed. The aforementioned steps are repeated until all the tubes in  $Q$  are processed and assigned with a tube set label. Besides determining the behavior interactions between tubes, the recursive tube-grouping approach also alleviates the tracking and segmentation issues by aggregating the over-segmented or occluded tubes together to form same tube sets. The computational complexity of the proposed tube-grouping algorithm is  $O(M^2F)$ , where  $M$  is the number of tubes and  $F$  denotes the number of frames in the original video.

### 3.3 Spatio-temporal cube voting algorithm

After discovering the relationships between tubes and grouping them accordingly into tube sets, the remaining main process is to find the optimal start-times of tube sets to rearrange them and generate a condensed video. To obtain a condensed video with minimum false interactions, we may have to shift the tubes such that the object trajectories should not overlap in the spatio-temporal domain except for overlaps in the original video. However, shifting the tubes spatially can alter the original interactions between them and hence, we focus on shifting the tubes in the temporal domain only.

To optimally rearrange the tube sets for preserving tube interactions, we propose a spatio-temporal cube voting algorithm. Since related tubes are grouped together to form tube sets, a tube set will become the basic unit for further processing. In our approach, we represent the synopsis video as a 3-D spatio-temporal cubic volume  $V(s_x, s_y, f)$ , divided into several space-time cubes, where  $(s_x, s_y)$  represents the spatial location of a pixel at frame  $f$ , satisfying  $1 \leq s_x \leq W$ ;  $1 \leq s_y \leq H$ ;  $1 \leq f \leq F$ .  $W$  and  $H$  represents the width and height of input video frames, respectively.  $F$  is the

total number of frames initialized for the synopsis video. The duration of synopsis video  $F$  is initialized with the duration determined by the length estimation method detailed in Section 3.5.

### 3.3.1 Advantages

Most of the state-of-the-art methods [13, 27, 32, 33, 39, 51] in video synopsis formulate the tube rearrangement process as a constrained energy minimization problem [22, 23, 46, 49]. However, the optimization process is time-consuming and computationally intensive since it requires redundant computation of activity, collision and temporal costs iteratively during the minimization of energy functions. Unlike other energy minimization approaches [27, 33, 39], the proposed cube voting method computes the collision relationship between tubes only once for each tube set. Moreover, the result of optimization methods will be a trade-off among the baseline standards mentioned in Section 1, which does not ensure the maintenance of original interactions between tubes.

The main objective of the proposed spatio-temporal cube voting algorithm is to maintain the relative interactions between tubes by preserving the true collisions and reducing false collisions in the synopsis video. Given a video, if more than one tube set rearrangement solutions exists that preserves the relative temporal order of moving objects, the position labels which avoids false collision to the maximum extent will be selected for synopsis generation. Following a tube set arrangement solution, each tube set is filled into the volume  $V$  according to the temporal location defined by the solution with already known spatial location from the original video. Subsequently, it is able to know which all space-time cubes will be covered by the tubes with the aforementioned placement. Here, our major objective is to minimize the number of trajectory occurrences in each cube using a method of voting by space-time cubes. Moreover, since each grouped tube set is filled into  $V$  as a whole, this approach preserves the original behavioral interactions between tubes without altering their sequence of interaction.

### 3.3.2 Implementation details

The detailed flow of the proposed spatio-temporal cube voting method is given in Algorithm 2. The key objective of this algorithm is to find out an optimal starting position for each tube set so that multiple tube trajectories that cover the same spatio-temporal cube in  $V$  simultaneously are minimized, when placed at those optimal temporal locations. There will be several possible temporal positions for the placement of each tube sets with the utmost acceptable position computed as

$$p_{max} = F - length(tuberset) + 1 \quad (3)$$

For positions beyond  $p_{max}$ , tube sets cannot be completely covered within the defined synopsis duration. We denote each possible position  $p_i$ , for a tube set  $g_i$ , as a (tube set  $g_i$ , position  $p_i$ ) solution pair. When the positioning of  $g_i$  at  $p_i$  covers a cube in  $V$ , the cube will vote for this pair. To begin with the voting procedure, each cube in  $V$  is initialized with a (tube set, position) pair matrix  $Z$  of size  $N \times F$ , where  $N$  is the total tube sets.  $Z$  is initialized with zero votes for every (tube set, position) pair.

**Algorithm 2** Spatio-temporal cube voting algorithm.

---

```

1: Input: Set of  $N$  tube sets,  $G = \{g_1, g_2, \dots, g_N\}$ 
2: Output: Set of start-time labels for tube sets  $L = \{l_1, l_2, \dots, l_N\}$ ;
   Set of tube sets  $A$ , with start-time labels assigned
3: Initialization:  $L \leftarrow \Phi$ 
4: (Tube set, position) pair matrix  $Z_{m \times k} \leftarrow 0$ ,
    $\forall m \in \{1, \dots, N\}, k \in \{1, \dots, F\}$ 
5:  $C_i \leftarrow Z_{m \times k}, \forall$  cube  $C_i \in V$ 
6: Weight matrix  $WT(g_i, p_j) \leftarrow 0, \forall$  tube set  $i \in \{1, \dots, N\}$ , position  $j \in \{1, \dots, F\}$ 

7: // Stage-1 Voting
8: for each tube set  $g_i$  in  $\{G - A\}$  do
9:    $\max = F - \text{length}(g_i) + 1$ 
10:  for each possible position  $\{p_i\}_{i=1}^{\max}$  of  $g_i$  do
11:    for each cube  $C_i$  in  $V$  do
12:      if  $C_i$  is covered by  $g_i$  placed at  $p_i$  then
13:         $C_i\{Z(g_i, p_i)\} \leftarrow 1$ 
14:      end if
15:    end for
16:  end for
17: end for
18: // Stage-2 Voting
19: while  $A! = G$  do
20:  for each cube  $C_i$  in  $V$  do
21:     $\{g_{cov}, p_{cov}\} \leftarrow$  retrieve all  $(g_i, p_i)$  pairs which can cover  $C_i$ , where  $g_i \notin A$ 
22:    if  $C_i$  is not covered by any  $g_i \in A$  then
23:      // Weighted Voting
24:       $\alpha = \frac{1}{\#\{g_{cov}, p_{cov}\} \text{ pairs}}$ 
25:       $WT(g_i, p_i) \leftarrow WT(g_i, p_i) + \alpha, \forall (g_i, p_i) \in \{g_{cov}, p_{cov}\}$ 
26:    else
27:      // Penalty Voting
28:       $\beta =$  number of times  $C_i$  was previously covered
29:       $WT(g_i, p_i) \leftarrow WT(g_i, p_i) - \beta, \forall (g_i, p_i) \in \{g_{cov}, p_{cov}\}$ 
30:    end if
31:  end for
32:   $(g_i, p_i) \leftarrow \arg \max_{(g_i, p_i)} WT$  for  $g_i \notin A$ 
33:   $l_i \leftarrow p_i$ 
34:   $L = L \cup l_i$ 
35:   $A = A \cup g_i$ 
36: end while

```

---

The algorithm begins by initializing logical sets:  $G$  containing all  $N$  tube sets,  $L$  for the start-time labels of tubes sets and  $A$  for the tube sets which are already been assigned with a new start-time label by the cube-voting method. The overall voting procedure consists of two stages. In the first stage, the cubes that will be covered by the positioning of all tube sets at their possible temporal locations are determined. This begins by fetching out each tube set  $g_i$  in  $G$ , but not in  $A$ , one by one for processing. Then, all possible temporal locations for the placement of  $g_i$  is determined. For each possible position  $p_i$ , we will identify the

cubes that will be covered by the positioning of  $g_i$  at  $p_i$ . If  $g_i$  covers  $C_i$  when placed at  $p_i$ , then  $(g_i, p_i)$  pair will update  $Z(g_i, p_i)$  at  $C_i$  as 1.

