International Journal of Computer Vision (2025) 133:2653–2669
https://doi.org/10.1007/s11263-024-02302-5

Occlusion-Preserved Surveillance Video Synopsis with Flexible Object Graph

Yongwei Nie[1] · Wei Ge[1] · Siming Zeng[1] · Qing Zhang[2] · Guiqing Li[1] · Ping Li[3,4] · Hongmin Cai[1,5]

Received: 27 February 2024 / Accepted: 5 November 2024 / Published online: 9 December 2024

Abstract

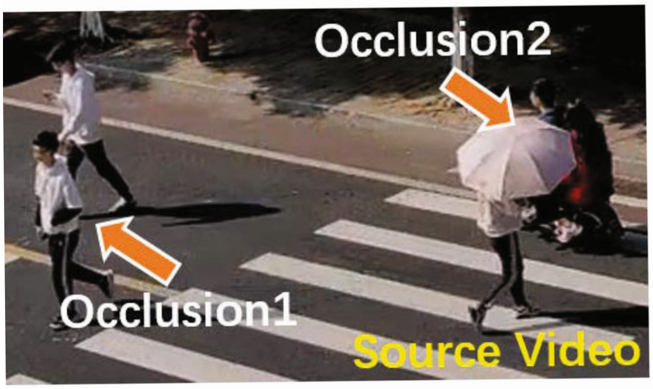
Video synopsis is a technique that condenses a long surveillance video to a short summary. It faces challenges to process objects originally occluding each other in the source video. Previous approaches either treat occlusion objects as a single object, which however reduce compression ratio; or have to separate occlusion objects individually, but destroy interactions between them and yield visual artifacts. This paper presents a novel data structure called Flexible Object Graph (FOG) to handle original occlusions. Our FOG-based video synopsis approach can manipulate each object flexibly while preserving the original occlusions between them, achieving high synopsis ratio while maintaining interactions of objects. A challenging issue that comes with the introduction of FOG is that FOG may contain circulations that yield conflicts. We solve this problem by proposing a circulation conflict resolving algorithm. Furthermore, video synopsis methods usually minimize a multi-objective energy function. Previous approaches optimize the multiple objectives simultaneously which needs to strike a balance between them. Instead, we propose a stepwise optimization strategy consuming less running time while producing higher quality. Experiments demonstrate the effectiveness of our method.

Keywords Video synopsis · Occlusion objects · Graph structure · MCMC

1 Introduction

Nowadays, surveillance videos are widely captured for security concerns. Since most of them are lengthy and redundant, the technique of condensing a surveillance video into a short summary is of great importance to the fast browsing, lightweight storage, and efficient transferring of the vast amount of surveillance videos.

For surveillance videos, a kind of very effective approaches is called "*video synopsis*"



(Rav-Acha et al., 2006; Pritch et al., 2007, 2008, 2009), which extract object tubes and shift tubes forward along the time axis to condense objects into a shorter video space. By "video synopsis", objects that originally appear in different frames are shown in the same frame, leveraging the empty space in the video background and squeezing out redundancy between objects. However, the input videos usually contain

Communicated by Gang Hua.

✉ Hongmin Cai
hmcai@scut.edu.cn

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

[2] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China

[3] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

[4] School of Design, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

[5] School of Future Technology, South China University of Technology, Guangzhou 510641, China

occlusions (see the right inset) between objects posing significant challenges to video synopsis methods.

In practice, “occlusion” is a common phenomenon in surveillance videos due to the perspective projection camera model. Figure 1 illustrates two object tubes with one occlusion point ($x - t$ 2D view). Previous approaches (Pritch et al., 2008; Nie et al., 2012; Li et al., 2009, 2015; Nie et al., 2019) typically group occlusion objects together and view them as a single tube (Fig. 1a). All the objects are shifted as a whole and scaled uniformly. Therefore, the occlusion relationships between them are well preserved, but the flexibility to manipulate each object individually is lost, yielding less compact synopsis videos. On the contrary, if a method pursues a high compression ratio, it shall be able to freely manipulate objects, just like the case in Fig. 1b. This, however, may yield occlusion mismatch artifacts as illustrated in Fig. 1b and by real examples in Fig. 2.

The above analysis shows that there is a contradiction between improving compression ratio and maintaining occlusion relationships. In this paper, we attempt to tackle this dilemma, proposing a video synopsis method supporting manipulation of every object as flexible as possible while preserving occlusion interactions between them.

Our idea is illustrated in Fig. 1c. We divide object tubes into segments according to occlusions between objects. Based on the segments, we build an occlusion graph where each segment is a node of the graph, and occlusion points indicate edges between nodes. To manipulate the graph, we first specify a root node for the graph, which is usually the temporally frontmost segment. We assign a global shifting variable to the root node, and assign each segment with a length-scaling variable. When the scaling value is less than 1.0 (e.g. 0.5), the segment will be shortened (halved). On the contrary, if the scaling value is greater than 1.0 (such as 2.0), the segment will be lengthened (by a factor of 2). We apply a depth-first traversal algorithm to the occlusion graph to update position of the graph. Firstly, the root node is shifted according to the global shifting variable. Then, from the root node, we adjust length of all the segments according to their scaling variables one after another.

The above way of object manipulation has two promising properties. (1) Flexibility: thanks to the local scaling adjustment, segments can be flexibly manipulated. (2) Occlusion-Preserving: since the segments are updated one after another by the depth-first traversal algorithm, the occlusion relationships between objects are preserved.

The occlusion graph works well when it does not contain any circulation, such as the example in Fig. 1. However, when there is a circulation, the depth-first traversal algorithm will encounter the problem of circulation conflict. For example, Fig. 3a shows three objects with three occlusion points. We divide the objects into segments in Fig. 3b where the arrows show the traversal path of the depth-first traversal algorithm. The red arrow indicates a place where a circulation conflict may occur. In Fig. 3c, we show a configuration of scaling variables that indeed yields a conflict. As can be seen, the bottom-left occlusion point is mismatched. In this paper, we propose a circulation conflict resolving algorithm to solve this problem. The idea is illustrated in Fig. 3d and e. The core step is to add as few pseudo occlusions as possible into the graph (Fig. 3d). With the pseudo occlusions, the depth-first traversal algorithm will generate the traversal path in Fig. 3e instead of b. Please pay attention to the green and blue arrows. They will be enforced to have identical scaling variables. In this way, we can synchronize the segments indicated by the green/blue arrows in this example to obtain the result in Fig. 3f without circulation conflict. We call the occlusion graph enhanced by the circulation conflict resolving algorithm as Flexible Object Graph (FOG). Note that Fig. 3e–f just show a simple example to illustrate the basic idea. In the main text, we will describe the circulation conflict resolving algorithm in detail, which can handle more complex situations.

With the well-defined FOGs, our synopsis method optimizes global temporal positions of FOGs and local scaling variables of segments to obtain a synopsis video. The optimization is usually guided by three objectives (Pritch et al., 2008; Nie et al., 2019): maximizing activities condensed into the synopsis, keeping chronological order of objects, and minimizing false collisions. Existing approaches (Pritch et al., 2008; Nie et al., 2012, 2019; Ghatak et al., 2020; Moussa & Shoitan, 2021) usually optimize the multiple objectives simultaneously. Differently, we propose a stepwise optimization strategy based on the MCMC sampling algorithm (Nie et al., 2019). While optimizing the current objective, our method always maintains the goals achieved in previous steps by rejecting MCMC samplings that destroy previously accomplished goals. In this way, we do not need to balance between multiple objectives. Instead, we optimize each objective as much as possible at each step, obtaining better results in less time.

In summary, our contributions include:

- We propose a new data structure, i.e., FOG, for surveillance video synopsis, with a circulation conflict resolving mechanism. Our FOG-based synopsis method can manipulate segments of objects as flexible as possible (thus obtaining higher compression ratio) while preserving occlusion relationships between them (avoiding visual artifacts).
- We propose a stepwise optimization strategy that can generate better synopsis results using less time without the need of striking a balance between multiple optimization objectives.

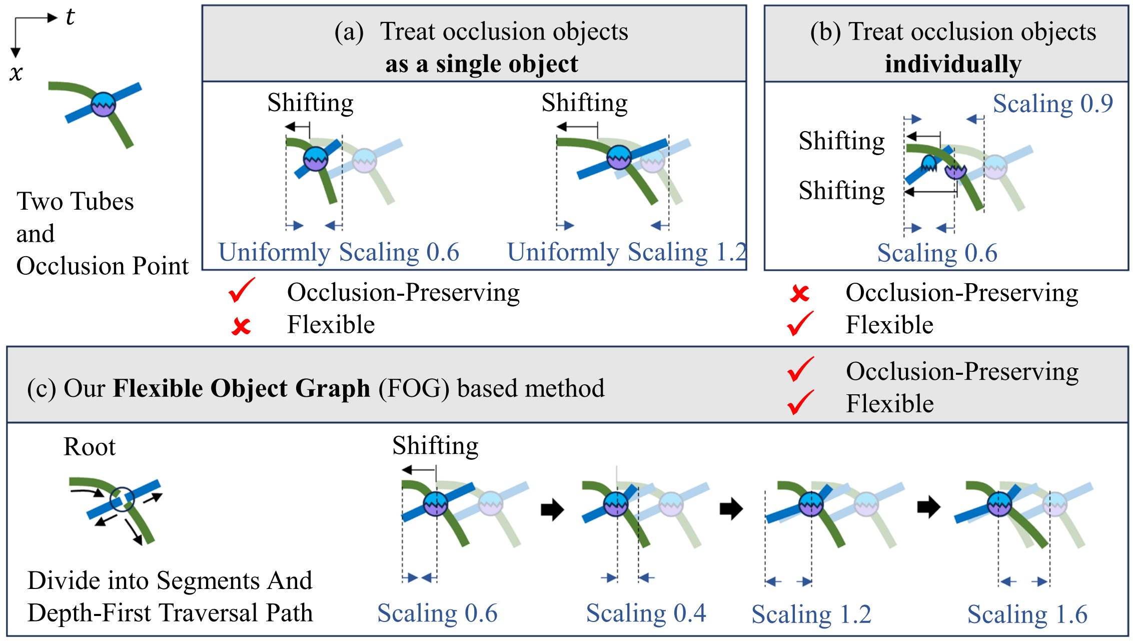


Fig. 1 An illustration of two object tubes with one occlusion point in $x - t$ 2D view. **a** Most of previous approaches treat multiple occlusion objects as a single object which are shifted as a whole and scaled uniformly. The occlusion points are preserved, but the manipulation flexibility is minimized. **b** If shifting and scaling every object individually, the manipulation flexibility is maximized to obtain higher compression ratio, but the occlusion relationships may be corrupted. **c** We divide occlusion tubes into segments, and build an **occlusion graph** based on the segments. We adopt a depth-first traversal algorithm to adjust the length of segments to increase the flexibility of manipulation. The algorithm naturally protects occlusions. Besides, we support shifting the graph as a whole by applying the shifting offset to the root segment

