Rearranging Online Tubes for Streaming Video Synopsis: A Dynamic Graph Coloring Approach

Tao Ruan, Shikui Wei, *Senior Member, IEEE*, Jia Li, *Senior Member, IEEE*, and Yao Zhao, *Senior Member, IEEE*

***Abstract*— To efficiently browse long surveillance videos, the video synopsis technique is often used to rearrange tubes (i.e., tracks of moving objects) along the temporal axis to form a much shorter video. In this process, two key issues need to be addressed, i.e., the minimization of spatial tube collision and the maximization of temporal video condensation. In addition, when a surveillance video comes as a stream, an online algorithm with the capability of dynamically rearranging tubes is also required. Toward this end, this paper proposes a novel graph-based tube rearrangement approach for online video synopsis. The relationships among tubes are modeled with a dynamic graph, whose nodes (i.e., object masks of tubes) and edges (i.e., relationships) can be progressively inserted and updated. Based on this graph, we propose a dynamic graph coloring algorithm to efficiently rearrange all tubes by determining when they should appear. Extensive experimental results show that our approach can condense online surveillance video streams in real time with less tube collision and high compact ratio.**

***Index Terms*— Streaming video synopsis, surveillance, tube rearrangement, dynamic graph coloring.**

I. Introduction

As reported by the IDC's Data Age 2025 Study [1], the amount of data generated worldwide is 16.1 zettabytes in 2016, and the global datasphere will grow to 163 zettabytes in 2025. A large portion of these data is image and video content [2]–[4], especially for non-entertainment purposes such as video surveillance. Actually, surveillance video is growing up to the biggest big data all over the world [5]. To make better use of the long surveillance videos, a straightforward way is to group them into categories and reduce their amount and duration. Toward this end, many techniques have been invented for video classification [6], [7], abstraction [8], [9], montage [10], [11], condensation [12], [13] and synopsis [14], [15]. Among these techniques, video synopsis has attracted much attention in recent years, which aims to condensate a long surveillance video into a much shorter clip by modeling the background [16] as well as extracting [17], [18], rearranging [19], [20] and stitching [21] tubes.

In the past decade, dozens of video synopsis approaches have been proposed, which can be roughly grouped into the offline and online categories. The offline category assumes that the video is static and its condensation is performed when the attributes of all tubes are available. For example, Pritch *et al.* [14] formulated the tube rearrangement as a global Gibbs energy-minimization problem. Nie *et al.* [20] enabled the movement of tubes in both temporal and spatial subspaces. Although these approaches have achieved impressive synopsis effects, they are not suitable for online surveillance systems. In addition, the computational cost of these approaches can be very high since they usually save the complete spatio-temporal information of all tubes and then rearrange them via time-consuming optimization.

To address these problems, online video synopsis approaches were proposed to divide a large tube set into smaller sub-sets [15] or process them one-by-one [22]. For example, He *et al.* [15] found out a new pattern to describe the relationship between tubes. They modeled the sub-sets of tubes as several Potential Collision Graphs (PCGs) and then applied an offline graph coloring algorithm to process the graph. Feng *et al.* [23] and Zhu *et al.* [22] proposed a Tetris based real-time method, which treated tubes as Tetrominos and maintained a buffer to cache a certain number of tubes. However, such greedy framework often leads to local optimal condensation since arrangement of previous tubes have been fixed and the information of newly coming tubes are not involved. Therefore, it is necessary to develop an online approach that can dynamically rearrange previous tubes based on the newly coming tubes.

Toward this end, this paper proposes a novel graph-based tube rearrangement approach for online video synopsis, where the relationships of tubes are modeled with a dynamic graph. Different from previous works, the dynamic graph can be progressively updated along with the video streaming. Based on this graph, a dynamic graph coloring algorithm is proposed

Manuscript received May 4, 2018; revised December 25, 2018; accepted February 25, 2019. Date of publication March 8, 2019; date of current version June 20, 2019. This work was supported in part by the National Key Research and Development of China under Grant 2017YFC1703503, in part by the National Natural Science Foundation of China under Grant 61572065 and Grant 61532005, in part by the Program of China Scholarship Council under Grant 201807095006, in part by the Beijing Nova Program under Grant Z181100006218063, and in part by the Fundamental Research Funds for the Central Universities under Grant 2018JBZ001. The work of S. Wei was supported by the National Engineering Laboratory for Urban Rail Transit Communication and Operation Control. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kalpana Seshadrinathan. *(Corresponding authors: Shikui Wei; Jia Li.)*

T. Ruan, S. Wei, and Y. Zhao are with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China, and also with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China (e-mail: 16112064@bjtu.edu.cn; shkwei@bjtu.edu.cn; yzhao@bjtu.edu.cn).

J. Li is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, also with the Peng Cheng Laboratory, Shenzhen 518000, China, and also with the Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China (e-mail: jiali@buaa.edu.cn).

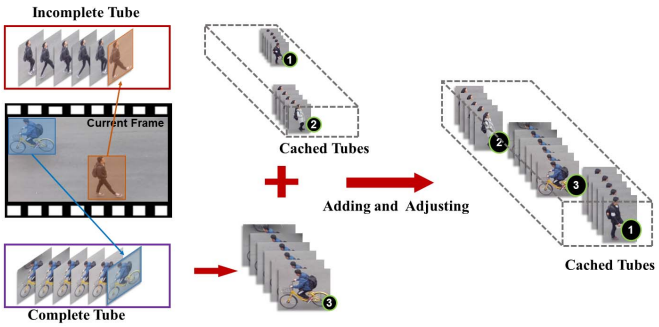

Fig. 1. The demonstration of our tube rearranging method. It is worth noticing that the time locations of previously rearranged tubes 1 and 2 are updated after inserting new tube 3.

to efficiently re-assign the time location of tubes so as to minimize the spatial collision and maximize the condensation rate. The key idea is illustrated in Fig. 1. When a complete tube (*i.e.*, a complete trajectory of an object) is extracted and added into a buffer of tubes, the adding and adjusting procedures assign the new tube to a time location and the time locations of previously rearranged tubes are also updated so as to make rearrangement better. In our work, the cached tubes are modeled by a dynamic graph, and the updating procedure is treated as vertex inserting and dynamic graph coloring operations. Experimental results show that the proposed approach can generate impressive condensation results for online surveillance video streams in real time.

The main contributions of this paper are summarized as follows: 1) we propose a graph model that can dynamically describe the tubes and their relationships in streaming videos; 2) we propose a dynamic graph coloring algorithm that can efficiently rearrange tubes with low spatial collision and high condensation rate; and 3) we develop a system that can be deployed for online video synopsis, whose effectiveness has been validated in extensive experiments.

II. Related Work

In this section, we introduce three categories of techniques that are closely related to video synopsis, including 1) video abstraction, 2) video montage and ribbon carving-based condensation, and 3) video synopsis.

A. Video Abstraction

Video abstraction [24] is a relatively early solution that generates a summary of long video by keeping crucial video frames. Typically, there are two kinds of video abstraction approaches: video summary and video skimming. Video summary approaches [25]–[27] generated static storyboards by selecting several discrete frames from the original video. On the contrary, video skimming approaches [28]–[30] preserved more dynamic information by selecting representative clips with continuous video frames. Generally speaking, video abstraction can significantly condensate long video, but some interesting objects may be removed together with discarded frames or clips.

B. Video Montage and Ribbon Carving-Based Condensation

To preserve the most informative foreground objects in condensed video, the video montage technique was proposed [10], which separated the input video into several space-time portions and fused these segments into a shorter video according to the informative content (*e.g.*, human actions). In this manner, rich video information can be compressed into a narrower video space. Nevertheless, this algorithm is very complex and may bring obvious seams caused by portion fusion. To address these issues, Li *et al.* [12] proposed to iteratively remove structures called *ribbon* with the help of dynamic programming so as to avoid removing a whole frame once. Nguyen *et al.* [13] improved this method by using a three-stage condensation scheme to preserve the chronological order, which achieved a more proper way to deal with a relatively long video. While Ribbon carving-based video condensation is simple and fast, they usually have low condensation rate and noticeable seams in synopsis videos.

C. Video Synopsis

A trade-off technique between the video montage and ribbon carving-based video condensation is video synopsis [14], [31]–[34]. Video synopsis aims at not only achieving high condensation rate but also providing much more natural experiences for human browsing. In the video synopsis approaches, tube rearranging is a key step to reduce the length of the final compressed video and the collisions among foreground objects. In traditional methods, the tube rearranging problem is usually solved by using some off-line optimization algorithms. For example, Pritch *et al.* [14], [35] built a Markov Random Field model over the tubes, and formulated the tube rearranging as a Gibbs energy minimization problem. Li *et al.* [36] scaled down the objects when collisions occurred in the tube rearranging phase, and proposed an adaptive way to find the proper zoom-out coefficients to further reduce collision cost. Nie *et al.* [20] proposed to move tubes in both temporal and spatial subspaces so as to achieve both high compression and low spatial collision. In most cases, the offline methods can produce impressive synopsis results. However, they are often time-consuming and not suitable for some online scenarios.