The second stage implements a voting method where a spatio-temporal cube votes for all (tube set, position) pairs which can possibly cover it. If a cube  $C_i$  can be covered by more than one  $(g_i, p_i)$  pair,  $C_i$  will vote to all such candidate pairs by distributing the votes equally among them. For example, suppose, a cube can be covered by 8 possible (tube set, position) solutions corresponding to 3 different tube sets. Then, that specific cube votes 1/8 to each 8 solutions. Similarly, a  $(g_i, p_i)$  pair will receive votes from several cubes in  $V$  when the positioning of  $g_i$  at  $p_i$  covers multiple cubes. The voting procedure by a cube  $C_i$  handles two situations

- 1) *Weighted Voting*: when  $C_i$  was not yet covered by the positioning of any  $g_i \in A$ ;
- 2) *Penalty Voting*: when  $C_i$  was already been covered once or more than once in the previous placements of any  $g_i \in A$ .

The second stage begins by initializing a weight matrix  $WT$  of size  $N \times F$ . For each cube  $C_i$  in  $V$ , all  $(g_i, p_i)$  pairs where  $g_i \notin A$  and which can cover  $C_i$  are denoted as  $\{g_{cov}, p_{cov}\}$ . Next, in weighted voting,  $C_i$  votes to each of the aforementioned  $(g_i, p_i) \in \{g_{cov}, p_{cov}\}$  by incrementing their already received votes from other  $C_j, j \neq i$  as follows

$$\alpha = \frac{1}{\text{number of}\{g_{cov}, p_{cov}\}\text{pairs}} \tag{4}$$

$$WT(g_i, p_i) \leftarrow WT(g_i, p_i) + \alpha \tag{5}$$

With respect to the second situation, previously covered  $C_i$  participates further in the voting procedure by voting with a penalty to each  $(g_i, p_i) \in \{g_{cov}, p_{cov}\}$  for reducing the chance of placement of  $g_i$  at  $p_i$ . Subsequently, the penalty voting ensures minimal sharing of  $C_i$  among multiple candidate sets which in turn reduces false collision. We define the negative penalty as the number of times  $C_i$  was covered in the previous positioning of tube sets. Thus, the solution  $(g_i, p_i)$  will be heavily penalized as the degree of collisions or cube overlaps increases. The vote for each such  $(g_i, p_i)$  is updated as

$$\beta = \text{number of times } C_i \text{ was covered} \tag{6}$$

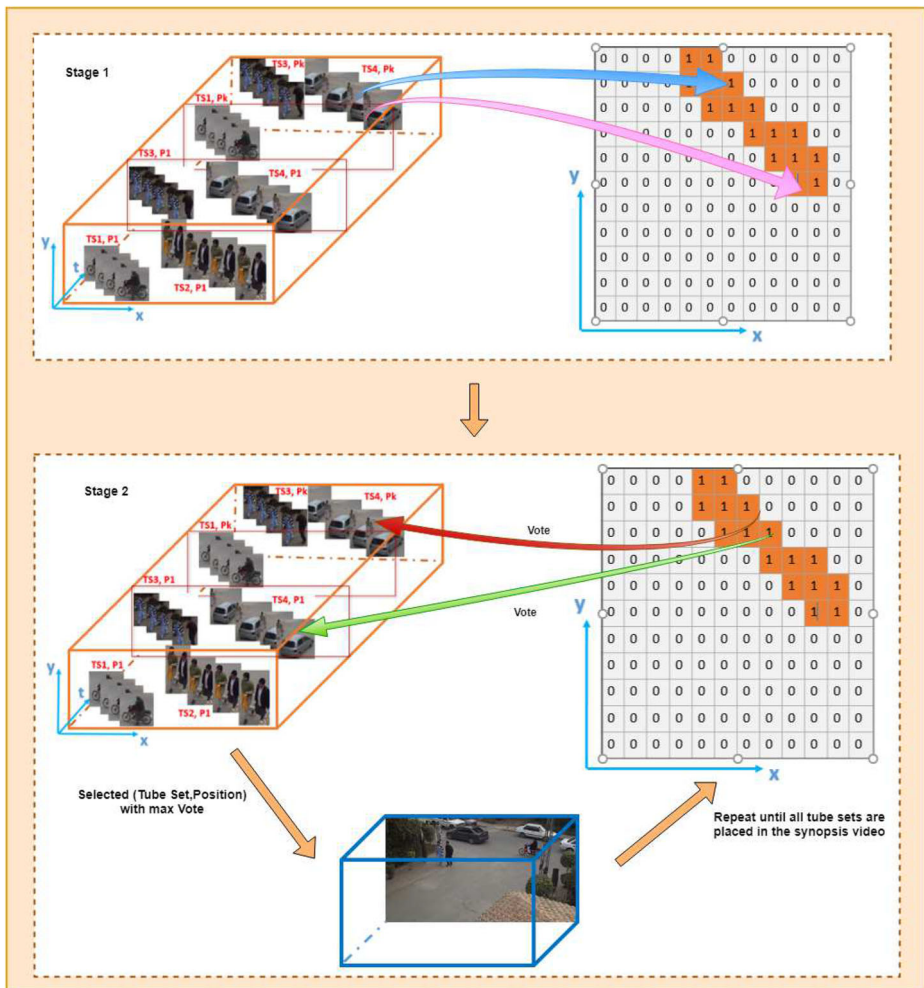
$$WT(g_i, p_i) \leftarrow WT(g_i, p_i) - \beta \tag{7}$$

Further, the total votes that each  $(g_i, p_i)$  solution pair receives from different cubes determine the selection of a (tube set, position) pair to be added into the synopsis video. The  $(g_i, p_i)$  solution which obtains the maximum vote in the weight matrix  $WT$ , will be the placement with minimum false collisions since the cubes covered by such a placement are shared minimally. Subsequently, the tube set  $g_i$  corresponding to the aforementioned solution is selected to be placed in the synopsis video immediately. Thus, the new start-time  $l_i$  of  $g_i$  in the synopsis video is assigned with  $p_i$  specified by the solution.

Since each tube set represents a set of mutually interacting tubes, we need to maintain the relative time interval between these tubes while positioning the tube sets in the final synopsis video. Hence, the tube in  $g_i$  with the least original start-time is positioned at temporal position  $p_i$ . In order to preserve the relationships between tubes, we label the successive start-times of tubes in  $g_i$  with respect to the first position  $p_i$ . Subsequently, our approach preserves the true collisions of the original video in the synopsis video as well. The tube set to which the optimal start-time is assigned will be added to  $A$  and further, it will not participate in the voting procedure. The second stage of voting is repeated until all the tube sets in  $G$  are assigned with an optimal start-time label. Thus, the process of selection of (tube

set, position) solution with maximum votes is continued and the corresponding tube sets are positioned in the synopsis video, one after the other.