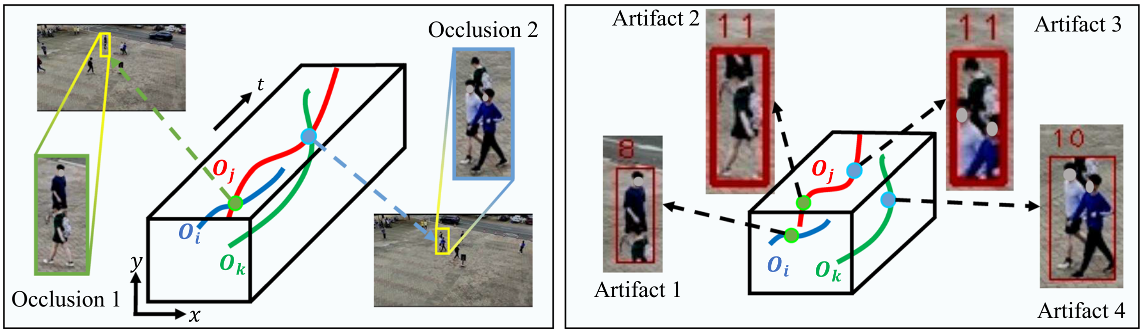


Fig. 2 Left shows an input video with three object tubes occluding at two positions. Right shows the synopsis video with occlusion-destroyed artifacts

- Extensive experiments demonstrate that our method generates high-compression-ratio synopsis results with fewer visual artifacts.

2 Related work

Different ways have been developed for generating video summaries, such as video summarization (Ma & Zhang, 2002; Höferlin et al., 2011; Lee & Grauman, 2015; Kumar et al., 2018; Zhao et al., 2018; Rochan & Waang, 2019; Zhong et al., 2022; Nimmagadda et al., 2023; Negi et al., 2023; Hsu et al., 2023) that extract key frames/shots from input videos, and video montage (Kang et al., 2006; Li et al., 2009; Sun et al., 2014) that stitch spatiotemporal video portions together by seam carving. In this paper, we focus on video synopsis: a technique that is very effective for condensing surveillance videos.

The work of Rav-Acha et al. (2006); Pritch et al. (2007, 2008) presented the first series of video synopsis approaches. They first extract tubes from a surveillance video, then attempt to condense all the extracted tubes into a synopsis video of much shorter length than the input video, by optimizing the temporal positions of the tubes. The optimization

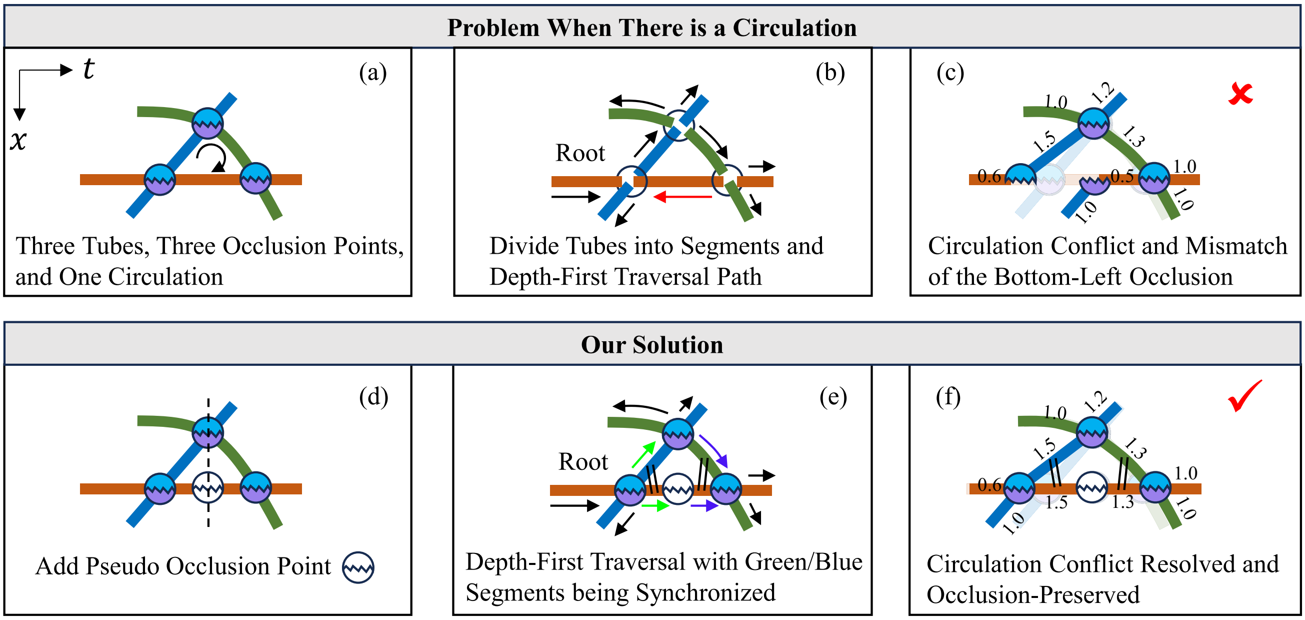

Problem When There is a Circulation
(a)
t
x
Three Tubes, Three Occlusion Points, and One Circulation
(b)
Root
Divide Tubes into Segments and Depth-First Traversal Path
(c)
1.0
1.2
1.5
1.3
0.6
0.5
1.0
1.0
1.0
Circulation Conflict and Mismatch of the Bottom-Left Occlusion
Our Solution
(d)
Add Pseudo Occlusion Point
(e)
Root
Depth-First Traversal with Green/Blue Segments being Synchronized
(f)
1.0
1.2
1.5
1.3
0.6
1.5
1.3
1.0
1.0
1.0
Circulation Conflict Resolved and Occlusion-Preserved

Fig. 3 **a** An illustration of three object tubes with three occlusion points, where there is a circulation in the occlusion graph. **b** The depth-first traversal path where the red arrow indicates the place where a circulation conflict may occur. **c** The numbers on segments are scaling variables with which the circulation conflict problem does occur. **d** To solve the circulation conflict problem, we add a pseudo occlusion point to the bottom tube. **e** We synchronize the segments specified by the green (blue) arrows and update them simultaneously using the depth-first traversal algorithm. **f** The conflict artifact is avoided (Color figure online)

maximizes the number of tubes condensed into the synopsis video, preserves the temporal order of tubes, and prevents false collisions between the tubes. During the past ten more years, lots of attention has been paid to improve these pioneering works from the following aspects.

Object Tube Extraction. Some attention has been paid to the extraction of object tubes. In Pritch et al. (2008); Nie et al. (2012, 2019), objects are extracted by subtracting a background image from video frames. In Lu et al. (2013), foreground objects and their shadows are modeled by a Gaussian mixture model and a texture method, and then tracked through a particle filter. In Huang et al. (2012), objects are tracked using a contrast context histogram-based appearance model and a velocity-based motion model. In their later work Huang et al. (2014), all the appearance and motion models of moving objects are integrated into a MAP formulation for better tracking accuracy. In Zhong et al. (2014); Liao et al. (2017), a 3D graph-cuts algorithm is used for fast object tube extraction in the compression domain of video. Recently, more sophisticated object detection and tracking approaches have been developed, such as KCF (Henriques et al., 2014), STRUCT (Hare et al., 2015), and deep learning-based approaches (Wojke et al., 2017; Wang et al., 2020). Namitha et al. (2022) adopted Yolov3 (Redmon & Farhadi, 2018) for object detection and then Deep-SORT (Wojke et al., 2017) for object tracking across video frames. In this paper, we use the deep-learning tracker of Wang et al. (2020) for tube extraction, obtaining tracked bounding boxes of objects.

Preventing False Collisions. Many approaches aim at reducing false collisions between rearranged object tubes (Pritch et al., 2008; Xu et al., 2008; Nie et al., 2012; Li et al., 2015; Nie et al., 2019; He et al., 2017; Ruan et al., 2019). The work of Pritch et al. (2008) rearranged objects in the temporal domain. Nie et al. (2012) and Zhang et al. (2023) extended the synopsis space into the x and y spatial space. Li et al. (2015) scaled down the size of objects. Nie et al. (2019) changed both speed and size of objects. All the above methods optimize an energy function containing a term measuring the amount of false collisions. Minimizing the energy function thus reduces the false collisions. Instead of performing an optimization, works of He et al. (2016, 2017); Ruan et al. (2019); Pappalardo et al. (2019) pre-computed potential collisions between objects, and build a potential collision graph based on which the video synopsis task is formulated as a graph coloring problem. In this paper, we mainly follow the method of Nie et al. (2019). Our method is different in that the method of Nie et al. (2019) treats occlusion objects as a single tube, and all the objects in the tube are strictly synchronized. In contrast, we synchronize as few as possible segments, while the remaining segments can be manipulated flexibly.