In order to solve this problem and achieve synopsis results in real-time, some online methods [22], [23], [37], [38] have been proposed to speed up the tube rearranging procedure. For example, Feng *et al.* [23] proposed a Tetris-based real time method for tube rearranging, which treated tubes as Tetrominos and maintained a tube buffer that would be cleared when the tube number touches the upper bound. Zhu *et al.* [22] further improved this method with multi-threading and GPU support. He *et al.* [38] discovered a new pattern to describe the relationship between tubes as a Potential Collision Graph. By applying a simple greedy method to process the graph, tubes can be rearranged rapidly. Furthermore, He *et al.* [15] proposed an off-line graph coloring-based method to better use the Potential Collision Graph to rearrange tubes. Similar constraints are also employed in online video synopsis methods in [22] and [23].

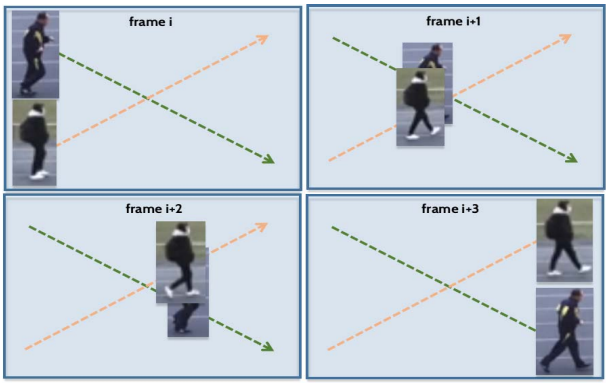

Fig. 2. An example of activity collision.

To sum up, most efforts in existing video synopsis works are spent on the offline tube rearrangement, while some researchers start to notice the problem of online video synopsis that will be helpful to process the overwhelming surveillance videos. In such online synopsis researches, a key issue is to rearrange dynamically formed tubes in real time, which is also the major concern of this study.

III. Problem Formulation

Video synopsis aims at condensing a long video into a much shorter one by rearranging the essential tubes along the temporal axis. In this process, three key factors should be considered, including:

A. Tube Collision

Tubes rearranged to the same time interval may overlap with each other, which may result in poor visualization experience. As shown in Fig. 2, two persons running along the green and red lines collide in the second and third frames, leading to severe occlusion. Such kind of tube collision should be minimized in video synopsis.

B. Activity Completeness

Missing key activities in video synopsis will lead to serious consequences in video surveillance. Therefore, the completeness of every single tube needs to be maximized in video synopsis, *i.e.*, the object masks in all frames and the pixels they cover should be preserved as much as possible.

C. Interaction Preservation

In surveillance videos, the interaction between objects (*e.g.*, conversation) needs to be detected and preserved as much as possible in tube rearrangement phase.

In practice, the activity completeness can be maximized by forming tubes without discarding any object even when the activity lasts an extremely long period. The interaction between objects can be preserved by fusing multiple objects, once interacted at a specific time, into a single tube (*e.g.*, two persons ever embraced are fused into a single tube). Therefore, the major factor we need to take into account is the tube collision. With this key factor in mind, we state the problem of video synopsis as follows. Let $\mathbb{V} = \{\mathcal{F}_k\}_{k=1}^{K}$ be a streaming video with K frames already played and $\mathbb{T} = \{\mathcal{T}_m\}_{m=1}^{M}$ be the set of M tubes contained in $\mathbb{V}$. Here, $\mathcal{F}_k$ is the kth frame of $\mathbb{V}$ and $\mathcal{T}_m$ is the mth tube formed by a set of chronological masks:

$$\mathcal{T}_m = \{\{O_{mn}\}_{n=1}^{N_m}, t_m\}, \tag{1}$$

where O_{mn} is the nth spatial mask of the object with identified label m. Here, a spatial mask O_{mn} is represented by a set of pixels covering the object in a specific frame. The positive integer t_m indicates the temporal position of the first spatial mask O_{m1} in $\mathbb{V}$, and N_m is the length of the tube $\mathcal{T}_m$ (*i.e.*, the number of frames containing the object). By assuming that the object appears in consecutive frames, the temporal location of the ith object mask O_{mi} can be computed as $t_m + i - 1$.

In video synopsis, the primary objective is to derive a new start temporal location $\hat{t}_m$ for each tube $\mathcal{T}_m$ so that it forms a new tube $\hat{\mathcal{T}}_m$ in the condensed video with

$$\hat{\mathcal{T}}_m = \{\{O_{mn}\}_{n=1}^{N_m}, \hat{t}_m)\}. \tag{2}$$

As a result, the synopsis process of $\mathbb{V}$ can be formulated as the minimization problem

$$\begin{aligned} \{\hat{t}_m^*\} = \arg\min_{\{\hat{t}_m\}} \hat{K} + \lambda \cdot \sum_{u=1}^{M} \sum_{v=u+1}^{M} \Omega(\hat{\mathcal{T}}_u, \hat{\mathcal{T}}_v), \\ \text{where } \hat{t}_m \in \{1, \ldots, \hat{K} - N_m + 1\}, \forall m, \\ \max(\{\hat{t}_m + N_m - 1 | \forall m\}) = \hat{K}, \\ \min(\{\hat{t}_m | \forall m\}) = 1 \text{ and } \hat{K} \leq K, \end{aligned} \tag{3}$$

where $\hat{K}$ is the number of frames in the condensed video. The term $\Omega(\cdot)$ penalizes the probable collision between two tubes, which can be defined as

$$\Omega(\hat{\mathcal{T}}_u, \hat{\mathcal{T}}_v) = \sum_{i=1}^{N_u} \sum_{j=1}^{N_v} \delta(\hat{t}_u + i = \hat{t}_v + j) \cdot \phi(O_{ui}, O_{vj}) \tag{4}$$

where $\delta(\mathbf{e})$ is an indicator function that equals 1 if the event $\mathbf{e}$ holds and 0 otherwise. The term $\phi(O_{ui}, O_{vj})$ indicates the spatial collision loss when the object O_{ui} from the uth tube and the object O_{vj} from the vth tube appear in the same frame of the condensed video. Such a loss can be defined according to the ratio of spatial overlapping between the two objects. For efficient video condensation, we set $\phi(O_{ui}, O_{vj})$ to 1 if O_{ui} and O_{vj} have any degree of overlapping in spatial location and 0 otherwise.

By incorporating Eq. (4) into Eq. (3), we can see that video synopsis aims to minimize the length of condensed video while preserving the visibility of tubes (*i.e.*, minimal collision). For off-line approaches, the tubes are formed before the optimization process so that the optimization problem can be effectively resolved. However, in the online scenario, the tubes are dynamically formed, while the spatial masks and length of tubes will change from time to time. In this case, an early tube may be placed at inappropriate time positions without knowing what and how many tubes may come in subsequent video streams. Therefore, we propose to develop

an online approach that can dynamically rearrange all tubes in the buffered video streams.

IV. A Dynamic Graph Coloring Approach for Online Video Synopsis

The graph coloring problem is one of the most classic problems in graph theory. Given a graph $G = \{V, E\}$ with a set of vertices V and a set of edges E, the objective of graph coloring in our scenario is to assign a limited number of time locations (colors) to the spatial masks (vertices) in tubes under certain constraints (edges) [39], [40]. In this section, we will introduce a dynamic graph coloring approach to rearrange the tubes formed dynamically from steaming video.

A. Overview of Online Video Synopsis

For surveillance video streams, a key challenge in resolving the optimization problem (3) is that the tube number and tube relationships are updated from time to time. As a result, the penalty term $\Omega(\hat{\mathcal{T}}_u, \hat{\mathcal{T}}_v)$ should be defined in an easy-to-resolve form to handle the dynamic tubes. Toward this end, we first represent the tubes and their relationships as a graph that is easy to perceive and understand. Therefore, we directly abstract a spatial object mask O_{mn} in a tube $\mathcal{T}_m$ as a vertex and model the relationship between two spatial object masks meeting certain constraints as an edge. For the sake of simplicity, we denote a vertex with the same symbol O_{mn}.

In video synopsis, two relationship constraints between object masks, *i.e.*, collision constraint and chronological consistency constraint between two spatial object masks, are taken into account. These two constraints are modeled by two types of edges. The first type of edges E_u is undirected, which conveys the spatial collision between two spatial masks in different tubes. That is, if $\phi(O_{ui}, O_{vj}) = 1$ and $u \neq v$, they are connected by an undirected edge. The second type of edges E_d is directed, which is employed to encode the chronological order of spacial masks in the same tube. Given any two contiguous spatial masks O_{ui} and O_{uj}, $j = i + 1$ in the same tube $\mathcal{T}_u$, there exist a directed edge from O_{ui} to O_{uj}. Note that two spatial object masks that fail to meet such relationship constraints will have no connection. An example of the graph constructed from two tubes can be found in Fig. 3.

Once a graph is constructed, we use a graph coloring algorithm to assign the minimal number of colors to graph vertices, while such colors, represented by integers, can be viewed as the time locations of spatial object masks. Two vertices connected by an undirected edge should be assigned to different colors (*i.e.*, time locations), while vertices connected by directed edges should be assigned consecutive colors. Instead of constructing the graph once, the dynamic graph coloring constructs the graph incrementally to ensure efficient optimization, and the assigned colors to previous vertices can be further re-colored according to the optimization result after taking into account new coming vertices. Therefore, the dynamic graph coloring process can be viewed as a stepwise optimization problem on a sequence of graphs $\mathcal{G} = \{G(t)\}_{t=1}^{M}$.