Figure 4 illustrates the two stages of spatio-temporal cube voting algorithm with tube sets  $g_1, \dots, g_4$  and their acceptable temporal positions  $p_1, \dots, p_k$ . The jump arrows in stage 1 show the cubes/grids (orange colored) that will be covered by a tube set when positioned at any given temporal location. In stage 2, the jump arrows indicate the votes cast by each cube/grid to the (tube set, position) that covers it. The tracking and segmentation issues by aggregating the over-segmented or occluded tubes together to form same tube sets. The computational complexity of the proposed cube-voting algorithm is  $O(NPC)$ , where  $N$  denotes the total number of tube sets,  $P$  represents the possible temporal locations of tube sets, and  $C$  denotes the number of cubes in  $V$ .



**Fig. 4** Illustration of spatio-temporal cube voting method. The first stage updates all cubes (grids in the spatial domain) that will be covered by different tube sets at various positions. In the second stage, each cube votes for (tube set, position) pairs that pass through it. The tube set that gets a maximum vote is selected to add into the synopsis video and the process is repeated until all tube sets are placed in the synopsis video

### 3.4 Tube set stitching

In the final step of video synopsis, the background of the original video has to be extracted to form a background video with the same length as that of synopsis video. The background of each frame in the input video is computed using temporal medians over its neighboring frames for each one minute of the video. Each background pixel corresponds to the median of current frame i.e. 30 seconds frame before and after the current frame. A uniform temporal sampling of active periods is utilized to create background video from the background images. Finally, the tube sets are stitched into the background video according to the new start-time labels computed in the tube set positioning phase. Like most existing video synopsis methods, we employ Poisson image editing [36] to stitch tube sets into the background. Since stitching does not affect the accuracy of synopsis approach, no great attention has been paid on this step.

### 3.5 Synopsis length estimation

During the generation of synopsis video, the proposed recursive tube-grouping algorithm identifies the interacting tubes in the original video based on their spatio-temporal proximity, and groups the related tubes together to form tube sets. The proposed spatio-temporal cube voting algorithm optimally rearranges the tube sets in synopsis video. When the length of the synopsis video is less than that of any tube set, spatial and temporal relationships between tubes in that tube set cannot be maintained in synopsis as such in the original video. Therefore, the minimum length of the synopsis video should be the length of the longest tube set. This ensures the preservation of relative time intervals between tubes in all the tube sets.

The synopsis length that reduces false collision to the maximum possible extent is the sum of the duration of all tube sets. Thus, the duration (length) of synopsis can be defined in the boundary  $[length(g_{max}), max(length(O), \sum_{i=1}^N length(g_i))]$ , where  $g_i$  represents  $i^{th}$  tube set,  $g_{max}$  is the longest tube set and  $O$  denotes the original video. To minimize collisions in the synopsis video, we propose a method to estimate the duration based on an entropy measure of tube collisions.

The proposed method considers each frame of original video as a 2D grid ( $x$ - $y$  spatial plane). Then, the grids that will be covered corresponding to the original spatial positions of tube sets are analyzed. This projection of tube sets onto 2D spatial plane will indicate the probable collisions that can occur in the synopsis. For example, a grid shared by two or more tube sets indicates an area of false collision if those tube sets are shifted inappropriately along the time axis. However, sharing of a grid by tubes within a tube set shows true collisions that will be preserved in the synopsis by our tube set rearrangement method. For each grid  $X$ , the probability by which each tube set  $g_i$  covers it can be defined as

$$p(g_i) = \frac{f_i}{f_X} \quad (8)$$

where  $f_i$  represents the number of frames in which  $g_i$  pass through  $X$  and  $f_X$  denotes the total number of frames in which  $X$  is covered by any tube set. The entropy ( $H$ ) at each grid is the expected number of tube collisions determined across all frames in the original video

$$H_X = - \sum_{i=1}^N p(g_i) \log_2 p(g_i) \quad (9)$$

We compute the total entropy  $H_{tot}$  as

$$H_{tot} = \sum_{\forall grids} H_X \quad (10)$$

The synopsis duration will be an estimate based on  $H_{tot}$  and duration of the longest tube set. Thus, the length (L) of the synopsis video is estimated as

$$L = \begin{cases} length(g_{max}) * mean(H_{tot}), & \text{if } H_{tot} \neq 0 \\ length(g_{max}), & \text{otherwise} \end{cases} \quad (11)$$

## 4 Experiments

To evaluate the performance of the proposed approach, we have conducted extensive experiments, which are detailed in this section. The experiments are conducted in MATLAB R2018b on an Intel Core i5-6500 3.20-GHz processor with 8-GB of memory.

### 4.1 Datasets

We carried out the experiments on seven publicly available surveillance videos, whose characteristics are summarized in Table 2. We have selected these 7 videos by taking the scenarios that have behavioral interactions between objects into consideration. Some of the diverse scenarios covered by these videos are depicted in Fig. 5.

### 4.2 Evaluation metrics

In the following, we evaluate the proposed approach using 6 aspects: (1) frame condensation ratio, (2) total collision area, (3) total false collision area, (4) interactions violation ratio, (5) chronological disorder ratio, and (6) visual quality. The first five metrics are used for objective comparisons while the visual quality metric reflects the vision comfortable degree of users.

- 1) *Frame condensation ratio (FR)*: The *FR* is defined as the ratio between the number of frames in the synopsis video and original video. One of the qualitative standards commonly used in video synopsis requires the condensation ratio to be high.

**Table 2** Parameters of experimental surveillance videos: Video Number (Num), Video Name (Video), Resolution, Frame Rate (fps), Number of Frames (#Frames), Number of Tubes (#Tubes)

Num	Video	Resolution	fps	#Frames	#Tubes
1	Subway Station [6]	448 × 360	30	4470	120
2	Town-Centre [4]	1920 × 1080	25	7500	230
3	BEHAVE-1 [5]	640 × 480	25	14683	35
4	BEHAVE-2 [5]	640 × 480	25	6694	43
5	Street-UET [35]	1280 × 720	30	14730	194
6	CAVIAR-1 [11]	384 × 288	25	3700	16
7	CAVIAR-2 [11]	384 × 288	25	1650	9





**Fig. 5** Sample test scenarios. **a** Town-Centre, **b** Street-UET, **c** CAVIAR-1, **d** CAVIAR-2, **e** BEHAVE-1, **f** Subway Station, **g** BEHAVE-2

2) *Total collision area (CA)*: The total collision area between two tubes  $i$  and  $j$  is defined as

$$CA(i, j) = \sum_{f \in F_i \cap F_j} I(\text{box}(i^f) \cap \text{box}(j^f)) \tag{12}$$

where  $F_i \cap F_j$  denotes the common frames between  $i$  and  $j$  and  $I(\text{box}(i^f) \cap \text{box}(j^f))$  is the intersecting area between bounding boxes of tubes  $i$  and  $j$  in frame  $f$ .