Preserving Object Interactions. Since one of our goals is to maintain occlusions, our method is related to recent synopsis works on preserving object interactions (Li et al., 2018; Yang et al., 2021; Narayanan, 2020; Namitha et al., 2022; Zhang & Zheng, 2023). These approaches adopt sophisticated ways to identify object interactions. For example, Yang et al. (2021) considered depth of objects and scene complexity, Tian et al. (2021) exploited the use of face orientations, Namitha et al. (2022) synthesized bird-eye view, etc. In this paper, we aim at preserving "occlusions" between objects, similar to Ruan et al. (2019); Feng et al. (2012); Zhu et al. (2014); Ahmed et al. (2017). With modern object tracking approaches, occlusions are simple and robust to detect. Since occlusions are very common in videos, maintaining original occlusions is of great importance to the understandability of the synopsis results. After interaction detection, the above approaches group objects to preserve interactions, not supporting manipulation of each of the objects separately. Our method supports flexible operations on individual objects.

Multi-Objective Optimization. Similar to Pritch et al. (2008); Nie et al. (2012, 2019), our method needs to solve an energy function composed of three energy terms. Different algorithms have been used to minimize the energy function. For example in Pritch et al. (2009); Rodriguez (2010), the simulated annealing algorithm is adopted. In Ghatak et al. (2020), a hybrid algorithm using simulated annealing and teaching learning optimization is proposed. In Moussa and Shoitan (2021), the particle swarm optimization method is used. Nie et al. (2019) used the MCMC algorithm to sample solutions. All the above methods minimize the multiple-objective energy function as a whole. Differently, we achieve the multiple objectives step by step. At each step, our method focuses on optimizing the current objective, while maintaining the previously optimized objectives.

There are also work studying online video synopsis (Feng et al., 2012; Zhu et al., 2014; Ra & Kim, 2018; Fu et al., 2014; Thirumalaiah & Immanuel Alex Pandian, 2023), generating synopsis videos according to user queries (Pritch et al., 2009; Ahmed et al., 2019; Lin et al., 2015; Namitha et al., 2022; Shoitan et al., 2023; Priyadharshini & Mahapatra, 2023b), and multi-view video synopsis (Zhu et al., 2015; Hoshen & Peleg, 2015; Mahapatra et al., 2016; Zhang et al., 2019; Priyadharshini & Mahapatra, 2023a; Ingle & Kim, 2023; Priyadharshini & Mahapatra, 2023b), etc. Please refer to the survey (Baskurt & Samet, 2019; Ingle & Kim, 2023) for detailed review.

3 Methodology

Figure 4 illustrates the basic idea of this paper. In (a), we illustrate the 3D view of an example with 6 tubes interacting in 6 occlusion points. For convenience, we transform the 3D view to the 2D view in (b) where a row of horizontal rectangles represent an object tube. The tubes are naturally divided into segments by occlusion intervals. The pair of segments with the same color and connected by a black bidirectional arrow indicates an occlusion between two tubes. Note that the graph in (b) is not naturally a FOG. When there are circulations in (b), we propose a circulation conflict resolving algorithm to enhance (b) to the FOG in (c) by further dividing segments and adding pseudo occlusion relationships (indicated by the gray bidirectional arrows). With the FOG, we can manipulate the segments flexibly while preserving the occlusion relationships. Please see (d) to (g) for several examples. Figure 4 (h) illustrates the synopsis result (3D view) corresponding to the FOG in (g).

In the following, we introduce our method step by step.

3.1 Extracting Tubes, Identifying Occlusions, and Building Occlusion Graph

Object Tube Extraction. As buildingblocks, object tubes are extracted from videos at first. We adopt the deep-learning approach of Wang et al. (2020) to detect and track objects. An object tube is a sequence of bounding boxes enclosing the object. The method of Wang et al. (2020) can help extract most of the tubes, but sometimes produces wrong detection and tracks. For example, a tube may disappear abruptly (lost tracking) at somewhere of the scene. We remove such tubes. Some tubes may first disappear and then reappear with the same ID. We adopt linear interpolation to fill in the missing bounding boxes of these trajectories. Formally, we assume N object tubes $\{O_i | i \in [1, N]\}$ are extracted from a given video. We denote a tube as $O_i = (s_i, e_i)$ starting from frame s_i and stopping at frame e_i.

Occlusions and Segment Splitting. We determine whether two objects occlude each other by checking if they appear in the same frame and whether their bounding boxes overlap. We use $(O_i, O_j, a_{ij}, b_{ij})$ to indicate that object i and j occlude each other from frame a_{ij} to b_{ij}. The occlusions naturally split the object tubes into segments. Sometimes, e.g., in Fig. 5a, two objects may occlude each other during multiple time intervals. We merge them together with a_{ij} denoting the earliest occlusion frame and b_{ij} the latest frame, as shown in Fig. 5b. In Fig. 5c, object O_i occludes both O_j and O_k, and the occlusion segments overlap. We split the occlusion segments at the overlapping boundaries to obtain finer-grained occlusion segments in (d). If multiple objects occlude at the same location, we apply the way in Fig. 5d multiple times. Recall that we have N tubes. Here, we define $\{\mathcal{S}_i^p | i \in [1, N], p \in [1, P_i]\}$ be the divided segments of all the tubes, where $\mathcal{S}_i^p$ is the p^{th} segment of the i^{th} tube, and P_i is the total number of segments of the i^{th} tube.

Building Occlusion Graphs. According to the occlusion relationships, we divide all the tubes $\{O_i | i \in [1, N]\}$ in the source video into M groups, where objects in the same group are connected by occlusions. We build a graph for each group

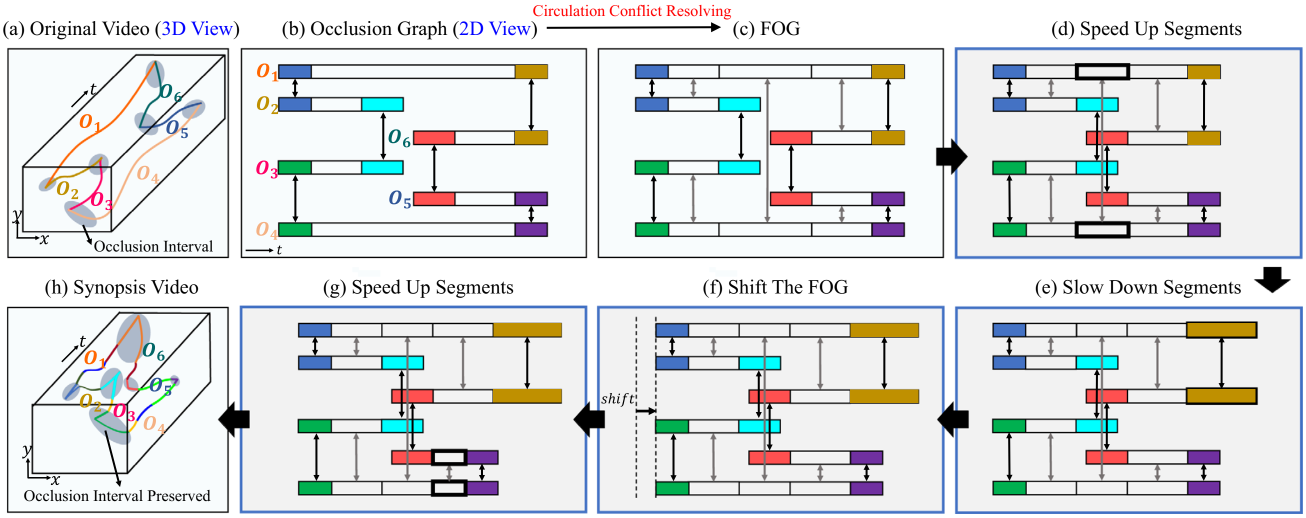


Fig. 4 Overview of the proposed method. **a** Original video (3D view) with 6 tubes interacting in 6 occlusion intervals. **b** Occlusion graph (2D view) of (**a**), where a tube (a row of horizontal rectangles) are divided into segments (i.e., rectangles). The same color rectangles from different tubes connected by a black bidirectional arrow indicate a pair of occlusion segments. **c** The FOG is generated based on **b** by a circulation conflict resolving algorithm which further divides segments when necessary and adds pseudo occlusion connections (indicated by gray arrows). **d**–**g** The FOG can be manipulated by operations including scaling up or slowing down segments, or shifting the whole FOG, without destroying occlusions. **h** The synopsis result corresponding to the FOG in (**g**) (Color figure online)

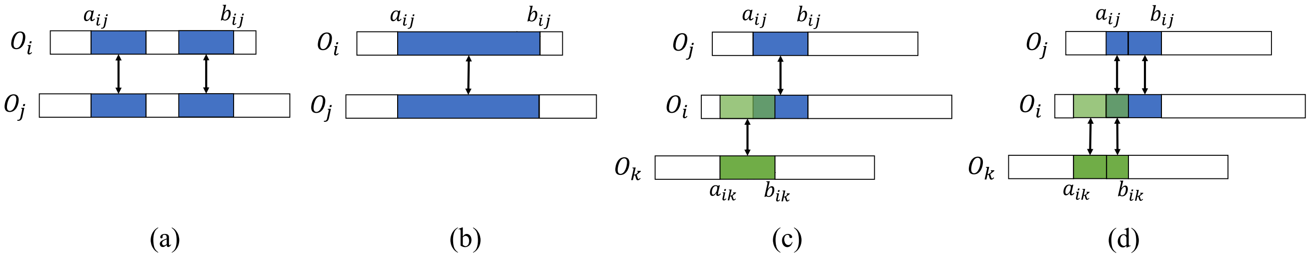


Fig. 5 **a** Objects O_i and O_j occlude in multiple intervals. **b** We merge them into a single interval. **c** Object O_i occludes both O_j and O_k, and the occlusion intervals overlap. **d** We further split the occlusion segments into finer-grained ones at the overlapping boundaries

and call it an occlusion graph. The way to build an occlusion graph is straightforward. We treat each segment as a node of the graph. Then, we add edges between adjacent nodes of the same tube, and also add edges between occlusion segments of different tubes. Finally, we use $\{G_k | k \in [1, M]\}$ to denote all the occlusion graphs.