In order to obtain real-time synopsis, we perform the graph coloring with two strategies. First, we reduce the maximum number of vertices in $G(t)$ so as to speed up the optimization. Second, we assume that $G(t)$ and $G(t+1)$ are quite alike so that the optimal solution for $G(t)$ can be treated as the initial solution for $G(t+1)$ that is nearly optimal. With these two strategies in mind, we set an upper bound P for the number of tubes and update only one tube once. That is, after the number of tubes in $G(t)$ gets up to P, a tube will be selected and fused into synopsis video when a new tube $\mathcal{T}_{t+1}$ is coming. Note that a long tube will surely influence the computational cost of a synopsis video. However, we do not make any limitation on the length of a tube. Given a long tube, we can uniformly divide it into several shorter ones and then use methods like sticky tracking [22] to group them into a single tube.

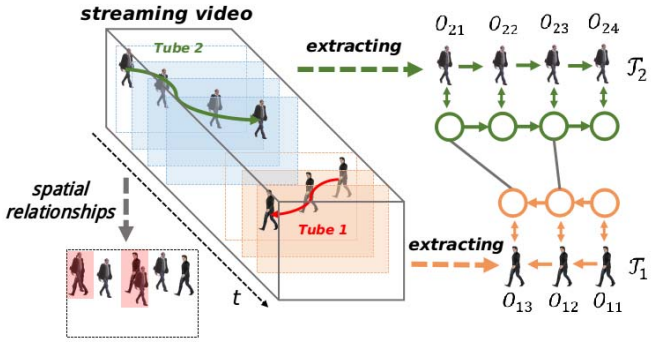

Fig. 3. An example of the graph constructed to describe the relationships of object masks in two tubes. $\mathcal{T}_1 = \{\{O_{1n}\}_{n=1}^{3}, t_1\}$ and $\mathcal{T}_2 = \{\{O_{2n}\}_{n=1}^{4}, t_2\}$ are two tubes. The vertices O_{11},O_{12}, and O_{13} in tube $\mathcal{T}_1$ are connected by two directed edges. In addition, O_{21} collides with O_{13} if they are rearranged into the same frame in the synopsis video. Therefore, we connect them with an undirected edge. Similarly, O_{23} and O_{12} are also connected by an undirected edge.

Fig. 4 illustrates the overview of the proposed approach. Giving a streaming video, tubes are extracted and fed into the dynamic graph one by one in chronological order. When the number of tubes in $G(t)$ reaches up to P, a colored tube is selected and moved to synopsis video. Then, the new coming tube $\mathcal{T}_{t+1}$ is added to the $G(t)$, and the graph is updated to form a new graph $G(t+1)$. The iterative process is continued until all the tubes are extracted and moved to synopsis video.

B. Updating of Dynamic Graph

Intuitively, we can update the synopsis result after receiving every new tube from the streaming video. However, dynamic graph coloring is an NP-hard problem, even if we limit the number of vertices in $G(t)$ and make $G(t)$ and $G(t+1)$ be quite alike. Therefore, it is necessary to find an approximate solution to achieve a good balance between effectiveness and efficiency. In this section, we propose an efficient algorithm to update the graph $G(t)$. A detailed description of the updating procedure can be found in Algorithm 1. Noting that coloring a tube $\mathcal{T}_m$ is equivalent to assign a time location to its first spatial object mask O_{m1}. To make description more clear, we denote $gc(O_{mn})$ as the color (time location) that has been assigned to O_{mn}, and $gc(\mathcal{T}_m)$ is interchangeable to $gc(O_{m1})$.

When a new tube $\mathcal{T}_{t+1}$ is coming, it is added into the graph $G(t)$ in different strategies, *i.e.*, adding procedure

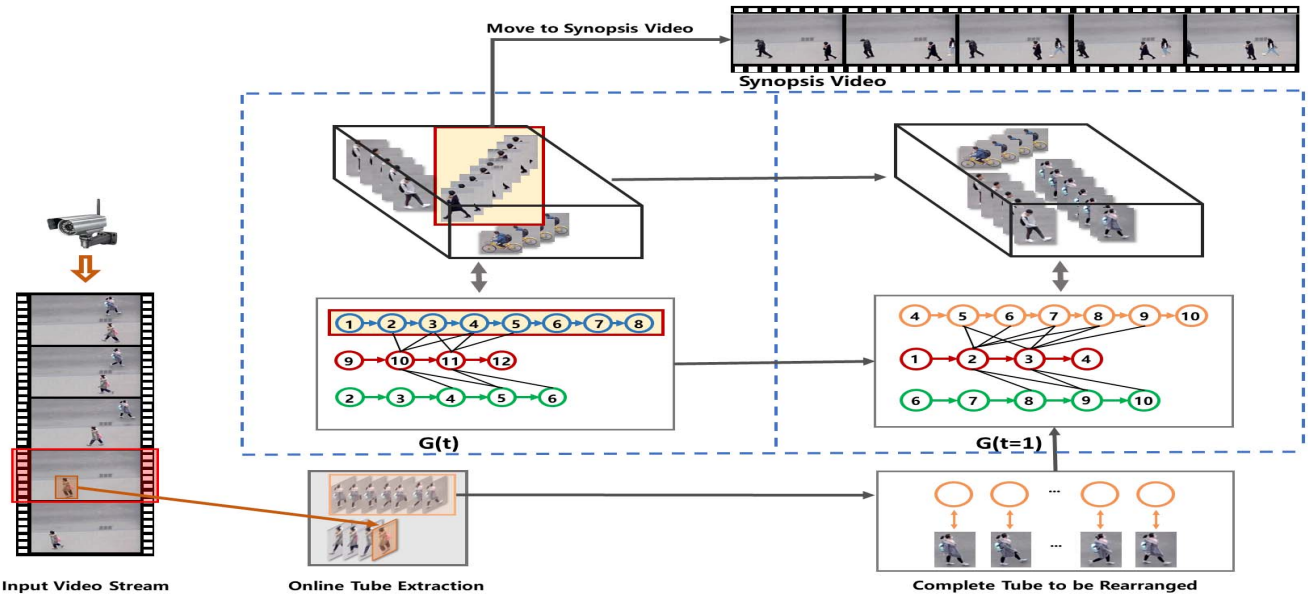

Fig. 4. Overview of the proposed online video synopsis based on dynamic graph coloring with $P = 3$.

and adjusting procedure. For the first one, we employ the greedy algorithm, which is shown in procedure $adding(\cdot)$ in algorithm 1. The key idea is to make a statistic for $\mathcal{T}_{t+1}$ on all possible collisions occurred with the vertices $\{O_{mn}\}$ in $G(t)$ at any colors (time locations) that are feasible to color the $O_{(t+1)1}$. After that, we can obtain the minimum feasible color, at which the number of collisions is zero or below to a pre-defined tolerance. In particular, given a colored graph $G(t)$, we record the number of collisions at each time location in a big array $NC[\cdot]$. Here, the index of $NC[\cdot]$ denotes the time location, and $NC[c_{tmp}]$ means the total number of collisions with vertices in $G(t)$ if $\mathcal{T}_{t+1}$ is rearranged to the time location c_{tmp}. When $O_{(t+1)i} \in \mathcal{T}_{t+1}$ has spatial overlap with $O_{mn} \in V(t)$ and $O_{(t+1)i}$ is assigned to the same color $gc(O_{mn})$, there exists a collision. Since the color of $gc(\mathcal{T}_{t+1})$ can be inferred from $gc(O_{mn})-i+1$, a collision is accumulated for T_{t+1} at the time location c_{tmp}. In this way, the collisions occurred among spatial object masks are converted to the collisions at all possible starting time locations assigned to the tube. Once we have collected the array $NC[\cdot]$, we need to find the minimum available color to color $\mathcal{T}_{t+1}$, *i.e.*, $coloring(\cdot)$ procedure.

If we want to avoid all the collisions, $gc(\mathcal{T}_{t+1})$ will be the minimum index x of $NC[\cdot]$, which satisfies both $x \geqslant c_{min}$ and $NC[x] = 0$. Here, c_{min} is the minimum available color currently, which is updated dynamically by the starting time location of removed tube. However, the proper solution is too strict for the video synopsis scenario. Generally speaking, it is allowable to tolerate a certain degree of collisions so as to get more compactly compressed video. Therefore, we pre-define a nonnegative constant h to balance the collision and compressed performance. Here, h indicates the number of continuous frames having spatial collision for two tubes in synopsis video. Once we pre-define the tolerance of collision to be h, we can color the new tube $\mathcal{T}_{t+1}$ with the minimum feasible color x, which satisfies both $x \geqslant c_{min}$ and $NC[x] \leqslant h$.