3) *Total false collision area (TFCA)*: The *TFCA* measures the total area of false tube collisions in the synopsis video. The false collision area (*FCA*) between two tubes  $i$  and  $j$  is defined as follows

$$FCA(i, j) = \begin{cases} CA_s(i, j) - CA_o(i, j), & \text{if } t_i^s - t_j^s = t_i^s - t_j^s \\ CA_s(i, j), & \text{otherwise} \end{cases} \tag{13}$$

where  $CA_o(i, j)$  and  $CA_s(i, j)$  are the total collision area between tubes  $i$  and  $j$  in the original video and synopsis video, respectively.  $t_i^s$  and  $t_j^s$  denote the start-times of tubes  $i$  and  $j$  in the original video. Similarly,  $t_i^s$  and  $t_j^s$  represent the start-times of tubes  $i$  and  $j$  in the synopsis video.

4) *Interactions violation ratio (IVR)*: *IVR* measures the degree of original behavioral interactions not preserved in the synopsis video. It is defined as the ratio between the number of tube pairs where interaction is violated in synopsis video and total number of interacting tube pairs in the original video. A higher *IVR* score means greater number of altered original interactions, and a synopsis video with all interactions preserved will achieve an *IVR* score of 0.

5) *Chronological disorder ratio (CDR)*: *CDR* measures the degree of violation in the chronological order of tubes. It is computed as the ratio between the number of pairs of tubes whose chronological orders are reversed in the synopsis video when compared to the original video. A higher *CDR* indicates a greater violation of chronological order.

**Table 3** Performance comparison with (G) or without recursive tube-grouping algorithm. IVR - Interactions Violation Ratio, TFCA - Total False Collision Area

Video Num	IVR	IVR (G)	TFCA	TFCA (G)
1	0.82	0	$5.79 \times 10^6$	$1.06 \times 10^6$
2	0.21	0	$1.74 \times 10^7$	$1.94 \times 10^7$
3	0.53	0	$2.43 \times 10^7$	$5.24 \times 10^6$
4	0.67	0	$4.02 \times 10^7$	$2.67 \times 10^6$
5	0.33	0	$1.03 \times 10^7$	$3.90 \times 10^5$
6	0.75	0	$6.56 \times 10^6$	$6.58 \times 10^5$
7	0.88	0	$2.75 \times 10^6$	$1.03 \times 10^5$

### 4.3 Performance comparison

We compare our proposed framework with a classical synopsis method [39], a scaling down method to minimize collisions [27], and a group-partition approach to deal with crowded scenes [28]. The comparison of proposed synopsis framework with the previous video synopsis methods are discussed in Section 4.3.4. In addition, evaluation of the



**Fig. 6** The first two columns show the representative frames with interactions (marked in yellow and red color ellipses) in original videos. The third and fourth columns represent corresponding results generated without recursive tube-grouping. The last column shows the synopsis results generated with recursive tube-grouping. Video sequences **a–e** Street-UET, **f–j** BEHAVE-2, **k–o** Town-Centre

**Table 4** Objective comparison of the proposed spatio-temporal cube voting approach and state-of-the-art video synopsis optimization methods [39], [27], [28]. Num - Video Number, TCA - Total True Collision Area, TFCA - Total False Collision Area, CA - Total Collision Area

Num	TCA	TFCA [39]	TFCA [27]	TFCA [28]	TFCA [Our]	CA [39]	CA [27]	CA [28]	CA [Our]
1	$3.62 \times 10^6$	$3.04 \times 10^6$	$1.43 \times 10^7$	$2.14 \times 10^6$	$1.06 \times 10^6$	$4.66 \times 10^6$	$1.43 \times 10^7$	$4.89 \times 10^6$	$4.68 \times 10^6$
2	$1.00 \times 10^7$	$2.28 \times 10^7$	$3.62 \times 10^7$	$2.46 \times 10^7$	$1.94 \times 10^7$	$3.34 \times 10^7$	$3.63 \times 10^7$	$3.03 \times 10^7$	$2.94 \times 10^7$
3	$5.19 \times 10^7$	$7.36 \times 10^6$	$5.25 \times 10^7$	$6.17 \times 10^6$	$5.24 \times 10^6$	$5.92 \times 10^7$	$6.18 \times 10^7$	$5.84 \times 10^7$	$5.72 \times 10^7$
4	$2.41 \times 10^7$	$5.89 \times 10^6$	$3.94 \times 10^7$	$4.28 \times 10^6$	$2.67 \times 10^6$	$3.17 \times 10^7$	$3.94 \times 10^7$	$3.39 \times 10^7$	$2.67 \times 10^7$
5	$7.50 \times 10^6$	$4.21 \times 10^6$	$1.41 \times 10^7$	$4.07 \times 10^6$	$3.90 \times 10^5$	$7.14 \times 10^6$	$1.41 \times 10^7$	$7.94 \times 10^6$	$7.89 \times 10^6$
6	$3.22 \times 10^6$	$3.28 \times 10^6$	$7.48 \times 10^6$	$3.73 \times 10^6$	$6.58 \times 10^5$	$5.19 \times 10^6$	$7.48 \times 10^6$	$5.01 \times 10^6$	$3.88 \times 10^6$
7	$2.41 \times 10^6$	$3.46 \times 10^6$	$5.63 \times 10^6$	$2.39 \times 10^5$	$1.03 \times 10^5$	$4.05 \times 10^6$	$5.63 \times 10^6$	$3.17 \times 10^6$	$2.51 \times 10^6$

**Table 5** Objective comparison between synopsis results generated using proposed entropy-based length method (1) and results with length equal to that of longest tube (2). Num - Video Number, CA - Total Collision Area, FR - Frame Condensation Ratio

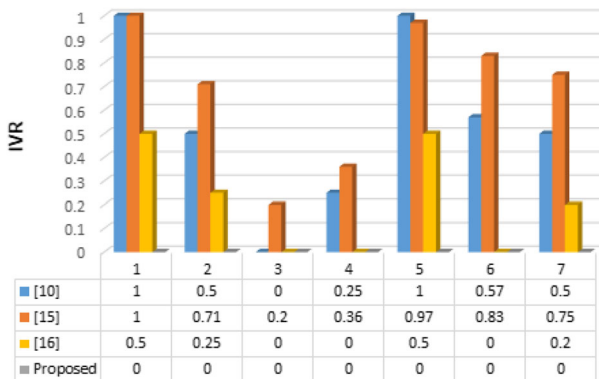
Num	CA (1)	CA (2)	FR (1)	FR (2)
1	$4.68 \times 10^6$	$5.79 \times 10^6$	0.468	0.289
2	$2.94 \times 10^7$	$1.74 \times 10^7$	0.307	0.287
3	$5.72 \times 10^7$	$5.72 \times 10^7$	0.385	0.385
4	$2.67 \times 10^7$	$4.23 \times 10^7$	0.491	0.342
5	$7.89 \times 10^6$	$1.03 \times 10^7$	0.280	0.179
6	$3.88 \times 10^6$	$6.56 \times 10^6$	0.432	0.281
7	$2.51 \times 10^6$	$2.75 \times 10^6$	0.636	0.418

individual components in our proposed framework are discussed in the following three subsections.