3.2 Depth-First Traversal Algorithm

We adopt a depth-first traversal algorithm to manipulate an occlusion graph, i.e., change its position so that it can be placed at the appropriate place in the synopsis video. Our strategy is to shift the occlusion graph globally along the time axis and also adjust the length of segments locally. For this, we define global shifting variables for occlusion graphs, denoted by $\Upsilon = \{x_k | k \in [1, M]\}$. We also define scaling variables for segments. Recall that $\{\mathcal{S}_i^p | i \in [1, N], p \in [1, P_i]\}$ are the segments. We use $\Gamma = \{\gamma_i^p | i \in [1, N], p \in [1, P_i]\}$ to denote the scaling variables of the corresponding segments.

Now, we introduce how the depth-first traversal algorithm works. First, we designate the temporally earliest segment of G_k as the root node of the graph (see Fig. 3e for an example). Let O_r be the object containing the root node, $\mathcal{S}_r^1$ be the root node, and (a_r^1, b_r^1) be the time interval of $\mathcal{S}_r^1$ in the source video. The new position of the root node in the synopsis video is calculated as:

$$\begin{cases} \hat{a}_r^1 = a_r^1 + x_k, \\ \hat{b}_r^1 = \hat{a}_r^1 + \gamma_r^1 \times l_r^1 - 1, \end{cases} \tag{1}$$

where l_r^1 is the original length of segment $\mathcal{S}_r^1$. The above equation computes the new start time $\hat{a}_r^1$ by adding the global shifting variable x_k of G_k to the original start time. The new end time is obtained by adding the new length $\gamma_r^1 \times l_r^1$ to the new start time.

Next we update the positions of all the other segments by applying the following depth-first traversal algorithm (please refer to Alg. s1 of the supplementary material for the pseudocode). Assume the current segment is $\mathcal{S}_i^p$ and its position has already been updated to $(\hat{a}_i^p, \hat{b}_i^p)$.

1. For any segment $\mathcal{S}_x^y$ that occludes with $\mathcal{S}_i^p$ and has not yet been visited, its duration is updated to $(\hat{a}_i^p, \hat{b}_i^p)$ which is exactly the same as $\mathcal{S}_i^p$.
2. If $\mathcal{S}_i^{p+1}$ (the segment after $\mathcal{S}_i^p$) exists and has not yet been visited, we update its position to $(\hat{b}_i^p + 1, \hat{b}_i^p + \gamma_i^{p+1} \times l_i^{p+1})$. That is, we simply append $\mathcal{S}_i^{p+1}$ to $\mathcal{S}_i^p$ and change the length of $\mathcal{S}_i^{p+1}$ according to its scaling variable.
3. If $\mathcal{S}_i^{p-1}$ (the segment before $\mathcal{S}_i^p$) exists and has not yet been visited, we update its position to $(\hat{a}_i^p - \gamma_i^{p-1} \times l_i^{p-1}, \hat{a}_i^p - 1)$.

We first update occlusion segments, and then adjacent segments. The order cannot be reversed, otherwise the occlusion segments cannot be synchronized. The above procedure is called recursively until all nodes of the occlusion graph are updated.

3.3 Circulation Conflict Problem and Greedy-based Resolving Algorithm: Improving Occlusion Graph to FOG

When there is no circulation in occlusion graph, the depth-first traversal algorithm works very well. However, when there exist circulations, the algorithm may lead to circulation conflicts. Figure 6a illustrates an occlusion graph in which a circulation (indicated by the arrow from segment A to B) exists. Figure 6b illustrates the circulation conflict problem, in which we extend the length of A to the dashed portion by setting the scaling variable of A to be greater than 1. This triggers the depth-first traversal algorithm to update the positions of all the other segments, which finally yields an overlap between segment A and B which we call a "circulation conflict".

One can solve the conflict problem by properly setting scaling variables in a way that the lengthenings and shortenings of segments exactly cancel out each other, making the last frame of B just placed before the first frame of A. However, this is very challenging to achieve. We propose a different strategy to solve the problem, as shown from Fig. 6c–e. The idea comes from the observation that the conflict can be avoided when all the objects are synchronized. However, this reduces manipulation flexibility. We propose a method to resolve conflicts by synchronizing as few object contents as possible. To this end, we further divide segments in a circulation into finer-grained synchronizable segments (see Fig. 6c) and then add pseudo occlusion edges in a greedy manner (see the gray bidirectional arrows in Fig. 6e).

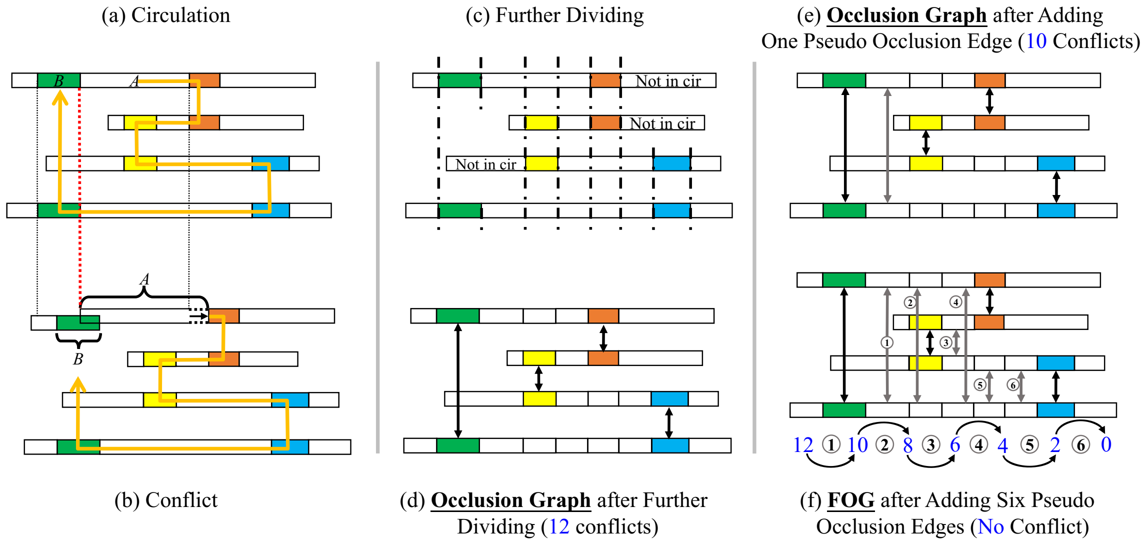
Figure 6c illustrates how we perform the further segment dividing. We draw vertical lines at the boundaries of occlusion segments, and cut the segments of other objects along the vertical lines. Note that only segments within the circulation are divided, while those outside the circulation (i.e., "Not in cir") are not affected. With this procedure, we obtain pairs of synchronizable segments from different objects with the same time interval. Figure 6d shows the occlusion graph after the further dividing. There are 12 conflicts in the graph. The number is obtained by the procedure CountNumberOfConflicts of Alg. s2 in the supplemental material. The idea behind CountNumberOfConflicts is simple. For each segment of a circulation, we increase its length and use the depth-first traversal algorithm to update other segments. If there is a conflict, we increase the number of conflicts by 1.

The remaining problem is how to add pseudo edges into the occlusion graph. We propose a method that adds edges between synchronizable segments in a greedy manner (see Alg. s3 in the supplemental material). For each pair of synchronizable segments, we temporarily add an edge between them and then count the number of conflicts. Among all the edges, we choose the one that yields the largest drop in the number of conflicts and add it into the occlusion graph. After adding one edge, we add the next one using the same greedy procedure. This process proceeds until there is no conflict in the circulation. Figure 6e shows the first pseudo edge added into the occlusion graph. After adding the pseudo edge, the conflict number is decreased from 12 to 10. Figure 6f shows all the pseudo edges added into the occlusion graph, where the circled numbers on each edge indicate the order in which they were added. The bottom of (f) shows how the number of conflicts decreases after adding each pseudo edge. The finally obtained occlusion graph in (f) is called a Flexible Object Graph (FOG) with no conflict problem.

3.4 FOG-based Video Synopsis

As stated above, we have extracted N tubes: $\{O_i | i \in [1, N]\}$, and established M FOGs: $\{G_k | k \in [1, M]\}$. With the global shifting variables $\Upsilon = \{x_k | k \in [1, M]\}$ and local length scaling variables $\Gamma = \{\gamma_i^p | i \in [1, N], p \in [1, P_i]\}$, we can manipulate all the FOGs by using of the above depth-first traversal algorithm. Let $\{\hat{O}_i | i \in [1, N]\}$ be the objects after manipulation. Following energy terms in Pritch et al. (2008), our FOG-based video synopsis method is going to minimize the following objective function:

$$E(\Upsilon, \Gamma) = \sum_{i=1}^{N} E_A(\hat{O}_i) + \sum_{i,j}^{N} E_T(\hat{O}_i, \hat{O}_j) + E_C(\hat{O}_i, \hat{O}_j). \quad (2)$$


(a) Circulation

(b) Conflict

(c) Further Dividing

(d) **Occlusion Graph** after Further Dividing (12 conflicts)

(e) **Occlusion Graph** after Adding One Pseudo Occlusion Edge (10 Conflicts)

(f) **FOG** after Adding Six Pseudo Occlusion Edges (No Conflict)

Fig. 6 **a** An example consisting of four objects with complex occlusions. The arrow from segment A to B shows a circulation. **b** Slightly extending the length of segment A to the dashed portion results in a conflict between segments A and B. **c** Further dividing the segments in the circulation along the vertical timelines at the boundaries of these segments. **d** The occlusion graph after the further dividing. There are 12 possible conflict positions in the graph. **e** Greedily adding the first pseudo edge (the gray bidirectional arrow) decreases conflict number from 12 to 10. **f** Six pseudo occlusion edges are added into the occlusion graph (see the circled numbers on the edges for the order in which they were added), yielding a FOG with no conflict problem. The blue numbers at the bottom show how the number of conflicts decreases as the number of pseudo edges increases (Color figure online)

The activity term E_A measures how many objects are not in the synopsis video:

$$E_A(\Upsilon, \Gamma) = \sum_{i=1}^{N} \sum_{k=1}^{l_i} \delta(\hat{k} < 0 \, or \, \hat{k} \geq L_{syn}) \phi(O_i(k)), \tag{3}$$

where δ is a function returns 1 if its input is true and 0 otherwise, L_{syn} is the length of the synopsis video, $\hat{k}$ indexes the frame of $\hat{O}_i$ that corresponds to the frame k of O_i, and $\phi(O_i(k))$ denotes the amount of activity of O_i at frame k.