As indicated in the $adding(\cdot)$ procedure, only the new coming tube is colored, and the others keep no change. Although this strategy makes full use of colored results of $G(t)$ and leads to efficient dynamic graph coloring, it also improves the probability of bad result. The best way is to employ a greedy algorithm to renew all the tubes in $G(t+1)$. However, it is time-consuming, and real-time performance will be sacrificed. Therefore, we introduce an alternative strategy to give a chance for obtaining better results as well as keeping real-time performance. We call it $adjusting(\cdot)$ procedure. The key idea is to directly set the color of new tube $\mathcal{T}_j$ to the minimum available color c_{min}. Then, all the colored $\{\mathcal{T}_m\}$ that have collisions with $\mathcal{T}_j$ are removed from $G(t)$. Finally, the removed tubes are re-added to $G(t)$ to form $G(t+1)$ by using the $adding(.)$ procedure. Both the $adding(\cdot)$ and $adjusting(\cdot)$ procedures can treated as individual approximate solutions. In the updating procedure, we select the better one from them. Since the objective is to minimize the length of the condensed video, we select the $G(t+1)$ that have minimum ending time location as the better one. In algorithm 1, $EC(\cdot)$ denotes the set of ending time locations of all tubes in condensed video, and $max\{\cdot\}$ is the maximum value in the set. In fact, it is reasonable and straightforward, since the earlier ending time means shorter length of synopsis video.

C. Implementation Details of the Proposed Approach

The proposed online video synopsis approach consists of three main components that focus on the extraction, rearrangement and stitching of tubes, respectively. In order to extract tubes from videos, we first employ a fast and robust method, ViBe [18], to extract the foreground pixels from the raw frame, followed by a background modeling procedure. Here we use the mean value of received frames to build the background. After that, the sticky tracking algorithm proposed in [22] is employed to perform interaction-preserved object tracking. The object masks generated by the tracking

Algorithm 1 Updating the Graph

Procedure *Updating*($G(t)$, $\mathcal{T}_{t+1}$, c_{min}, h)
  $G_{tmp_1} = adding(G(t), \mathcal{T}_{t+1}, c_{min}, h)$;
  $G_{tmp_2} = adjusting(G(t), \mathcal{T}_{t+1}, c_{min}, h))$;
  $maxE_{tmp_1} = max\{EC(V_{tmp_1})\}$;
  $maxE_{temp_2} = max\{EC(V_{tmp_2})\}$;
  if $maxE_{tmp_1} \leq maxE_{tmp_2}$ **then**
    $G(t+1) = G_{tmp_1}$
  else
    $G(t+1) = G_{tmp_2}$
  end
  return $G(t+1)$;
end
Procedure *adding*($G(t)$, $\mathcal{T}_{t+1}$, c_{min}, h)
  forall $O_{(t+1)i} \in \mathcal{T}_{t+1}$ **do**
    forall $O_{mn} \in V(t)$ **do**
      if $\phi(O_{(t+1)i}, O_{mn}) = 1$ **then**
        $c_{tmp} = gc(O_{mn}) - i + 1$;
      end
      if $c_{tmp} \geqslant 0$ **then**
        $NC[c_{tmp}] = NC[c_{tmp}] + 1$;
      end
    end
  end
  $gc(\mathcal{T}_{t+1})$ = coloring($NC(\cdot), c_{min}, h$);
  Add $\mathcal{T}_{t+1}$ into $G(t)$ to form $G(t+1)$;
  return $G(t+1)$;
end
Procedure *adjusting*($G(t)$, $\mathcal{T}_{t+1}$, c_{min}, h)
  $gc(\mathcal{T}_{t+1}) = c_{min}$;
  forall $\mathcal{T}_m$ *in* $G(t)$ **do**
    if $\exists\ \phi(O_{(t+1)i}, O_{mn}) = 1$ & $gc(O_{(t+1)i}) = gc(O_{mn})$ **then**
      Remove the tube $\mathcal{T}_m$ from $G(t)$;
      Add $\mathcal{T}_m$ into $Queue$;
    end
  end
  Add $\mathcal{T}_{t+1}$ into $G(t)$ to form $G(t+1)$;
  forall $\mathcal{T}_m$ *in* $Queue$ **do**
    $G(t+1)$ = Adding($G(t+1)$, $\mathcal{T}_m$, c_{min}, h);
  end
  return $G(t+1)$;
end

procedure are stored in a cache pool, and a complete tube (object mask sequence) for the object is obtained if no new object mask is detected in a short period (*e.g.*, 2 seconds). The complete tubes are fed into the algorithm 2 to perform dynamic rearrangement. Finally, the tubes removed from dynamic graph are stitched with the background to generate a short video clip. In our work, Poisson Editing [21] is employed for stitching.

The pipe line of tube rearrangement is shown in algorithm 2. Formally, given a set of tubes $\mathbb{T} = \{\mathcal{T}_m\}_{m=1}^{M}$ in chronological order, these tubes are added dynamically into graph $G(t)$ one by one by using the $updating(.)$ procedure. Once the number

TABLE I
PROPERTIES OF 12 SURVEILLANCE VIDEOS

Video	Size	Duration (s)	#Frame	#Obj.
Highway	320×80	799	23998	224
Overpass	320×120	798	23948	93
Yard	640×480	309	6181	212
Sidewalk	320×240	1200	24000	239
Playground	320×180	3751	112547	411
Road-1	640×360	474	14248	534
Road-2	640×360	896	26900	263
Road-3	640×360	243	7309	176
Road-4	640×360	380	11420	115
Square-1	640×360	915	27465	106
Square-2	640×360	661	19856	139
Crossroad	518×660	2679	80394	1074

Algorithm 2 Complete Pipeline

forall $\mathcal{T}_t$ ***in*** $\mathbb{T}$ **do**
  if $t > P$ **then**
    $\mathcal{T}_{del}$ = removing(G(t))
    $c_{min} = max(c_{min}, gc(\mathcal{T}_{del}))$;
  end
  $G(t) = Updating(G(t-1), \mathcal{T}_t, c_{min}, h)$;
end
Stitching rest tubes in the final graph $G(M)$;

of tubes in $G(t)$ gets up to the limitation P. A tube will be selected and moved into synopsis video before adding the new coming tube $\mathcal{T}_{t+1}$. In the proposed framework, the tube that has the minimum ending time location is selected, and the minimum available color c_{min} is updated to the maximum one in old c_{min} and the time location of the tube to be removed. In this way, the time locations below to c_{min} will be not assigned, which will minimize both the collisions and the length of synopsis video.

V. EXPERIMENTS

To validate the effectiveness of the proposed approach, we conduct extensive experiments on surveillance videos collected with diverse scenes and interactions.

A. Experiment Settings

In experiments, we collect 12 long videos for performance testing, whose properties are listed in Tab. I. These videos cover different surveillance scenarios and diverse object movement patterns, as illustrated in Fig. 5.

Based on these videos, we compare our approach with five state-of-the-art video synopsis methods, including three online methods (**MAP-VS** [41], **HPVC** [22] and **FastPCG** [38]) and two offline methods (**OVS**$_b$ [14] and **PCGC**$_b$ [15]). Among these methods, **MAP-VS** is based on fixed tube rearranging strategies, **HPVC** adopts energy minimization and content-aware tube filling, and **FastPCG** is based on graph operation. In particular, to ensure that the offline methods can