### 4.3.1 Evaluation of recursive tube-grouping algorithm

To validate the effectiveness of the recursive tube-grouping algorithm, we evaluate the synopsis videos generated for seven surveillance videos by our proposed method, with (G) and without recursive tube-grouping. The comparative results are shown in Table 3. An IVR score of 0 indicates that there is no violation of original interactions in the synopsis video. The results generated with tube-grouping achieve IVR score as 0 for all seven videos, validating the preservation of tube interactions by the proposed recursive tube-grouping algorithm. Similarly, TFCA score of our approach with grouping is significantly lesser compared to the results generated without tube-grouping in almost all test videos. This is caused by the fact that with the restriction of grouping, tubes have limited options for the placement in synopsis video, minimizing the chances of false collisions.

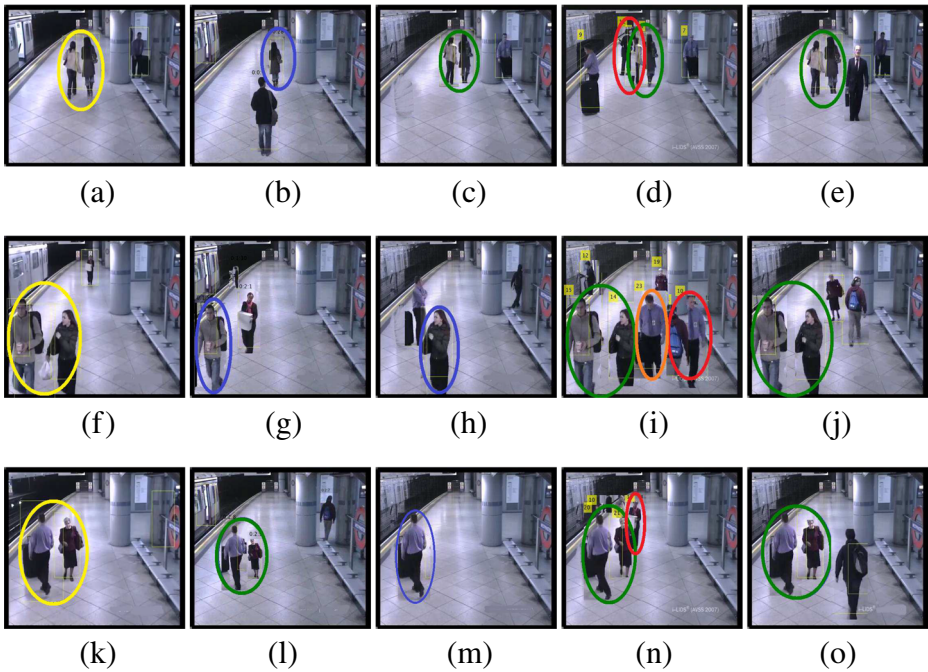
Some representative synopsis results generated with and without recursive tube-grouping algorithm are shown in Fig. 6, where interactions in small groups are highlighted within



**Fig. 7** Comparison of Interactions Violation Ratio (IVR) between state-of-the-art video synopsis methods [39], [27], [28] and proposed framework on 7 test videos

**Table 6** Objective comparison between previous video synopsis methods [39], [27], [28] and proposed framework. Num - Video Number, TFCA - Total False Collision Area, CDR - Chronological Disorder Ratio

Num	TFCA [39]	TFCA [27]	TFCA [28]	TFCA [Our]	CDR [39]	CDR [27]	CDR [28]	CDR [Our]
1	$3.04 \times 10^6$	$1.43 \times 10^7$	$2.14 \times 10^6$	$1.06 \times 10^6$	2.532	1.167	2.533	1.171
2	$2.28 \times 10^7$	$3.62 \times 10^7$	$2.46 \times 10^7$	$1.94 \times 10^7$	6.8286	3.5571	6.6	6.0571
3	$7.36 \times 10^6$	$5.25 \times 10^7$	$6.17 \times 10^6$	$5.24 \times 10^6$	1.739	0.4444	1	1
4	$5.89 \times 10^6$	$3.94 \times 10^7$	$4.28 \times 10^6$	$2.67 \times 10^6$	2.7	1	1.7	2.1
5	$4.21 \times 10^6$	$1.41 \times 10^7$	$4.07 \times 10^6$	$3.90 \times 10^5$	3.8182	2.1711	3.532	3.254
6	$3.28 \times 10^6$	$7.48 \times 10^6$	$3.73 \times 10^6$	$6.58 \times 10^5$	1.6333	0.23	1.271	1.04
7	$3.46 \times 10^6$	$5.63 \times 10^6$	$2.39 \times 10^5$	$1.03 \times 10^5$	3.21	1.714	2.63	2.429

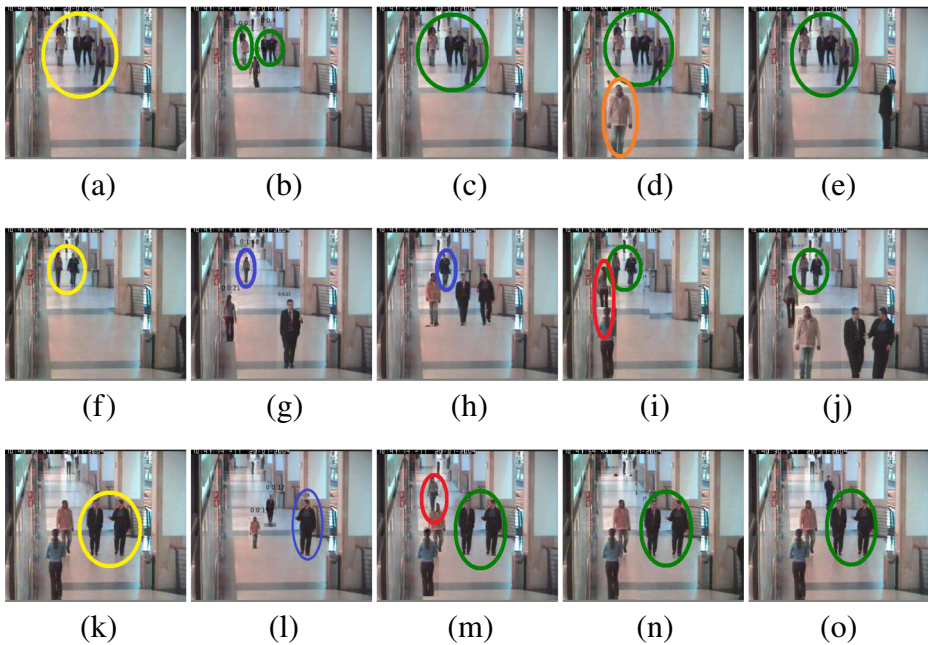


**Fig. 8** Comparison on the subway station sequence. From left to right: the first column shows frames of the original video. The second, third and fourth columns represent synopsis results of [27], [39] and [28], respectively. The fifth column shows the synopsis results of our proposed method

ellipses. First row represents scenarios from Street-UET video sequence, where two spatially close persons are walking together as shown in Figs. 6a and b. In the corresponding synopsis generation without tube-grouping, each of the two persons is considered as separate tubes while positioning them in the synopsis video (see Fig. 6c and d). As shown in Fig. 6e, synopsis generation with recursive tube-grouping preserves both original interactions. The second row presents scenarios from BEHAVE-2 video sequence, where two persons talk with each other while walking together. In synopsis results without tube-grouping, unrealistic scenes are created where the split tubes of a man and woman seem to walk with the gestures resembling a conversation (Figs. 6h and i). On the contrary, synopsis results with our tube-grouping approach well preserves the original activities as shown in Fig. 6j. Similarly, the last row from Town-Centre sequence shows scenes (Figs. 6k and l) where two people meet and walk together. Without tube-grouping, the persons are separated and displayed individually in the synopsis video. However, our approach effectively displays the people together without altering their original interactions.