The chronological term E_T measures the degree of corruption of chronological order between FOGs. Let G_m and G_n be two FOGs in the input video, and a_m and a_n denote the frontmost frames of them. Let $\hat{G}_m$ and $\hat{G}_n$ be the corresponding FOGs in the synopsis video, and $\hat{a}_m$ and $\hat{a}_n$ be the corresponding frames. We define E_T as:

$$E_T(\Upsilon, \Gamma) = \sum_{m,n}^{M} \delta\left((a_m - a_n) \cdot (\hat{a}_m - \hat{a}_n) < 0\right)$$
$$\cdot \cdot \, \delta\left(|\hat{a}_m - \hat{a}_n| > \tau\right) \eta \cdot |\hat{a}_m - \hat{a}_n|, \tag{4}$$

where the first δ checks whether the chronological order of two FOGs is reversed and the second δ checks if the reversal is larger than τ frames. Only when both of the two conditions are met, we increase the energy loss by $\eta \cdot |\hat{a}_m - \hat{a}_n|$, where $\eta = 100$.

The collision term E_C measures the amount of false collisions between objects:

$$E_C(\Upsilon, \Gamma) = \sum_{i,j}^{N} \sum_{k_1, k_2} \delta(\hat{k}_1 = \hat{k}_2) \psi\left(\hat{O}_i(\hat{k}_1), \hat{O}_j(\hat{k}_2)\right), \tag{5}$$

where $k_1 \in [1, l_i]$ and $k_2 \in [1, l_j]$ traverse frames of O_i and O_j, respectively, $\hat{k}_1$ and $\hat{k}_2$ index the corresponding frames of $\hat{O}_i$ and $\hat{O}_j$, and $\hat{O}_i(\hat{k}_1)$ and $\hat{O}_j(\hat{k}_2)$ are the occurrences of $\hat{O}_i$ and $\hat{O}_j$ at frames $\hat{k}_1$ and $\hat{k}_2$ in the synopsis video, finally $\psi(\cdot)$ computes the collision area between the two objects.

3.5 Stepwise MCMC Solver

The energy function defined in Eq. 2 combines three sub-objectives that contradict each other. It is not easy to strike a balance when simultaneously optimizing the three sub-objectives. We propose a stepwise algorithm to optimize them one by one.

Step 1: Preserving all objects. In the first step, we only optimize the activity term E_A. Previous approaches usually require E_A to be as small as possible. Differently, we assume all objects are important (e.g., when the objects are obtained by anomaly detection method (Liu et al., 2021)) and put all of them into the synopsis video, i.e., minimizing E_A to zero. To achieve this, we move FOGs ahead along the time axis to the very beginning of the synopsis video. Then, we uniformly scale all the segments of the FOGs to compress them into the synopsis video.

Step 2: Recovering chronological order of objects. In the second step, while keeping all FOGs staying in the synopsis video, we attempt to optimize the FOGs to recover the chronological order between them. Specifically, we use the Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) to sample from the following Boltzmann density function defined over the variables $\Theta = \{\Upsilon, \Gamma\}$ of FOGs (please refer to the pseudocode Alg. s4 in the supplemental material):

$$p(\Theta) = \frac{1}{Z}\exp(-\beta E_T(\Theta)), \tag{6}$$

where β is the temperature and Z is the partition function. MH algorithm samples from the distribution of $p(\Theta)$ by constructing a Markov chain composed of a set of states of FOGs whose equilibrium distribution is $p(\Theta)$. Let Θ^t be the current state. To obtain the next state of the Markov chain, we first change Θ^t to a new state Θ^* by a proposal distribution $q(\Theta^*|\Theta^t)$. Then, we compute the probability α:

$$\alpha(\Theta^t \rightarrow \Theta^*) = \min\left(1, \frac{p(\Theta^*)q(\Theta^t|\Theta^*)}{p(\Theta^t)q(\Theta^*|\Theta^t)}\right). \tag{7}$$

Let $u \sim [0, 1]$. The original MH algorithm accepts the new state and sets $\Theta^{t+1} = \Theta^*$ if $u < \alpha(\Theta^t \rightarrow \Theta^*)$. We modify the acceptance condition, i.e., we accept the new state only if $u < \alpha(\Theta^t \rightarrow \Theta^*)$ and $E_A(\Theta^*) = 0$. Otherwise, we reject the proposal and set Θ^{t+1} to the old state Θ^t. The second condition help ensure that all the objects are still maintained in the synopsis video while recovering chronological order.

For the proposal function $q(\Theta^*|\Theta^t)$ that modifies Θ^t to a new state Θ^*, it is usually designed to be reversible, i.e., $q(\Theta^*|\Theta^t) = q(\Theta^t|\Theta^*)$ which simplifies Eq. 7 to:

$$\alpha(\Theta^t \rightarrow \Theta^*) = \min\left(1, \frac{p(\Theta^*)}{p(\Theta^t)}\right). \tag{8}$$

Specifically, we modify the old state Θ^t to a new state Θ^* using one of the following two ways that are reversible: (1) Randomly choose two FOGs and exchange their positions. (2) Randomly choose a segment, perturb its scaling variable by a Gaussian distribution $\mathcal{N}(0, \sigma^2)$. We constrain the scaling variable to be in the range of $[\gamma_{min}, \gamma_{max}]$. That is, when a scaling variable is smaller than γ_{min}, we set it to γ_{min}, and if it is greater than γ_{max}, we set it to γ_{max}. The MH algorithm stops when $E_T = 0$, i.e., when the chronological order of FOGs are all recovered. We find this is easy to achieve as the false collision term E_C is not considered at this time.

Step 3: Reducing false collisions between objects. In this step, we optimize FOGs to reduce the false collisions between objects as much as possible. We use another MH algorithm (see Alg. s5 in the supplemental material) to sample from the following Boltzmann density function:

$$p(\Theta) = \frac{1}{Z}\exp(-\beta E_C(\Theta)). \tag{9}$$

We use the same way to propose a new state Θ^* and compute $\alpha(\Theta^t \rightarrow \Theta^*)$ according to Eq. 8. We accept the new state only when the following three conditions are met: (1) $u < \alpha(\Theta^t \rightarrow \Theta^*)$, (2) $E_A(\Theta^*) = 0$, and (3) $E_T(\Theta^*) = 0$. The MH algorithm stops when E_C is reduced to zero or after T iterations.

Determining γ_{min} and γ_{max} adaptively. γ_{min} and γ_{max} determine the range of speed scaling variables. We determine the two parameters adaptively according to the original speed of objects. The original speed $\bar{v}$ of an object is defined as the average number of pixels moved by the upper-left corner of the object's bounding box between two consecutive frames. We compute γ_{min} by:

$$\gamma_{min} = \min\left(1.0, \frac{\bar{v}}{\lambda_1}\right), \tag{10}$$

where λ_1 with the default value of 4.5 is a user-defined parameter which controls the maximum speed of an object. γ_{max} is computed by:

$$\gamma_{max} = \max\left(1.0, \frac{\bar{v}}{\frac{l}{\lambda_2} + \epsilon}\right), \tag{11}$$

where l is the original length of the object, λ_2 is a user-defined parameter and its default value is 100.0, and ϵ is set to 1.0 to prevent γ_{max} from being too large when l is small. When calculating γ_{max}, we take into account the length of the object, such that a long object will not become very longer.

4 Experiments

We implement our method in C++, and run it on an AMD Ryzen$^{\text{TM}}$ 5 2600 CPU and 16 G memory. In the following, we introduce experimental settings, compare our method with previous approaches, and conduct ablation studies about the key components of our method. *Please see the supplemental materials for more results.*

4.1 Experimental Settings

4.1.1 Dataset

Since the resolution of experimental videos in Huang et al. (2014) are too low to extract tubes and the works of Ruan et al. (2019) and SSOcT (Yang et al., 2021) do not make

Table 1 Information of test videos, including video name, original length (Org. Len.), frame rate (Fps), number of objects (Obj. Num.), and resolution (Res.) of the corresponding video

Name	Org. Len	fps	Obj. Num	Res
video-entry1	110326 (1h1'17")	30	819	1280×720
video-entry2	111645 (1h2'1")	30	693	1280×720
video-entry3	30795 (20'31")	25	456	640×368
video-gogo	20026 (2'47")	120	20	1024×576
video-st	10109 (6'44")	25	91	640×368
video-lib	3068 (1'42")	30	17	1920×1080
video-zgf	1038 (41")	25	19	960×540
video-a1	1846 (1'13")	25	49	960×540

their experimental videos public, we capture 8 videos by ourselves whose information is shown in Table 1. We also use videos from MOT17 (Dendorfer et al., 2021), MOT20 (Dendorfer et al., 2020), and DanceTrack (Sun et al., 2022) for the comparison which are put into the supplemental material. The scenes captured include parking lot, square, road, etc., which are common in a city. All videos contain busy activities of pedestrians or cars. Two videos among them are longer than 1 h, containing up to 819 objects. In these videos, occlusions between objects occur frequently, especially in MOT20-03 (Dendorfer et al., 2020), MOT20-05 (Dendorfer et al., 2020) and DanceTrack (Sun et al., 2022). The tube extraction method we used for ours and compared methods is always JDE (Wang et al., 2020).

4.1.2 Evaluation Metrics

We evaluate a video synopsis method using the following four metrics. (1) **Compression Ratio (CR).** Let L_{src} be the number of frames of the input video, and L_{syn} be the length of a synopsis video. CR is defined as CR $= L_{syn}/L_{src}$. (2) **Outside Activity (OA).** We use OA to represent the amount of activities that are not condensed into the synopsis video. (3) **Chronological Disorder Number (CDN).** We count the number of object pairs whose temporal order is reversed by at least τ frames. (4) **Collision Artifact (CA).** CA measures the amount of false collisions in a synopsis video. We use the method defined in Eq. 5 to compute CA.