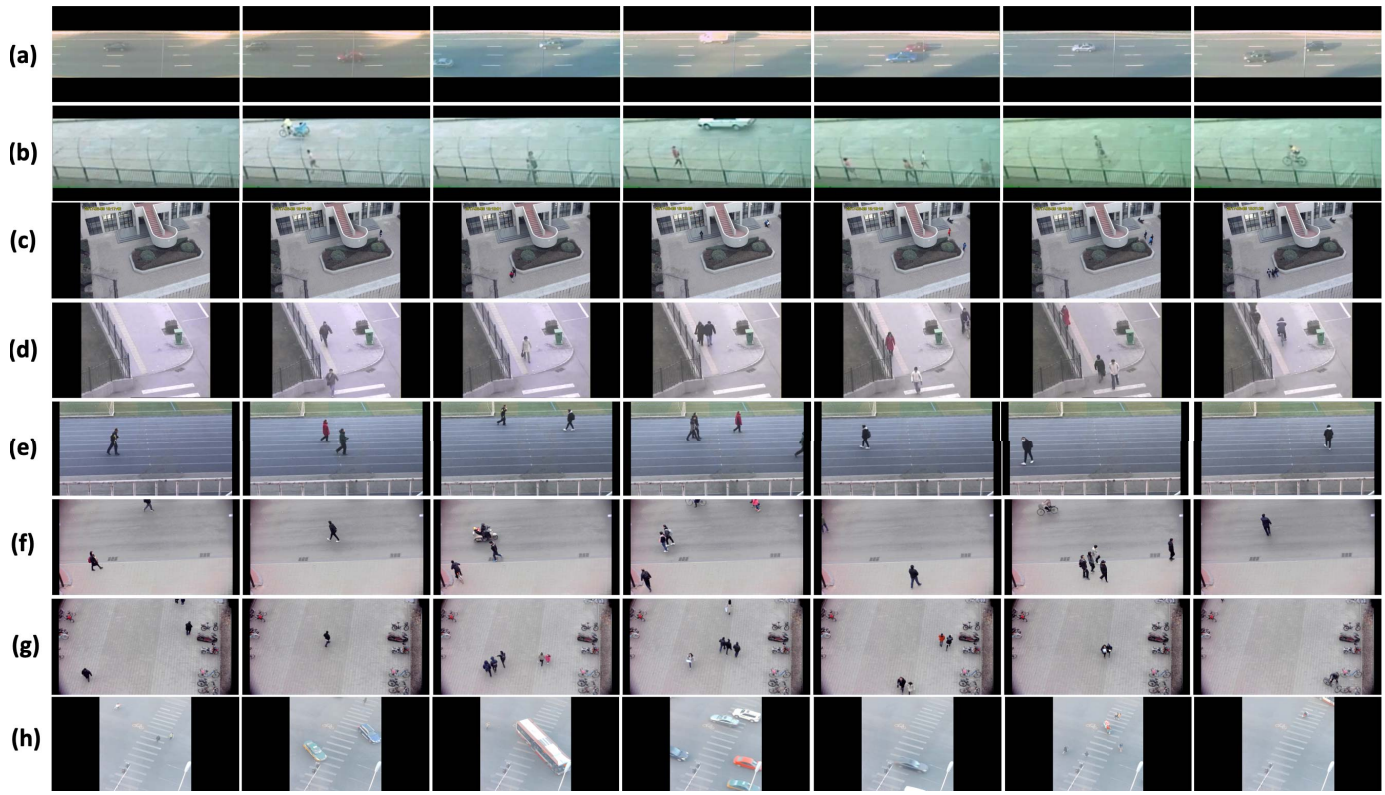

Fig. 5. Testing scenarios. (a) Highway, (b) Overpass, (c) Yard, (d) Sidewalk, (e) Playground, (f) Road, (g) Square, (h) Crossroad.

obtain online synopsis results from surveillance video streams, we equally divide the original video into $b \in \{1, 2, 4, 8\}$ batches for offline methods $\mathbf{OVS}_b$ [14] and $\mathbf{PCGC}_b$. The original offline results can be obtained at $b = 1$.

In the comparisons, we employ the commonly used evaluation metrics in previous works [38], [41], including

1) Frame Condensation Ratio (FR): is a temporal metric computed as the ratio between the numbers of frames in the synopsis video and the original video. A small FR means a higher condensation rate.

2) Frame Compact Ratio (CR): is a spatial metric that measures how the spatial space in a synopsis video is occupied by various objects. It can be computed as the ratio between the numbers of object pixels and total pixels in a synopsis video.

3) Non-Overlapping Ratio (NOR): measures the degree of collision among tubes in a synopsis video, which can be computed as the ratio between the number of pixels occupied by all objects and the sum of pixels of each individual object mask. A higher NOR score means less collision, and a perfect synopsis video without any tube collision will lead to a NOR score of 1.

4) Chronological Disorder (CD): measures the quantities of tube pairs ($\mathcal{T}_a$, $\mathcal{T}_b$) that $\mathcal{T}_a$ appears earlier than $\mathcal{T}_b$ in the original video but opposite in the synopsis video. It can be computed as the ratio between the number of reversed tube pairs and the total number of tubes. A higher CD means a more serious break of chronological order.

To make fair comparisons with these four metrics, we fix the frame condensation ratio (FR) and compare the scores of the other three metrics when using the same tube extraction algorithm for all methods. In addition, we re-implement and compare all these methods with a single-thread and run them on a C++ platform with a 3.5GHz CPU and 16GB memory.

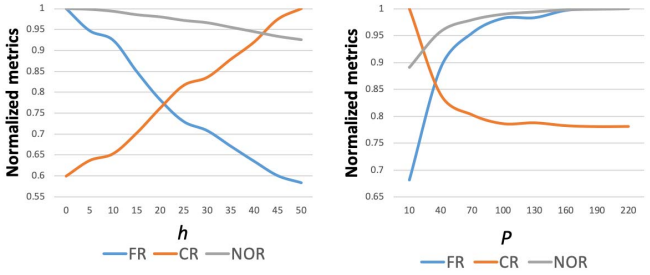

Fig. 6. Effects of two parameters in our approach on the video "Highway". The performance scores are normalized to [0, 1] for clear visualization.

B. Parameter Analysis

In the proposed method, there are two key parameters, *i.e.*, h and P. The parameter h indicates the number of frames that two tubes have spatial collision in a synopsis video, and P is the upper bound of the maximum tube number in a dynamic graph. In order to evaluate the effect of these two parameters, we conduct experiments on the video "Highway." As shown in the left side of Fig. 6, the FR score will remarkably reduce when h increases, indicating higher condensation rates. This is caused by the fact that a higher value of h can tolerate more spatial collisions. In contrast, the CR score will significantly increase since a higher condensation rate greatly reduces the total number of pixels in a synopsis video. Meanwhile, NOR decreases with an increasing h, indicating

TABLE II

COMPARISON OF FRAME COMPACT RATIO WHEN THE FRAME CONDENSATION RATIO IS FIXED

Model (fixed FR)	Highway (0.0679)	Overpass (0.1858)	Yard (0.4351)	Sidewalk (0.2550)	Playground (0.0929)	Road-1 (0.4837)	Road-2 (0.1738)	Road-3 (0.2545)	Road-4 (0.1847)	Square-1 (0.0977)	Square-2 (0.1259)	Crossroad (0.2358)	Average -
OVS$_{b=1}$	0.2436	0.1076	0.0394	0.1304	0.1408	0.2164	0.1249	0.1101	0.1747	0.0812	0.0683	0.1752	0.1344
OVS$_{b=2}$	0.2339	0.1078	0.0388	0.1302	**0.1417**	0.2243	0.1250	0.1097	0.1747	0.0811	0.0682	0.1956	0.1359
OVS$_{b=4}$	0.2334	0.1060	0.0378	0.1299	0.1413	0.2242	0.1247	0.1083	0.1743	0.0807	0.0679	0.1953	0.1354
OVS$_{b=8}$	0.2380	0.1027	0.0379	0.1290	0.1402	0.2249	0.1243	0.1083	0.1709	0.0782	0.0644	0.1949	0.1345
PCGC$_{b=1}$	0.2917	0.0921	0.0488	0.1121	0.1313	0.2014	0.1123	0.0823	0.1436	0.0830	0.0731	0.1660	0.1281
PCGC$_{b=2}$	0.2445	0.0931	0.0474	0.1138	0.1371	0.2071	0.1213	0.0917	0.1428	0.0712	0.0749	0.1770	0.1269
PCGC$_{b=4}$	0.2167	0.0913	0.0451	0.1125	0.1299	0.2024	0.1124	0.0811	0.1426	0.0705	0.0899	0.1724	0.1223
PCGC$_{b=8}$	0.2312	0.0973	0.0461	0.1120	0.1249	0.2010	0.1126	0.0848	0.1484	0.0725	0.0824	0.1693	0.1236
MAP-VS	0.3592	0.0992	0.0402	0.1263	0.1151	0.2136	0.1182	0.1067	0.1675	0.0770	0.0645	0.1854	0.1394
HPVC	0.2652	0.0970	0.0446	0.1183	0.1194	0.1889	0.1112	0.0984	0.1513	0.0900	0.0815	0.1690	0.1279
FastPCG	0.3498	0.0967	0.0462	0.1139	0.1196	0.1994	0.1122	0.0880	0.1345	0.0767	0.0725	0.1749	0.1320
Our	**0.3732**	**0.1210**	**0.0559**	**0.1338**	0.1407	**0.2352**	**0.1303**	**0.1143**	**0.1836**	**0.0933**	**0.0913**	**0.1981**	**0.1559**

TABLE III

COMPARISON OF NON-OVERLAPPING RATIO WHEN THE FRAME CONDENSATION RATIO IS FIXED

Model (fixed FR)	Highway (0.0679)	Overpass (0.1858)	Yard (0.4351)	Sidewalk (0.2550)	Playground (0.0929)	Road-1 (0.4837)	Road-2 (0.1738)	Road-3 (0.2545)	Road-4 (0.1847)	Square-1 (0.0977)	Square-2 (0.1259)	Crossroad (0.2358)	Average
OVS$_{b=1}$	0.6416	0.9716	0.9634	0.9692	0.9789	0.9326	0.9873	0.9879	0.9885	0.9944	0.9935	0.8754	0.9404
OVS$_{b=2}$	0.6161	0.9735	0.9508	0.9690	**0.9797**	0.9664	0.9879	0.9850	0.9889	0.9935	0.9927	0.9781	0.9485
OVS$_{b=4}$	0.6148	0.9570	0.9257	0.9662	0.9792	0.9661	0.9852	0.9723	0.9866	0.9892	0.9885	0.9740	0.9421
OVS$_{b=8}$	0.6270	0.9277	0.9285	0.9625	0.9740	0.9690	0.9824	0.9723	0.9674	0.9577	0.9371	0.9666	0.9285
PCGC$_{b=1}$	0.6847	0.9249	0.9271	0.8238	0.7940	0.8730	0.9015	0.9259	0.8925	0.9324	0.8829	0.8717	0.8695
PCGC$_{b=2}$	0.6710	0.8247	0.9043	0.8145	0.6970	0.8575	0.9125	0.9414	0.8792	0.9125	0.8839	0.9175	0.8513
PCGC$_{b=4}$	0.6571	0.8912	0.8948	0.8240	0.6829	0.8545	0.8906	0.9224	0.8719	0.9125	0.8932	0.9063	0.8501
PCGC$_{b=8}$	0.6672	0.8930	0.7997	0.7411	0.5813	0.8010	0.9073	0.9424	0.8489	0.9134	0.8873	0.8976	0.8234
MAP-VS	0.9396	0.8891	0.9540	0.9046	0.8078	0.9110	0.9289	0.9160	0.9392	0.9380	0.9115	0.9092	0.9124
HPVC	0.7255	0.8765	0.9033	0.8752	0.8515	0.8089	0.9126	0.9047	0.8854	0.9245	0.8979	0.8774	0.8703
FastPCG	0.9031	0.8714	0.9037	0.8247	0.8951	0.8312	0.8908	0.9310	0.8574	0.9316	0.8908	0.9058	0.8864
Our	**0.9872**	**0.9997**	**0.9994**	**0.9718**	0.9745	**0.9787**	**0.9940**	**0.9947**	**0.9934**	**0.9984**	**0.9997**	**0.9841**	**0.9896**