#### 4.3.2 Evaluation of spatio-temporal cube voting algorithm

Reduction of collision between objects is one of the key factors [39] that should be considered during the generation of a synopsis video. The main objective of the proposed spatio-temporal cube voting method is to determine the optimal start-times for tube sets in the synopsis video. The metric TFCA, measures the total false collision area introduced in the synopsis video due to rearrangement of tubes. Therefore, the effectiveness of the proposed cube voting tube set rearrangement approach can be evaluated only by



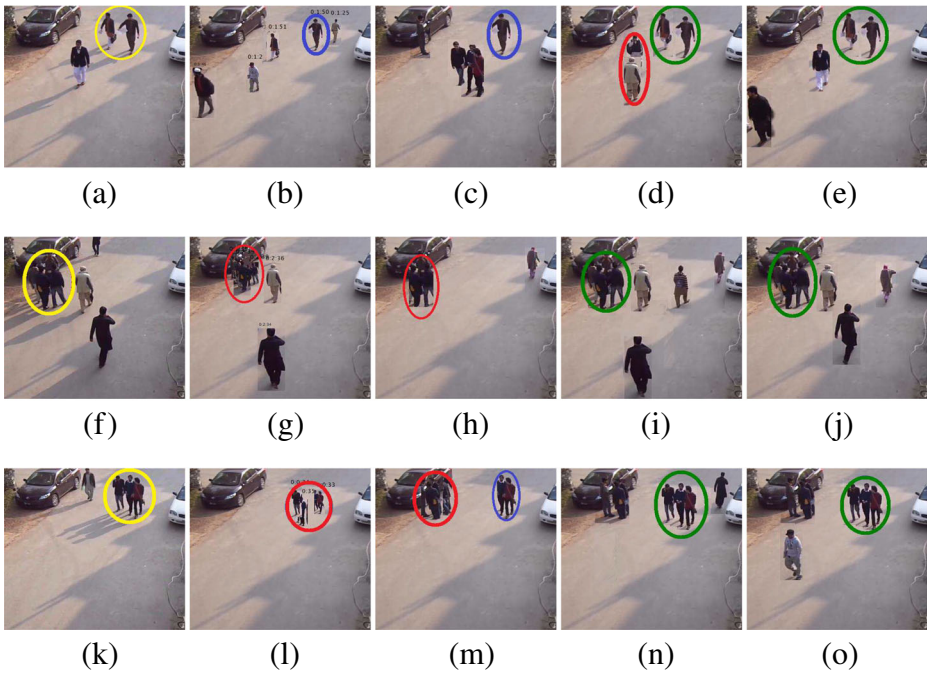
**Fig. 9** Comparison on the CAVIAR-1 sequence. From left to right: the first column shows frames of the original video. The second, third and fourth columns represent synopsis results of [27], [39] and [28], respectively. The fifth column shows the synopsis results of our proposed method

comparing the result of proposed approach with the results of state-of-the-art synopsis methods that comprise different tube rearrangement optimization approaches. Table 4 shows the objective comparison results of the proposed spatio-temporal cube voting approach and state-of-the-art video synopsis optimization methods [39], [27], [28].

From Table 4, we can see that our approach achieves less TFCA than those of the previous synopsis methods [39], [27] and [28], reduced by more than 7, 19 and 6 times, respectively. For any given input video, when TFCA becomes nearly equal to CA, we can conclude that the original interactions preserved in the corresponding synopsis video will be approximately zero. The TFCA achieved by our proposed method is significantly lesser than that of [39], [27] and [28] even for videos, where there is not much difference between the CA score of our approach and that of the other three synopsis methods. For our approach, the difference between TFCA and CA is almost the same as the total true collision area (TCA) for all seven videos. Thus, the experimental results demonstrate that the proposed spatio-temporal cube voting method creates fewer false collisions in the synopsis videos, no matter which types of scenarios, compared with the conventional synopsis methods.

### 4.3.3 Evaluation of length estimation

Most of the state-of-the-art video synopsis approaches [14, 27, 39] employ synopsis length as user-specified or same as the length of the longest tube in the synopsis video. The length ( $L$ ) of a synopsis video is counted by frames. The length estimation of the proposed framework is evaluated by experimenting with our proposed entropy-based length estimation (1) and state-of-the-art longest tube length method (2). The results are compared using the metrics CA and FR as shown in Table 5.



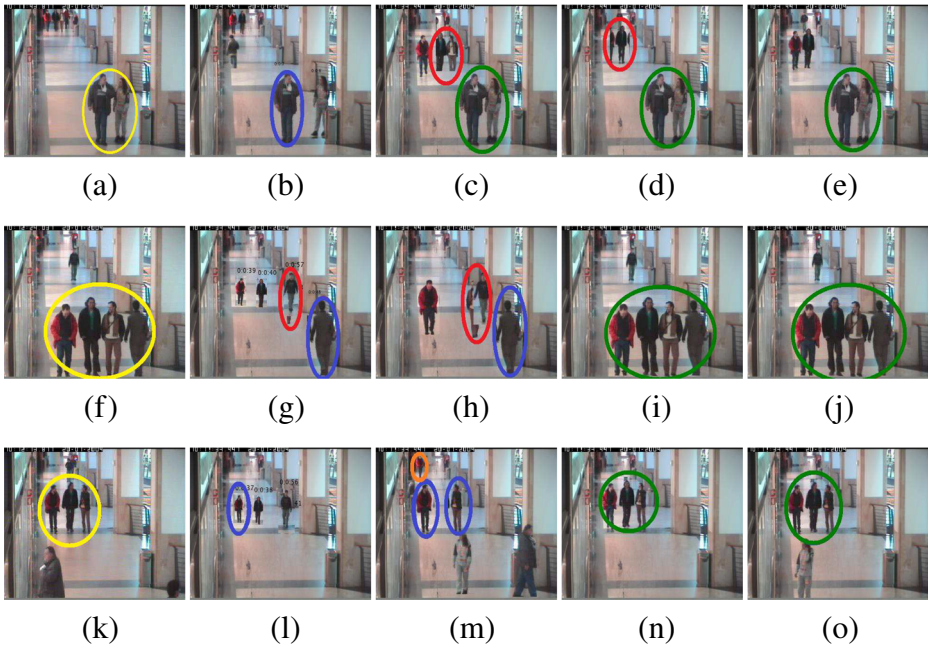
**Fig. 10** Comparison on the Street-UET sequence. From left to right: the first column shows frames of the original video. The second, third and fourth columns represent synopsis results of [27], [39] and [28], respectively. The fifth column shows the synopsis results of our proposed method

We find that our approach (1) achieves better scores for total collision area (CA) compared to the CA scores of results generated by taking length of longest tube (2). The proposed method is based on the entropy measure of tube collisions in the original video. In this way, the probable collisions are taken into account during the estimation of synopsis length, which significantly reduces the total collision in synopsis video. In contrast, the condensation ratio (FR) of synopsis videos by (2) obtains better scores compared to the FR values of our approach (1). This is due to the fact that length of the longest tube in a video will be generally lesser than the entropy-based estimated length that takes tube collisions and duration of longest tube set into consideration.