4.1.3 Parameter Setting

There are three parameters in our method. One is τ that controls the tolerance of the user about the degree of reversal of chronological order between objects. It is set as $\tau = 400$ unless otherwise specified. Another parameter is L_{syn} which is the length of the synopsis video in frames. For example, we can set L_{syn} to be 1/10 of the input video length, achieving a compression ratio (CR) of 10%. Finally, there is a parameter T denoting the maximum MCMC iteration number of step 3 in Sec. 3.5. We set $T = 20,000$ unless otherwise specified.

4.2 Comparison with Previous Approaches

We compare our method with recent approaches including SSOcT (Yang et al., 2021), PCCVA (Namitha et al., 2022), and FSF (Zhang & Zheng, 2023). We also compare our method with (Nie et al., 2019) in the ablation study section.

SSOcT (Yang et al., 2021) uses a global octree structure to represent the synopsis space and views each tube as an tube octree to perform fast and accurate collision detection of tubes. The rearrangement of tubes is completed by repeatedly calling the "insert" and "refine" process. PCCVA (Namitha et al., 2022) divides the synopsis space into cubes and groups tubes. The start time of each tube group in the synopsis is determined by cube voting. FSF (Zhang & Zheng, 2023) also divides tubes into tube sets and then adopts a frame sequence fusion algorithm which takes tube set as the processing unit to rearrange tubes. All the three methods do not provide code. We implement them by ourselves in C++, and conduct the comparisons on the same computer under the same running environment.

Table 2 gives the comparisonal results. There are some points to note. First, the first three videos are very long and contain a large number of tubes. For long videos, instead of processing them as a whole, we split them into clips, process clips one by one, and finally combine the summary results of clips together. The other videos are not that long, so we process them as a whole.

Second, the parameters used in the comparisons are given in the table. SSOcT (Yang et al., 2021), PCCVA (Namitha et al., 2022) and FSF (Zhang & Zheng, 2023) do not have parameter τ. We view them as having the parameter because for fair comparison the evaluation of CDN of these approaches is based on τ too, i.e., only when the start time of two objects are reversed by at least τ frames, we increase the number of CDN by 1. For the compared methods, their parameters β^*, d^* and ζ^* are also shown, where β^* is the key parameter of SSOcT (Yang et al., 2021) denoting the maximum number of collisions allowed when inserting a tube, d^* is the key parameter of PCCVA (Namitha et al., 2022) which

Table 2 Comparisons between SSOcT (Yang et al., 2021), PCCVA (Namitha et al., 2022), FSF (Zhang & Zheng, 2023), and our method. "-org. speed" and "-speed up" indicate with and without speed change of tubes, respectively. β^*, d^* and ζ^* are parameters of SSOcT (Yang et al., 2021), PCCVA (Namitha et al., 2022), and FSF (Zhang & Zheng, 2023), respectively. OA, CDN, and CA are three metrics measuring the quality of the synopsis results, which are the smaller the better

Data	Method	Parameters		OA↓	CDN↓	CA↓
		τ	L_{syn} (CR)			
video-entry1	SSOcT Yang et al. (2021)-org. speed (β^*=342)	400	11016 (10.0%)	**0**	22000	3.75e+08
	SSOcT Yang et al. (2021)-speed up (β^*=13)			**0**	20436	6.44e+06
	PCCVA Namitha et al. (2022)-org. speed (d^*=30)			**0**	34772	4.25e+08
	PCCVA Namitha et al. (2022)-speed up (d^*=35)			**0**	32263	9.06e+07
	FSF Zhang and Zheng (2023)-speed up (ζ^*=2.2e−04)			**0**	1435	8.20e+07
	Our			**0**	**0**	**2.77e+06**
video-entry2	SSOcT Yang et al. (2021)-org. speed (β^*=291)	400	11160 (10.0%)	**0**	12935	1.45e+08
	SSOcT Yang et al. (2021)-speed up (β^*=11)			**0**	12269	3.40e+06
	PCCVA Namitha et al. (2022)-org. speed (d^*=29)			**0**	13125	3.36e+08
	PCCVA Namitha et al. (2022)-speed up (d^*=40)			**0**	11443	9.94e+07
	FSF Zhang and Zheng (2023)-speed up (ζ^*=1.12e−04)			**0**	410	2.81e+07
	Our			**0**	**0**	**3.28e+06**
video-entry3	SSOcT Yang et al. (2021)-org. speed (β^*=2043)	400	3110 (10.0%)	**0**	2580	2.05e+08
	SSOcT Yang et al. (2021)-speed up (β^*=65)			**0**	4948	9.83e+06
	PCCVA Namitha et al. (2022)-org. speed (d^*=5)			**0**	10494	4.77e+07
	PCCVA Namitha et al. (2022)-speed up (d^*=10)			**0**	16773	7.38e+06
	FSF Zhang and Zheng (2023)-speed up (ζ^*=2.47e−03)			**0**	7958	1.45e+07
	Our			**0**	**0**	**396031**
video-gogo	SSOcT Yang et al. (2021)-org. speed (β^*=21)	500	1500 (7.5%)	**0**	**0**	58037
	SSOcT Yang et al. (2021)-speed up (β^*=2)			**0**	1	500
	PCCVA Namitha et al. (2022)-org. speed (d^*=14)			**0**	**0**	54362
	PCCVA Namitha et al. (2022)-speed up (d^*=14)			**0**	26	40079
	FSF Zhang and Zheng (2023)-speed up (ζ^*=1e-05)			**0**	6	3736
	Our			**0**	**0**	**0**
video-st	SSOcT Yang et al. (2021)-org. speed (β^*=700)	200	800 (7.9%)	**0**	63	4.21e+07
	SSOcT Yang et al. (2021)-speed up (β^*=10)			**0**	179	312985
	PCCVA Namitha et al. (2022)-org. speed (d^*=5)			**0**	1311	2.83e+07
	PCCVA Namitha et al. (2022)-speed up (d^*=14)			**0**	1668	2.13e+06
	FSF Zhang and Zheng (2023)-speed up (ζ^*=4.40e−04)			**0**	47	2.06e+06
	Our			**0**	**0**	**0**
video-lib	SSOcT Yang et al. (2021)-org. speed (β^*=150)	160	500 (16.3%)	**0**	2	1.65e+06
	SSOcT Yang et al. (2021)-speed up (β^*=13)			**0**	14	223740
	PCCVA Namitha et al. (2022)-org. speed (d^*=5)			**0**	**0**	1.87e+06
	PCCVA Namitha et al. (2022)-speed up (d^*=5)			**0**	21	231575
	FSF Zhang and Zheng (2023)-speed up (ζ^*=4.50e−04)			**0**	6	344558
	Our			**0**	**0**	**0**
video-zgf	SSOcT Yang et al. (2021)-org. speed (β^*=165)	70	220 (21.2%)	**0**	**0**	618145
	SSOcT Yang et al. (2021)-speed up (β^*=13)			**0**	3	139473
	PCCVA Namitha et al. (2022)-org. speed (d^*=5)			**0**	5	1.90e+06
	PCCVA Namitha et al. (2022)-speed up (d^*=5)			**0**	23	370043
	FSF Zhang and Zheng (2023)-org. speed (ζ^*=1.10e−03)			**0**	10	1.62e+06
	FSF Zhang and Zheng (2023)-speed up (ζ^*=3.00e−04)			**0**	4	109253
	Our			**0**	**0**	**0**

Table 2 continued

Data	Method	Parameters τ	L_{syn} (CR)	OA↓	CDN↓	CA↓
video-a1	SSOcT Yang et al. (2021)-org. speed (β*=300)	120	400 (21.7%)	**0**	5	9.41e+06
	SSOcT Yang et al. (2021)-speed up (β*=60)			**0**	19	1.95e+06
	PCCVA Namitha et al. (2022)-org. speed (d*=5)			**0**	44	2.35e+06
	PCCVA Namitha et al. (2022)-speed up (d*=5)			**0**	97	948056
	FSF Zhang and Zheng (2023)-org. speed (ζ*=9.00e−04)			**0**	143	8.75e+06
	FSF Zhang and Zheng (2023)-speed up (ζ*=5.00e−04)			**0**	33	2.73e+06
	Our			**0**	**0**	**0**

Bold indicates the best result among all the others

is the cube size, and ζ^* is from FSF (Zhang & Zheng, 2023) which is the maximum mean collision ratio in each frame.

Third, our method speeds up objects adaptively, while the compared approaches do not. To be fair, we compare with two versions of previous approaches: an original-speed version and speed-up version. For the speed-up version, we compute the average speed of the objects in our result and then use this speed to accelerate all the objects uniformly which are then processed by previous approaches. For FSF (Zhang & Zheng, 2023), we only provide the speed-up results for the first six testing videos, as the original-speed method cannot achieve the high compression ratio of 10%, 7.5%, 7.9% and 16.3%.