a higher collision. Interestingly, NOR decreases much slower than other metrics, implying that the proposed method can avoid severe collision even if the condensation rate is very high. Similarly, the effects of changing P is opposite to those of h, which is reasonable because increasing P will introduce more tubes into the rearrangement optimization process and thus incorporate more global information. In this manner, the collision becomes less so that NOR and FR increases and CR decreases.

C. Model Comparison With Fixed FR

In our experiments, we fix the FR score of synopsis videos generated by different methods so as to fairly compare the performances in terms of CR and NOR. The predefined FR for each video is determined by the output of MAP-VS method, since MAP-VS cannot generate synopsis videos with arbitrary FR. To obtain the same FR scores for all the other methods, we empirically tune the parameters h and P for each approach on each video. For all the methods used in our experiments except **OVS** that can automatically determine the FR, we use grid search to find the desired values of their parameters. The experimental results of CR and NOR are listed in Tabs. II and III, respectively. Some representative synopsis results can be found in Fig. 7, and more synopsis video clips can be found in the following website: http://mic.bjtu.edu.cn/project/video_synopsis/.

From Tabs. II and III, the proposed online video synopsis framework remarkably outperforms all the previous online video synopsis methods, *i.e.*, **MAP-VS**, **HPVC** and **FastPCG**. In particular, the NOR score of our approach reaches up to 0.9896, implying that the synopsis videos generated by our approach, no matter which types of scenarios, have almost no tube collisions. As shown in Fig. 7, the objects appeared at different time intervals are rearranged into the same time interval by our approach. For example, three groups of people occurred at different time locations in original Square-2 video appear in the same time period in the synopsis video (see the left column of Fig. 7). In addition, the proposed method has the capability to cluster tubes with similar trajectories, as shown in groups 1, 2 and 3 in Playground (the middle column of Fig. 7). Furthermore, our approach can effectively preserve the interactions among objects appeared at the same time in the original video. As shown in the right column of Fig. 7, the groups 1 and 2, which both contain several subjects in the original Road-1 video, can be well preserved as groups with fixed spatial and temporal relationships in the synopsis video.

One main reason is that our approach makes full use of context information of tubes to be rearranged. For most of the previous methods, the optimal time location reassigned to a tube is dependent only on previously rearranged tubes, while the information of future tubes is not taken into account. In addition, once the tube is rearranged, its time location

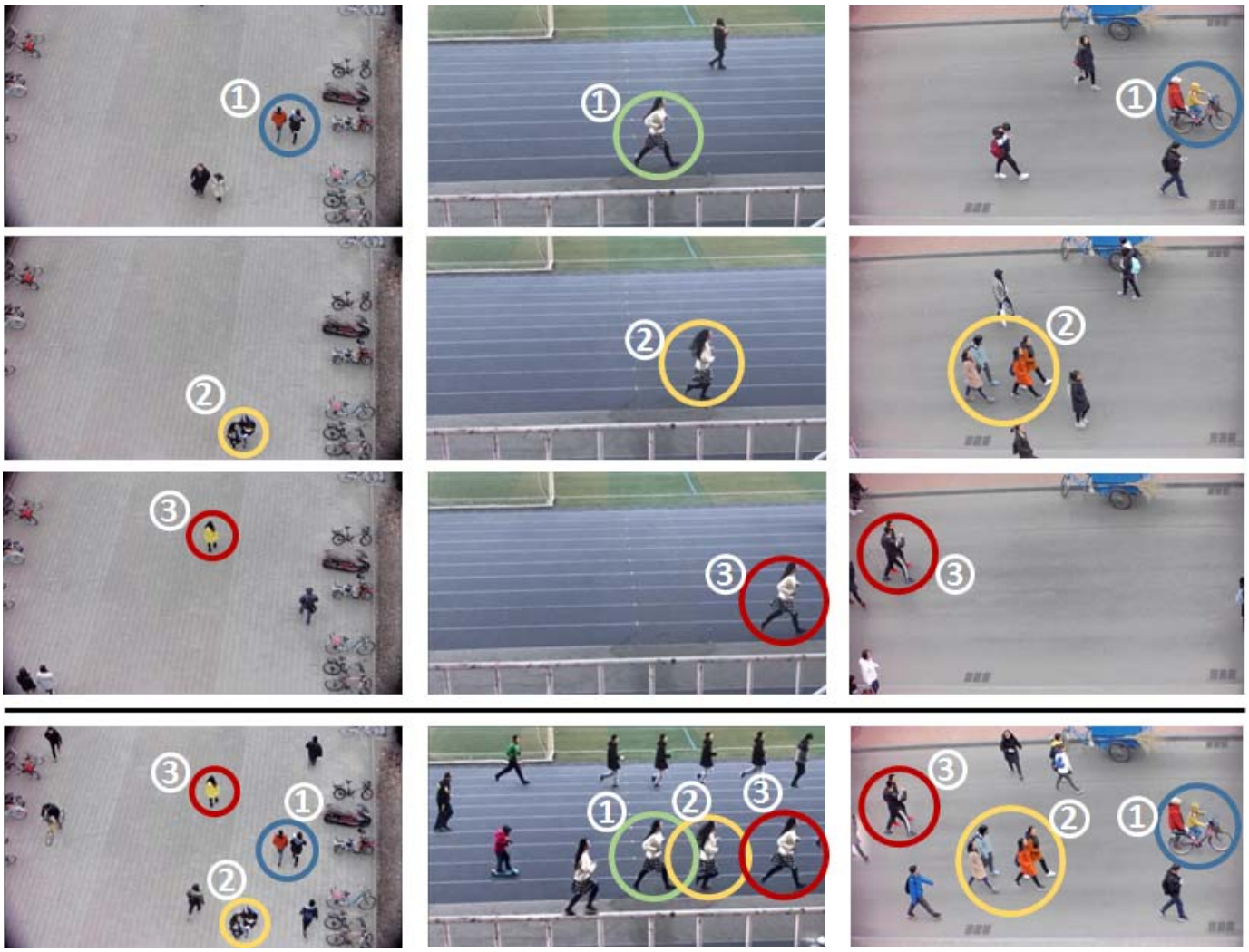

Fig. 7. Representative synopsis results of our approach. The first three rows, from top to bottom, show the representative frames in video streams. The last row demonstrates the synopsis results on Square-2, Playground and Road-1.

is never changed, leading to unsatisfactory synopsis results. Actually, when there comes more tubes, it is often necessary to rearrange previous tubes to more proper time locations so as to obtain a better synopsis effect. In contrast, our approach employs a dynamic graph to keep a certain number of previously rearranged tubes. For each tube in the dynamic graph, its time location is iteratively updated when a new tube is added into the graph. The updating procedure for the tube is continued until it is removed from the dynamic graph. In this way, the previously rearranged tubes get a chance to be rearranged to better time locations, leading to better CR and NOR performance. An exception for these previous online methods is **HPVC**, in which the information for future tubes is also considered by part previously rearranged tubes. However, this method selects only one previously rearranged tube for updating time locations. By using Roulette Wheel Selection algorithm, its solution reaches the local optimization and fail to make full use of the context information. We use the CD metric [41] to experimentally support the above conclusion. Experimental results are listed in Tab. IV. We find that our approach obtains the highest CD values in almost all testing videos. The main reason is that the context before and after a certain tube can be fully explored. In this way, tubes can be rearranged to better time locations, leading to better CR and NOR scores. Meanwhile, the optimal time locations for better CR and NOR may result in a more serious break of the chronological order among tubes. In fact, video synopsis is just a technique that can break the chronological order to get a much compact video for the purpose of efficient indexing and browsing long surveillance videos.