#### 4.3.4 Comparison of the proposed framework with previous approaches

Figure 7 shows the comparison results of IVR between existing video synopsis methods [27, 28, 39], and the proposed framework using 7 test videos. Since the proposed method preserves all tube interactions in the input video, an IVR score of 0 is obtained for all videos. The optimization process in [39] does not consider object relationships while computing the optimal start-times for tubes in synopsis video, which resulted in IVR scores greater than zero for most of the test videos. The scaling down method in [27] achieves higher IVR scores than the synopsis method in [39], as the reduction in object sizes may alter the spatio-temporal interactions between objects. The method proposed in [28] preserves the tube interactions with IVR scores in the range of 0 to 0.5.





**Fig. 11** Comparison on the CAVIAR-2 sequence. From left to right: the first column shows frames of the original video. The second, third and fourth columns represent synopsis results of [27], [39] and [28], respectively. The fifth column shows the synopsis results of our proposed method

Table 6 shows the detailed comparison results between the proposed framework and three state-of-the-art video synopsis methods [27, 28, 39]. The results show that our proposed framework obtains remarkably less TFCA than the previous video synopsis methods [27, 28, 39]. Similarly, Fig. 7 shows an IVR score of 0 achieved by the proposed framework. Thus, results from the evaluation of individual components, Table 6 and Fig. 7 validated that the proposed framework preserves the original interactions between tubes and generates relatively fewer false collisions in the synopsis video.

However, video synopsis method in [27] achieves the lowest CDR values while our framework achieved the second-lowest CDR in the evaluation using CDR metric. This is due to the reason that the proposed approach focuses on maintaining the relative chronological order of tubes within each tube set in order to preserve the original interactions between related tubes. Meanwhile, the rearrangement of tube sets to determine optimal positions in synopsis may result in the violation of chronological order among tube sets. However, with or without breaking the chronological order, the objective of video synopsis is to find an optimal arrangement of tubes to generate a compact video.

**Visual quality comparison** To further analyze the effectiveness of the proposed approach, a visual quality comparison is made between the results of the proposed method and the synopsis methods in [39], [27], and [28]. The results on four test sequences are shown in Figs. 8, 9, 10 and 11. In each figure, the first column presents the representative frames from original videos with a group of interactions shown in an ellipse. The second, third, fourth, and fifth columns present corresponding synopsis results by [27], [39], [28] and proposed approach, respectively. Yellow ellipses denotes interactions in the original video.

Original interactions preserved in the synopsis results are represented with green ellipses, whereas interactions that are altered in the synopsis video are denoted with blue ellipses. Red ellipses represent false collisions and unpleasant visual effects created in the synopsis results. Orange ellipses represent multiple instances of the same object that are displayed in the same frame of synopsis video.

Figure 8 represents the synopsis results of subway station video sequence. In the second and third column of Fig. 8, we can see that the interacting people are separated and displayed independently by positioning them at different temporal locations. This indicates that [27] and [39] fail to maintain the relationships between objects in most cases. Both [27] and [39] maintain the original interactions in one of the three sequences, even though the results deviate slightly from original scenes. Interactions in three sequences are preserved by the method in [28] as shown in the fourth column of Fig. 8. However, a few false interactions are introduced and multiple observations of the same person, highlighted with orange ellipses are displayed on the same frame of results. Compared to [39], [27] and [28], the proposed approach preserves the spatio-temporal relationships between objects in all three scenarios with fewer false collisions as shown in the last column of Fig. 8.

The visual quality comparative results of CAVIAR-1 sequence are presented in Fig. 9. Each tube is tracked individually by [27] and [39] without considering about the spatio-temporal interactions between them. However, it can be seen from Fig. 9(b) that overlapping tubes are considered as a single tube while tracking. Consequently, they are displayed together (see yellow ellipses in Fig. 9(b)) in the synopsis video by [27] albeit scaling down to reduce collision. Though [39] preserved the group interactions in two of the three sequences and [28] preserved interactions in all three sequences, obvious object overlaps (false collisions) are visible. The proposed recursive tube-grouping and cube voting method aggregates the related single tubes, maintains the relative temporal order between them and displays with a minimum false collisions.

Figure 10 illustrates the visual comparison results of Street-UET video sequence. This video presents crowded interactions as shown in the second row of Fig. 10. For all overlapping objects, [27] and [39] have maintained the interactions although partially as presented in Figs. 10(f), (g), (j) and (k). However, we can see visual artifacts in those results due to segmentation errors and object collisions. From the comparison between results of [27], [39] and [28], we can see that our proposed approach reconstructs the original activities without any visual errors even for crowded scenes.

Figure 11 shows the comparative synopsis results of CAVIAR-2 video sequence. Along with the failure to maintain object relationships in most results, some foreground segmentation errors by [27] and [39] lead to unpleasant synopsis results as shown in Figs. 11c, f, and k. Furthermore, a few collisions can be seen in Figs. 11f and g. More visually pleasing synopsis by [28] with fewer false collisions than the other two previous methods, are shown in the fourth column of Fig. 11. In Fig. 11k, we can see multiple observations of a person with red colored jacket due to tracking error. Our proposed approach alleviates the aforementioned problems to reconstruct original interactions using individually tracked tubes as shown in the last column of Fig. 11.

## 5 Conclusion

In this paper, we investigate how to preserve the relationships between tubes in the original video during the generation of synopsis video. We propose a recursive tube-grouping algorithm to identify and group the mutually interacting tubes in the original video to form

tube sets based on their spatial and temporal proximity. Further, to optimally arrange these tube sets so as to retain the discovered interactions, a spatio-temporal cube voting algorithm is proposed. This algorithm aims to maintain the relative interactions between tubes within a tube set while reducing false tube collisions. Furthermore, we propose a length estimation method for synopsis video based on an entropy measure of tube collisions. We also demonstrate the effectiveness of our proposed video synopsis framework with extensive experimental results. In the future, we will address the various challenges in video indexing and retrieval applications.

**Acknowledgements** We would like to express our gratitude to Dr. Geetha M, Assistant Professor, Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, for her support and valuable suggestions that greatly improved the manuscript.

**Funding** The authors declare that this research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

**Availability of data and material** The data that support the findings of this study are available in [“Subway Station” (<https://ieeexplore.ieee.org/abstract/document/4123801>)], “Town-Centre” (<https://ieeexplore.ieee.org/document/5995667>), “BEHAVE-1” (<http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/>), “BEHAVE-2” (<http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/>), “Street-UET” ([https://youtu.be/2bKXv\\_XviFc](https://youtu.be/2bKXv_XviFc)), “CAVIAR-1” (<http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>), “CAVIAR-2” (<http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>)]

**Code availability** Not applicable.

## Compliance with Ethical Standards

**Conflict of interests** Due to confidentiality agreements with the research institution, supporting custom code can only be made available to researchers subject to a non-disclosure agreement on request to the corresponding author [Namitha K].