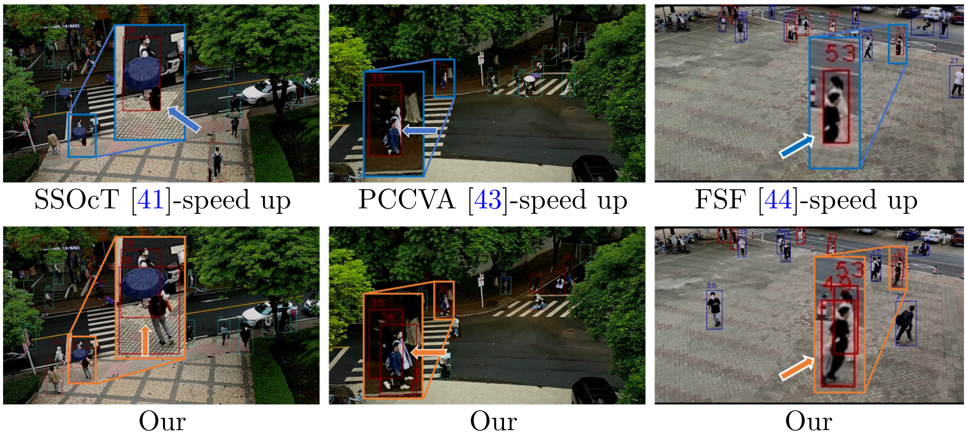
As seen in Table 2, all methods reduce OA to zero, but our method produces much fewer CA and CDN than the compared approaches. First of all, speed is critical. For example, the results generated by "SSOcT (Yang et al., 2021)-speed up" contain much less CA and CDN than "SSOcT (Yang et al., 2021)-org. speed", as do "PCCVA (Namitha et al., 2022)-speed up" and "PCCVA (Namitha et al., 2022)-org. speed". The main reason is that the length of tubes is shortened in "SSOcT (Yang et al., 2021)-speed up" and "PCCVA (Namitha et al., 2022)-speed up". Second, in this fair manners of comparison, i.e., when the speed of tubes is set to be equal to the average one in our method, "SSOcT (Yang et al., 2021)-speed up", "PCCVA (Namitha et al., 2022)-speed up" and "FSF (Zhang & Zheng, 2023)-speed up" generate higher CDN and CA than ours. Taking "video-entry2" as an example, FSF (Zhang & Zheng, 2023) obtains 2.81e+07 CA, while our proposed method achieves 3.28e+06 CA, which is approximately 1/8 of that of FSF (Zhang & Zheng, 2023). More importantly, our method reduces chronological disorder number (CDN) to 0, which means that our method is able to preserve appearance orders of objects perfectly. Instead, the CDN of FSF (Zhang & Zheng, 2023) on "video-entry2" is 410, which means that the synopsis video generated by FSF (Zhang & Zheng, 2023) contains 410 object pairs that are reversed temporally. The higher CDN of FSF (Zhang & Zheng, 2023) indicates that the sequence of events of objects in the synopsis video does not follow the original timeline, which severely disrupt the natural flow and relationships between objects of the input video, increasing the difficulty of understanding the synopsis result.

As can be seen in Table 2, our method reduces CDN to 0 on all test videos. This is because, in step 2 of the stepwise MCMC solver, the MH algorithm stops only when E_T = 0, i.e., when the chronological orders of objects are all recovered. This is easily to achieve since we do not take the false collision term E_C (Eq. 5) into account and allow the chronological order of objects to be reversed within a specified threshold τ. Moreover, by dividing objects into segments and manipulating object segments in the FOG, our method reduces CA significantly compared to other methods. Besides lower CDN and CA, another advantage of our method is that it can preserve original occlusions between objects, while the compared three methods are not always able to achieve this. This characteristic of our method greatly increases the readability of our results, and avoid annoying visual artifacts. Figure 7 shows images demonstrating this advantage, and more results are shown in our provided supplemental video.

Through the above comparisons, the effectiveness of our method on both long and short videos is validated. More comparisons on three MOT datasets (MOT17 (Dendorfer et al., 2021), MOT20 (Dendorfer et al., 2020) and DanceTrack (Sun et al., 2022)) can be seen in the supplemental material.

4.3 Ablation Study On FOG

Our work improves (Nie et al., 2019) by proposing the FOG data structure. Both our method and (Nie et al., 2019) groups originally occluded objects into the same tube, and therefore we can both preserve original occlusions in the synopsis video. The difference is that we rely on FOG to adaptively adjust the speed of each object. In contrast, Nie et al. (2019) synchronizes all segments of occlusion objects, thus having lower flexibility. We conduct the following comparisons between our method and (Nie et al., 2019) to validate the



SSOcT [41]-speed up PCCVA [43]-speed up FSF [44]-speed up

Our Our Our

Fig. 7 Comparisons between SSOcT Yang et al. (2021)-speed up, PCCVA Namitha et al. (2022)-speed up, FSF Zhang and Zheng (2023)-speed up and our method on preserving original occlusion relationships between objects. From left to right: video-entry1, video-entry2, and video-entry3. Results by SSOcT Yang et al. (2021)-speed up, PCCVA Namitha et al. (2022)-speed up and FSF Zhang and Zheng (2023)-speed up contain artifacts (indicated by blue arrows). Our results preserve original occlusion relationships between objects (orange arrows). See the supplemental video for dynamic results (Color figure online)

Table 3 Comparison with Nie et al. (2019), demonstrating that our method with FOG outperforms Nie et al. (2019)

		Parameters			OA↓	CDN↓	CA↓
		τ	$[\gamma_{min}, \gamma_{max}]$	L_{syn} (CR)			
video-entry1	Nie et al. (2019)	400	*adaptively*	11016 (10.0%)	**0**	33	3.39e+0.6
	Our				**0**	**0**	**2.77e+06**
video-entry2	Nie et al. (2019)	400	*adaptively*	11160 (10.0%)	**0**	13	7.86e+06
	Our				**0**	**0**	**3.28e+06**
video-entry3	Nie et al. (2019)	400	*adaptively*	3110 (10.0%)	**0**	87	1.77e+06
	Our				**0**	**0**	**396031**
video-gogo	Nie et al. (2019)	500	[0.1, 10]	1500 (7.5%)	**0**	4	404651
	Our				**0**	**0**	**0**
video-st	Nie et al. (2019)	200	[0.1, 10]	800 (7.9%)	**0**	**0**	111652
	Our				**0**	**0**	**0**
video-lib	Nie et al. (2019)	160	[0.2, 5]	500 (16.3%)	**0**	**0**	13157
	Our				**0**	**0**	**0**
video-zgf	Nie et al. (2019)	70	[0.2, 5]	220 (21.2%)	**0**	8	126523
	Our				**0**	**0**	**0**
video-a1	Nie et al. (2019)	120	[0.2, 5]	400 (21.7%)	**0**	25	76136
	Our				**0**	**0**	**0**

Bold indicates the best result among all the others

effectiveness of the core contribution of this paper, i.e, the FOG data structure.

Originally in Nie et al. (2019), a MCMC strategy is used to minimize an objective function composed of multiple terms. Differently, we propose a stepwise MCMC solver. Besides, the definition of objective functions in Nie et al. (2019) and our method are slightly different. For fair comparison, we re-implement (Nie et al., 2019) using the same energy terms defined in this paper, and use our stepwise MCMC solver to compute results for (Nie et al., 2019). Besides modifying motion speed, the method of Nie et al. (2019) additionally modifies object size. We set the size adjustment range of Nie et al. (2019) to [0.7, 1], though this is slightly unfair to our method as our method does not change object size. The method of Nie et al. (2019) also adopts parameters γ_{min} and γ_{max} as the range of segment scaling variables. They are set in the same way as ours.

The ablation results are shown in Table 3. For the first three long videos, γ_{min} and γ_{max} are adjusted dynamically, and the compression ratio is 1/10. Although our method and (Nie et al., 2019) have similar CA, our method can reduce CDN to zero, while the method of Nie et al. (2019) produces

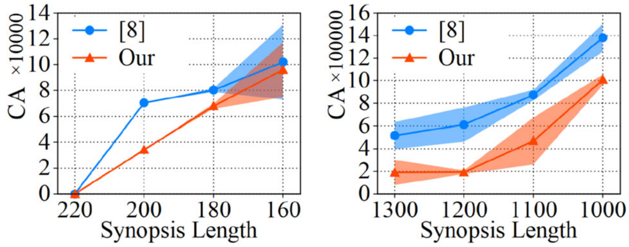


Fig. 8 Comparisons with Nie et al. (2019). From left to right: video-zgf, video-gogo. For each video, synopsis results of different lengths are generated. Curves of CA (the smaller the better) of the two methods with respect to the synopsis length are plotted

much higher CDN. For the other short videos, we manually set τ, L_{syn} and $[\gamma_{min}, \gamma_{max}]$ to make our method generate perfect synopsis results, i.e., OA=0, CDN=0, and CA as few as possible. For all these videos, the OA of Nie et al. (2019) is also reduced to zero, but we observe temporal order corruption or false collision artifacts in the results of Nie et al. (2019). For example for video-a1, the CDN and CA of Nie et al. (2019) are 25 and 76136 respectively, while ours are both zero. We observe that the CDN of Nie et al. (2019) is sometimes reduced to zero, but there is always a certain amount of CA, which demonstrates the challenging of avoiding false collision by Nie et al. (2019). In contrast, our method can better reduce false collision, mainly attributed to the proposed FOG-based segment adaptive adjustment algorithm. An additional point worthy noting from the experiments in Table 3 is that our method can achieve very high compression ratios. For example, CR for video-gogo of our method is 7.5%, and CR for video-st is 7.9%. These are very high CRs, considering there is no artifact in our results.

In Fig. 8, we compare our method with Nie et al. (2019) by checking how each method performs as L_{syn} decreases (i.e., the CR increases). We test both methods multiple times, and the shaded area indicates variance of CA. With FOG, our method consistently outperforms (Nie et al., 2019). In most cases, our worst result is better than the best result of Nie et al. (2019).

In Fig. 9, we show how our method preserves chronological order of objects better than (Nie et al., 2019). In each row, there are three frames selected from the synopsis result generated by the corresponding method, and the index of each frame is displayed above the corresponding frame. In the first row, an exception in chronological order is that objects 41, 42 and 43 (see frame 88) appear too earlier than objects 17, 18, and 19 (see frame 251). In contrast, the second row show that objects in our results generally appear in their original chronological order.

4.4 Comparison on Running Time

Table 4 compares between the time used by SSOcT (Yang et al., 2021), PCCVA (Namitha et al., 2022), FSF (Zhang & Zheng, 2023) and our method. When compressing long videos, PCCVA (Namitha et al., 2022) needs to divide the synopsis video space into a large number of cubes, and then calculate the vote of each cube on each tube group, so it takes much more time than other three methods. Since SSOcT (Yang et al., 2021) introduces octree structures to accelerate collision detection, it takes less time than PCCVA (Namitha et al., 2022) and our method. The collision detection of FSF (Zhang & Zheng, 2023) is based on the tube sets which means the number of collision detection units it processes is fewer, thus it takes the least amount of time. Our method reduces false collisions through an iterative MCMC optimization algorithm, and needs to perform depth-first traversal algorithm many times in each iteration. Therefore, the time spent depends on the number of iterations and the number of objects in the input video. Under the premise of ensuring that synopsis videos have fewer artifacts, our method takes a reasonable amount of time between SSOcT (Yang et al., 2021) and PCCVA (Namitha et al., 2022).