In addition, we find that our online approach outperforms the offline methods that use different batch sizes ($b = 1$ indicates that the offline approaches actually conduct global optimization on the entire video). For example, the overall performance of our approach reaches a CR of 0.1559 and a NOR of 0.9896, while such scores of $\text{OVS}_{b=1}$ reach only 0.1344 and 0.9404, respectively. The main reason is that our proposed method explicitly provides a mechanism for detecting spatial collisions at the frame level. In the proposed method, each object in various tubes is modeled as a vertex in

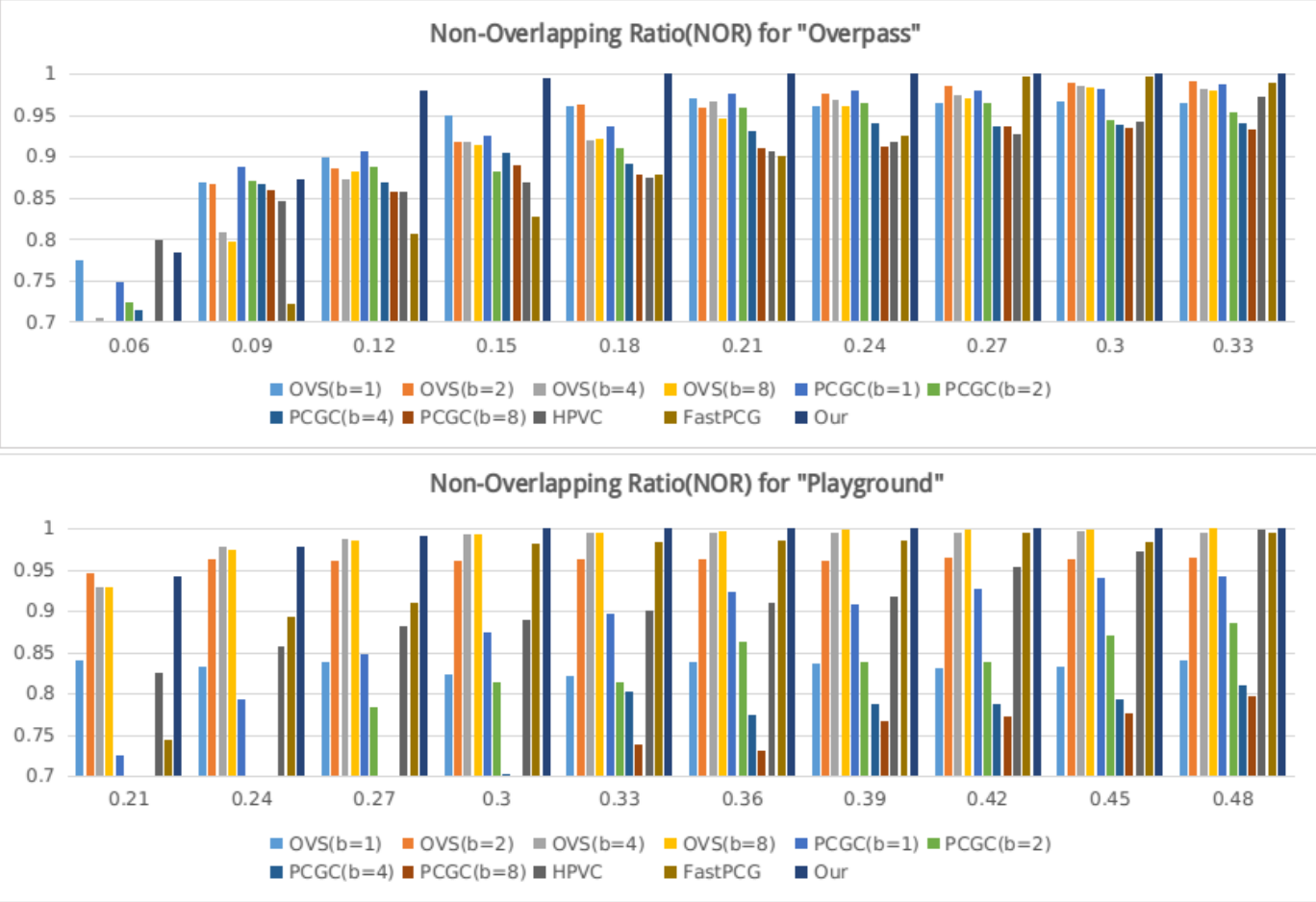

Fig. 8. The non-overlapping ratio (NOR) with different frame condensation ratios (FR) on two representative videos.

TABLE IV
PERFORMANCE COMPARISONS OF CHRONOLOGICAL DISORDER SCORES BETWEEN ONLINE METHODS

Model	MAP-VS	HPVC	FastPCG	Our
Highway	1.72	2.49	5.92	**34.70**
Overpass	12.89	2.39	12.83	**14.91**
Yard	12.48	9.23	36.84	**65.98**
Sidewalk	22.07	2.63	25.28	**59.93**
Playground	29.36	3.60	35.18	**64.80**
Road-1	7.92	3.41	48.26	**50.03**
Road-2	20.94	4.77	**43.51**	42.47
Road-3	18.43	5.33	16.30	**44.81**
Road-4	5.39	2.31	19.51	**34.99**
Square-1	12.66	3.42	7.17	**18.50**
Square-2	13.25	6.41	16.98	**26.88**
Crossroad	23.03	48.03	106.56	**122.07**

the dynamic graph, and the relationship of spatial collisions among objects in different tubes is modeled as the undirected edges. In this way, all the potential collisions are taken into account at the frame level, which, theoretically, can avoid all probable collisions. In contrast, the traditional offline methods evaluate the collisions at tube level, while **OVS** also partially involves the collisions among frames when computing the energy. Therefore, the proposed method can achieve better performance in terms of CR and NOR.

Moreover, we can see that a larger batch size b for **OVS** and **PCGC** leads to worse CR and NOR in most videos. It is consistent with the intuitive expectation since batch processing will break the global optimization problem into several local optimization sub-problems instead. Some exceptions of such phenomenon can be found in Playground and Road-1, which is caused by the fact that these videos have much more tubes. When directly processing so many tubes, the global optimization may lead to unsatisfactory results within a finite number of iterations (*e.g.*, the Simulated Annealing process used in **OVS**). By breaking the large optimization problem into several smaller sub-problems, the optimization process can be conducted to reach several better local optimums instead of an imperfect global optimum, leading to better results.

D. Model Comparison With Flexible Condensation Ratio

From the results reported in Tabs. II and III, we also find an interesting phenomenon that CR and NOR are tightly correlated with each other. That is, a higher CR is often accompanied by a larger NOR, and vice versa. This phenomenon is consistent with the intuitive impression that a synopsis video with less tube collision can make better use of the spatial space, leading to higher CR. Therefore, we focus on the

TABLE V
COMPARISON OF TIME COST IN TERMS OF FRAMES PER SECOND (FPS)

Model (fixed FR)	Highway (0.0679)	Overpass (0.1858)	Yard (0.4351)	Sidewalk (0.2550)	Playground (0.0929)	Road-1 (0.4837)	Road-2 (0.1738)	Road-3 (0.2545)	Road-4 (0.1847)	Square-1 (0.0977)	Square-2 (0.1259)	Crossroad (0.2358)	Average
OVS$_{b=1}$	52	129	9	26	93	7	86	56	98	219	149	6	78
OVS$_{b=2}$	89	160	18	50	195	15	104	68	117	275	182	11	107
OVS$_{b=4}$	105	176	21	61	254	20	127	67	113	265	181	20	118
OVS$_{b=8}$	109	158	23	72	314	24	126	104	94	221	155	24	119
PCGC$_{b=1}$	70	288	11	28	77	7	59	50	79	176	137	5	82
PCGC$_{b=2}$	171	487	33	62	137	17	95	74	101	249	198	10	136
PCGC$_{b=4}$	275	946	128	190	253	29	210	145	236	488	314	24	270
PCGC$_{b=8}$	487	1724	379	528	478	61	477	510	682	970	576	38	576
MAP-VS	190430	158565	25119	55503	175464	6431	31727	23557	21813	60288	70539	8502	68995
HPVC	84056	6965	1667	6027	6715	1034	3010	2471	4560	7612	3206	245	10631
FastPCG	11809	4571	1333	2132	2001	355	1283	1929	3472	7500	3397	137	3327
Our	942	999	474	1159	608	143	430	1218	1427	1750	1571	55	898

NOR scores in subsequent experiments that compare various models with flexible condensation ratios. The experiments are conducted on two videos (*i.e.*, Overpass and Playground) that our approach performs the best (with the highest NOR score) and worst (with the lowest NOR rank) in previous experiments. Note that **MAP-VS** is not compared here since it cannot adjusts the condensation ratio.

In the experiment, we specify several FRs and observe how the NOR changes. As shown in Fig. 8, our approach achieves stable and outstanding performance for both videos even when the FR is flexible. For example, our approach achieves high non-overlapping ratio even when the original video is highly condensed (*i.e.*, a small $FR = 0.12$). This result means that our proposed method can better rearrange tubes to avoid collisions in long and short synopsis videos. Even when the video is highly compressed into a very short clip, our method can still preserve most informative cues of objects.

E. Analysis of Time Complexity

From Algorithm 1 and Algorithm 2, we can observe that the main time cost lies in both $adding$ and $adjusting$ procedures. Once the $Updating$ operation is executed, it invokes above two procedures separately. For $adding$ procedure, we should loop through all the vertices representing the spatial masks of the given tube $\mathcal{T}_{t+1}$, all the undirected edges of each vertex are visited to construct the $NC[\cdot]$. Therefore, the total time is $O(N_{t+1} * N_{E(t+1)})$. Here, $N_{E(t+1)}$ means the average neighbors of each vertex in $\mathcal{T}_{t+1}$. For $adjusting$ procedure, the core operations include the $adding$ and $Queue$ operations. In worst case, the size of $Queue$ is $P-1$. So time complexity in worst case is $O(\sum_{x=1}^{P-1} N_x * N_{Ex})$.