## References

1. Aarathi R, Amudha J, Boomika K, Varrier A (2016) Detection of moving objects in surveillance video by integrating bottom-up approach with knowledge base. *Procedia Computer Science* 78:160–164
2. 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*
3. Baskurt KB, Samet R (2019) Video synopsis: A survey. *Comput Vis Image Underst* 181:26–38
4. Benfold B, Reid I (2011) Stable multi-target tracking in real-time surveillance video. In: *CVPR*. IEEE, pp 3457–3464
5. Blunsden S, Fisher R (2010) The behave video dataset: ground truthed video for multi-person behavior classification. *Annals of the BMVA* 4(1–12):4
6. Branch HOSD (2006) Imagery library for intelligent detection systems (i-lids). In: 2006 IET Conference on Crime and Security. IET, pp 445–448
7. 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
8. Cirne MVM, Pedrini H (2018) Viscom: A robust video summarization approach using color co-occurrence matrices. *Multimed Tools Appl* 77(1):857–875
9. Dollár P, Appel R, Belongie S, Perona P (2014) Fast feature pyramids for object detection. *IEEE Trans Pattern Anal Mach Intell* 36(8):1532–1545
10. Feng S, Lei Z, Yi D, Li SZ (2012) Online content-aware video condensation. In: *IEEE Conference on computer vision and pattern recognition*, pp 2082–2087

11. Fisher R, Santos-Victor J, Crowley J (2003) Ec funded caviar project IST 2001 37540. <http://homepages.inf.ed.ac.uk/rbf/CAVIAR>. Accessed 05 May 2020
12. Fu W, Wang J, Zhao C, Lu H, Ma S (2012) Object-centered narratives for video surveillance. In: 2012 19th IEEE International conference on image processing. IEEE, pp 29–32
13. Ghatak S, Rup S, Majhi B, Swamy M (2019) An improved surveillance video synopsis framework: a HSATLBO optimization approach. *Multimed Tools Appl*, pp 1–33
14. Ghatak S, Rup S, Majhi B, Swamy M (2020) HSAJAYA: An improved optimization scheme for consumer surveillance video synopsis generation. *IEEE Trans Consum Electron* 66(2):144–152
15. He Y, Gao C, Sang N, Qu Z, Han J (2017) Graph coloring based surveillance video synopsis. *Neurocomputing* 225:64–79
16. He Y, Qu Z, Gao C, Sang N (2016) Fast online video synopsis based on potential collision graph. *IEEE Signal Processing Letters* 24(1):22–26
17. Höferlin B, Höferlin M, Weiskopf D, Heidemann G (2011) Information-based adaptive fast-forward for visual surveillance. *Multimed Tools Appl* 55(1):127–150
18. Hoshen Y, Peleg S (2015) Live video synopsis for multiple cameras. In: *IEEE International Conference on Image Processing (ICIP)*, pp 212–216
19. Huang CR, Chung PCJ, Yang DK, Chen HC, Huang GJ (2014) Maximum a posteriori probability estimation for online surveillance video synopsis. *IEEE Trans Circuits Sys Video Technol* 24(8):1417–1429
20. K N, Narayanan A (2018) Video synopsis: State-of-the-art and research challenges. In: *IEEE International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*
21. Kang HW, Matsushita Y, Tang X, Chen XQ (2006) Space-time video montage. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol 2, pp 1331–1338
22. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
23. Kolmogorov V, Zabih R (2002) What energy functions can be minimized via graph cuts? In: *European conference on computer vision*. Springer, pp 65–81
24. Kumar TS, Sivanandam S (2012) Object detection and tracking in video using particle filter. In: *Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*. IEEE, pp 1–10
25. Li Z, Ishwar P, Konrad J (2009) Video condensation by ribbon carving. *IEEE Trans Image Process* 18(11):2572–2583
26. Li Z, Tang J, Wang X, Liu J, Lu H (2016) Multimedia news summarization in search. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7(3):1–20
27. Li X, Wang Z, Lu X (2015) Surveillance video synopsis via scaling down objects. *IEEE Trans Image Process* 25(2):740–755
28. Li X, Wang Z, Lu X (2018) Video synopsis in complex situations. *IEEE Trans Image Process* 27(8):3798–3812
29. Lu M, Wang Y, Pan G (2013) Generating fluent tubes in video synopsis. In: *IEEE international conference on acoustics, speech and signal processing*, pp 2292–2296
30. Ma YF, Zhang HJ (2002) A model of motion attention for video skimming. In: *Proceedings. International Conference on Image Processing*, vol 1, pp 1–1
31. Mahapatra A, Sa PK, Majhi B, Padhy S (2016) Mvs: A multi-view video synopsis framework. *Signal Process Image Commun* 42:31–44
32. 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
33. 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
34. Parameswaran L, et al. (2013) A hybrid method for object identification and event detection in video. In: *Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*. IEEE, pp 1–4
35. Pedestrian walking, human activity recognition video, dataset by Uet Peshawar. [https://youtu.be/2bKXv\\_XviFc](https://youtu.be/2bKXv_XviFc) Accessed 05 May 2020
36. Pérez P, Gangnet M, Blake A (2003) Poisson image editing. *ACM Transactions on Graphics (TOG)* 22(3):313–318
37. 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
38. 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
39. Pritch Y, Rav-Acha A, Peleg S (2008) Nonchronological video synopsis and indexing. *IEEE Trans Pattern Anal Mach Intell* 30(11):1971–1984

40. Ra M, Kim WY (2018) Parallelized tube rearrangement algorithm for online video synopsis. *IEEE Signal Processing Letters* 25(8):1186–1190
41. 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
42. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection, pp 779–788
43. Ruan T, Wei S, Li J (2019) Zhao Y, Rearranging online tubes for streaming video synopsis: A dynamic graph coloring approach. *IEEE Transactions on Image Processing*
44. Stauffer C, Grimson WEL (1999) Adaptive background mixture models for real-time tracking. In: *Proceedings. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol 2, pp 246–252
45. Su ST, Chen YY (2008) Moving object segmentation using improved running gaussian average background model. In: *Digital Image Computing: Techniques and Applications. IEEE*, pp 24–31
46. Sun J, Zhang W, Tang X, Shum HY (2006) Background cut. In: *European conference on computer vision. Springer*, pp 628–641
47. Wang WC, Chung PC, Huang CR, Huang WY (2017) Event based surveillance video synopsis using trajectory kinematics descriptors. In: *Fifteenth IAPR international conference on machine vision applications (MVA), IEEE*
48. Xu L, Liu H, Yan X, Liao S, Zhang X (2015) Optimization method for trajectory combination in surveillance video synopsis based on genetic algorithm. *J Ambient Intell Humaniz Comput* 6(5):623–633
49. Yedidia JS, Freeman WT, Weiss Y (2003) Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium* 8:236–239
50. 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
51. Zhu J, Feng S, Yi D, Liao S, Lei Z, Li SZ (2014) High-performance video condensation system. *IEEE Trans Circuits Sys Video Technol* 25(7):1113–1124
52. Zhu J, Liao S, Li SZ (2016) Multicamera joint video synopsis. *IEEE Trans Circuits Sys Video Technol* 26(6):1058–1069
53. Zhu X, Liu J, Wang J, Lu H (2014) Key observation selection-based effective video synopsis for camera network. *Mach Vis Appl* 25(1):145–157

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



**Namitha K** 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 CSE 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.