To quantitatively evaluate the time complexity, we list the time cost of the tube rearrangement phase for all methods in Tab. V. We can see that the offline methods and most of their invariants are far inferior to the online methods. However, our method is slower than all online baselines including HPVC. Nevertheless, it still achieves real-time performance, since its slowest rearranging speed is 55 FPS that is greater than the frame ratio of a standard real-time video.

VI. CONCLUSIONS

In this paper, we investigate how to use dynamic strategy to make the synopsis video more compact, and keep as much information as possible in the meanwhile. To explore the problem in a quantitative manner, we develop a novel tube rearranging algorithm based on online dynamic graph coloring, and explicitly formulate a new relationship between moving objects in the input video to help the rearrangement process. Unlike traditional online tube rearranging methods, we will adjust some already placed tubes, and formulate the tube rearranging problem as a dynamic graph coloring problem in the classical graph theory field, and every frame of objects is abstracted into a corresponding graph vertex. Furthermore, to preserve the interactive information between tubes, we stick the tubes which have intersection in the original video. After qualitative and quantitative analysis, we find that our algorithm exhibits greater advantage than five state-of-the-art methods on almost all of the video data.

REFERENCES

[1] D. Reinsel, J. Gantz, and J. Rydning, "Data age 2025: The evolution of data to life-critical," SEAGATE Inc., Cupertino, CA, USA, IDC White Paper, Apr. 2017, p. 1.

[2] S. Wang, Q. Huang, S. Jiang, and Q. Tian, "S^{3}MKL: Scalable semi-supervised multiple kernel learning for real-world image applications," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1259–1274, Aug. 2012.

[3] S. Wang, S. Jiang, Q. Huang, and Q. Tian, "Multi-feature metric learning with knowledge transfer among semantics and social tagging," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2240–2247.

[4] S. Wang, Y. Chen, J. Zhuo, Q. Huang, and Q. Tian, "Joint global and co-attentive representation learning for image-sentence retrieval," in *Proc. ACM Conf. Multimedia*, 2018, pp. 1398–1406.

[5] T. Huang, "Surveillance video: The biggest big data," *Comput. Now*, vol. 7, no. 2, pp. 82–91, 2014.

[6] B. T. Truong and C. Dorai, "Automatic genre identification for content-based video categorization," in *Proc. IEEE Int. Conf. Pattern Recognit.*, vol. 4, Sep. 2000, pp. 230–233.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[8] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 3, no. 1, p. 3, 2007.

[9] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1221–1226, 2006.

[10] H.-W. Kang, Y. Matsushita, X. Tang, and X.-Q. Chen, "Space-time video montage," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 1331–1338.

[11] M. Sun, A. Farhadi, B. Taskar, and S. Seitz, "Salient montages from unconstrained videos," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 472–488.

[12] Z. Li, P. Ishwar, and J. Konrad, "Video condensation by ribbon carving," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2572–2583, Nov. 2009.

[13] H. T. Nguyen, S.-W. Jung, and C. S. Won, "Order-preserving condensation of moving objects in surveillance videos," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2408–2418, Sep. 2016.

[14] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1971–1984, Nov. 2008.

[15] Y. He, C. Gao, N. Sang, Z. Qu, and J. Han, "Graph coloring based surveillance video synopsis," *Neurocomputing*, vol. 225, pp. 64–79, Feb. 2017.

[16] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. Eur. Conf. Comput. Vis.*, 2000, pp. 751–767.

[17] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[18] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[19] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[20] Y. Nie, C. Xiao, H. Sun, and P. Li, "Compact video synopsis via global spatiotemporal optimization," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 10, pp. 1664–1676, Oct. 2013.

[21] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, 2003.

[22] J. Zhu, S. Feng, D. Yi, S. Liao, Z. Lei, and S. Z. Li, "High-performance video condensation system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 7, pp. 1113–1124, Jul. 2015.

[23] S. Feng, Z. Lei, D. Yi, and S. Z. Li, "Online content-aware video condensation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2082–2087.

[24] Y. Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," HP Lab., Imaging Systems Laboratory of HP Labs Palo Alto, Palo Alto, CA, USA, Tech. Rep. HPL-2001-191, 2001.

[25] F. Arman, R. Depommier, A. Hsu, and M.-Y. Chiu, "Content-based browsing of video sequences," in *Proc. 2nd ACM Int. Conf. Multimedia*, 1994, pp. 97–103.

[26] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognit.*, vol. 30, no. 4, pp. 643–658, 1997.

[27] J. Peng and Q. Xiao-Lin, "Keyframe-based video summary using visual attention clues," *IEEE MultiMedia*, vol. 7, no. 2, pp. 64–73, Apr./Jun. 2009.

[28] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting digital movies automatically," *J. Vis. Commun. Image Represent.*, vol. 7, no. 4, pp. 345–353, 1996.

[29] M. A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding," in *Proc. IEEE Int. Workshop Content-Based Access Image Video Database*, Jan. 1998, pp. 61–70.

[30] Y.-F. Ma and H.-J. Zhang, "A model of motion attention for video skimming," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Sep. 2002, p. I.

[31] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a long video short: Dynamic video synopsis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2006, pp. 435–441.

[32] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg, "Webcam synopsis: Peeking around the world," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[33] U. Vural and Y. S. Akgul, "Eye-gaze based real-time surveillance video synopsis," *Pattern Recognit. Lett.*, vol. 30, no. 12, pp. 1151–1159, 2009.

[34] Y. Tian, H. Zheng, Q. Chen, D. Wang, and R. Lin, "Surveillance video synopsis generation method via keeping important relationship among objects," *IET Comput. Vis.*, vol. 10, no. 8, pp. 868–872, 2016.

[35] Y. Pritch, S. Ratovitch, A. Hendel, and S. Peleg, "Clustered synopsis of surveillance video," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Sep. 2009, pp. 195–200.

[36] X. Li, Z. Wang, and X. Lu, "Surveillance video synopsis via scaling down objects," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 740–755, Feb. 2016.

[37] J. Jin, F. Liu, Z. Gan, and Z. Cui, "Online video synopsis method through simple tube projection strategy," in *Proc. IEEE 8th Int. Conf. Wireless Commun. Signal Process.*, Oct. 2016, pp. 1–5.

[38] Y. He, Z. Qu, C. Gao, and N. Sang, "Fast online video synopsis based on potential collision graph," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 22–26, Jan. 2017.

[39] L. Ouerfelli and H. Bouziri, "Greedy algorithms for dynamic graph coloring," in *Proc. IEEE Int. Conf. Commun., Comput. Control Appl.*, Mar. 2011, pp. 1–5.

[40] R. M. R. Lewis, "Designing university timetables," in *A Guide to Graph Colouring*. Cham, Switzerland: Springer, 2016, pp. 195–221.

[41] C.-R. Huang, P.-C. J. Chung, D.-K. Yang, H.-C. Chen, and G.-J. Huang, "Maximum *a posteriori* probability estimationfor online surveillance video synopsis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1417–1429, Aug. 2014.

Tao Ruan received the B.E. degree in computer science and technology from Beijing Jiaotong University, Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree with the Institute of Information Science. His research interests include video synopsis, machine learning, and video parsing.

Shikui Wei received the B.E. degree from Hebei University in 2003, and the Ph.D. degree in signal and information processing from Beijing Jiaotong University (BJTU), Beijing, China, in 2010. From 2010 to 2011, he was a Research Fellow with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He is currently a Full Professor with the Institute of Information Science, BJTU. His research interests include computer vision, image/video analysis and retrieval, and machine learning. More information can be found at http://mic.bjtu.edu.cn.

Jia Li (M'12–SM'15) received the B.E. degree from Tsinghua University in 2005, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2011. He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University, Beijing, China. Before he joined Beihang University, he served in Nanyang Technological University, Peking University, and Shanda Innovations. His research interests include computer vision and multimedia big data, especially the cognitive vision toward evolvable algorithms and models. He has authored or co-authored over 50 technical articles in refereed journals and conferences, such as TPAMI, TIP, IJCV, ICCV, and CVPR. More information can be found at http://cvteam.net.

Yao Zhao received the B.S. degree from the Radio Engineering Department, Fuzhou University, Fuzhou, China, in 1989, the M.E. degree from the Radio Engineering Department, Southeast University, Nanjing, China, in 1992, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1996. He became an Associate Professor at BJTU in 1998 and a Full Professor in 2001. From 2001 to 2002, he was a Senior Research Fellow with the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. He is currently the Director of the Institute of Information Science, BJTU. His research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding. He is also leading several national research projects from the 973 Program, the 863 Program, and the National Science Foundation of China. He serves on the editorial boards of several international journals, including as an Area Editor of *Signal Processing: Image Communication* (Elsevier) and as an Associate Editor of *Circuits, System and Signal Processing* (Springer). He was named a Distinguished Young Scholar by the National Science Foundation of China in 2010.