Table 5 compares between the time used by our approach at different optimization steps. The first and second steps are very efficient, since the first step simply puts FOGs into the synopsis video, while the second step allows the existence of false collisions. In the third step, since the chronological order must be preserved, it is not easy to reduce the false collision artifacts, and the optimization algorithm needs more MCMC iterations to sample a good result, thus costing more time.

Detailed time complexity analysis can be found in the supplemental material. The conclusion is that the more objects, the more time our method takes. For example, the total time used for “video-st” with 91 (see Table 1) objects is 482.732s, while for “video-lib” with only 17 objects the time used is 45.440s. Splitting long videos into clips with fewer objects is a way to solve this problem.

4.5 Limitations

The proposed method can generate high-compression synopsis results with fewer artifacts. A shortfall is that it is a little time-consuming to compute these results. Our currently implemented MCMC algorithm for reducing false collisions costs most of the time, which makes our method can only be used in an offline manner. In the future, we can accelerate the optimization process by simplifying the computation of the collision term defined in Eq. 5, e.g., using more high-level proxy of objects instead of bounding boxes. Or, we can implement the MCMC sampling in parallel. In addition, our method is sensitive to the quality of the extracted object tubes.

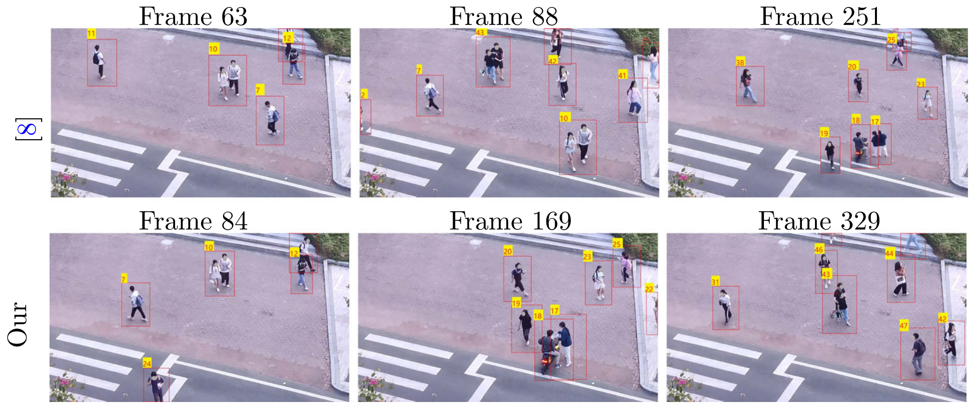


Fig. 9 Comparison with Nie et al. (2019) on preserving chronological order of objects of video-a1. We show frame index on the top of each image. The number in the yellow box on top of each person indicates the order of that person appears in the input video. For Nie et al. (2019), frame 88 contains the 43th object, while frame 251 contains the 17th object, which is a large corruption of chronological order. In contrast, our method does not show such problem (Color figure online)

Table 4 Time comparisons between SSOcT Yang et al. (2021), PCCVA Namitha et al. (2022), FSF Zhang and Zheng (2023), and our method

Data	SSOcT Yang et al. (2021)	PCCVA Namitha et al. (2022)	FSF Zhang and Zheng (2023)	Our
video-entry1 (1h1’17”)	981.6s	17049.7s	**47.3s**	1791.7s
video-entry2 (1h2’1”)	2840.4s	16202.7s	**58.4s**	2856.4s
video-entry3 (20’31”)	97.0s	26285.0s	**21.1s**	2175.4s

Bold indicates the best result among all the others

Table 5 Time used by our method at different optimization steps

Data	Step1	Step2	Step3	Total
video-gogo	0.005s	0.055s	71.140s	71.200s
video-st	0.041s	0.523s	482.168s	482.732s
video-lib	0.005s	0.014s	45.421s	45.440s
video-zgf	0.039s	0.006s	186.080s	186.125s
video-a1	0.041s	0.076s	333.274s	333.391s

Some of the tubes used in this paper are post-processed manually. This problem is possible to be solved with the advance of object detection and tracking algorithms.

Finally, due to space limit, we put intermediate results of the stepwise optimization process into the supplemental material. We also analyze the difference between our proposed stepwise solver and prior optimizers at there.

5 Conclusion and Future Work

This paper presents a novel video synopsis approach. When generating synopsis videos, how to handle the interactions/occlusions between moving objects is a key challenge. We have addressed this challenging problem with the main contribution of the novel data structure called “FOG” that supports flexible object manipulation while preserving the interaction relationships between them. With FOG, we can remove redundancy among objects with complex occlusion relationships. As far as we know, no previous approach can achieve that. Based on FOG, this paper presents an effective FOG-based video synopsis approach, which in its essence is a multi-objective optimization problem. In order to solve the optimization problem, our second contribution is the stepwise MCMC optimization strategy by which we achieve each objective step by step. Our FOG-based video synopsis approach, together with the stepwise optimization method, rewards us with impressive synopsis results better than those of previous approaches. In the future, we will accelerate the current implementation, exploring the possibility of making our method work online.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11263-024-02302-5.

Acknowledgements The work was supported by the Fundamental Research Funds for the Central Universities (No. 2024ZYGXZR021), the National Key Research and Development Program of China (2024YFF1206600); the Natural Science Foundation of China (No. U21A20520, 62325204,62072191, 62472180, 62471499), the Key-Area Research and Development Program of Guangzhou City (202206030009), and the Guangdong Basic and Applied Basic Research Foundation (No.2024A1515011995).

Data Availibility The datasets that support the findings of this study have been uploaded to the public data repository Science Data Bank for the research purpose.

References

Ahmed, A., Kar, S., Dogra, D. P., Patnaik, R., Lee, S., Choi, H., & Kim, I. (2017). Video synopsis generation using spatio-temporal groups. In *ICSIPA*, pp. 512–517. IEEE.

Ahmed, S. A., Dogra, D. P., Kar, S., Patnaik, R., Lee, S.-C., Choi, H., Nam, G. P., & Kim, I.-J. (2019). Query-based video synopsis for intelligent traffic monitoring applications. *IEEE Transactions on Intelligent Transportation Systems, 21*(8), 3457–3468.

Baskurt, K. B., & Samet, R. (2019). Video synopsis: A survey. *Computer Vision and Image Understanding, 181*, 26–38.

Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., & Leal-Taixé, L. (2020). Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003.

Dendorfer, P., Osep, A., Milan, A., Schindler, K., Cremers, D., Reid, I., Roth, S., & Leal-Taixé, L. (2021). Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision, 129*, 845–881.

Feng, S., Lei, Z., Yi, D., & Li, S. Z. (2012). Online content-aware video condensation. In *CVPR*, pp. 2082–2087. IEEE.

Fu, W., Wang, J., Gui, L., Lu, H., & Ma, S. (2014). Online video synopsis of structured motion. *Neurocomputing, 135*, 155–162.

Ghatak, S., Rup, S., Majhi, B., & Swamy, M. (2020). An improved surveillance video synopsis framework: a HSATLBO optimization approach. *Multimedia Tools and Applications, 79*(7), 4429–4461.

Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M. M., Hicks, S. L., & Torr, P. H. (2015). Struck: Structured output tracking with kernels *IEEE Transactions on Pattern Analysis and Machine Intelligence,38*(10), 2096–2109.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika, 57*(1), 97–109.

He, Y., Gao, C., Sang, N., Qu, Z., & Han, J. (2017). Graph coloring based surveillance video synopsis. *Neurocomputing, 225*, 64–79.

Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2014). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37*(3), 583–596.

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.

Höferlin, B., Höferlin, M., Weiskopf, D., & Heidemann, G. (2011). Information-based adaptive fast-forward for visual surveillance. *Multimedia Tools and Applications, 55*(1), 127–150.

Hoshen, Y., & Peleg, S. (2015). Live video synopsis for multiple cameras. In *ICIP*, pp. 212–216. IEEE.

Hsu, T. C., Liao, Y. S., & Huang, C. R. (2023). Video summarization with spatiotemporal vision transformer. *IEEE Transactions on Image Processing, 32*, 3013–3026.

Huang, C. R., Chen, H. C., & Chung, P. C. (2012). Online surveillance video synopsis. In *ISCAS*, pp. 1843–1846. IEEE.

Huang, C. R., Chung, P. C. J., Yang, D. K., Chen, H. C., & Huang, G. J. (2014). Maximum a posteriori probability estimation for online surveillance video synopsis. *IEEE Transactions on circuits and systems for video technology, 24*(8), 1417–1429.

Ingle, P. Y., & Kim, Y.-G. (2023). Multiview abnormal video synopsis in real-time. *Engineering Applications of Artificial Intelligence, 123*, 106406.

Ingle, P. Y., & Kim, Y. G. (2023). Video synopsis algorithms and framework: A survey and comparative evaluation. *Systems, 11*(2), 108.

Kang, H. W., Matsushita, Y., Tang, X., & Chen, X. Q. (2006). Space-time video montage. In *CVPR*, vol. 2, pp. 1331–1338. IEEE.

Kumar, K., Shrimankar, D. D., & Singh, N. (2018). Eratosthenes sieve based key-frame extraction technique for event summarization in videos. *Multimedia Tools and Applications, 77*, 7383–7404.

Lee, Y. J., & Grauman, K. (2015). Predicting important objects for egocentric video summarization. *International Journal of Computer Vision, 114*, 38–55.

Liao, W., Tu, Z., Wang, S., Li, Y., Zhong, R., & Zhong, H. (2017). Compressed-domain video synopsis via 3d graph cut and blank frame deletion. In *Proceedings of the on Thematic Workshops of ACM Multimedia*, pp. 253–261.

Li, Z., Ishwar, P., & Konrad, J. (2009). Video condensation by ribbon carving. *IEEE Transactions on Image Processing, 18*(11), 2572–2583.

Lin, W., Zhang, Y., Lu, J., Zhou, B., Wang, J., & Zhou, Y. (2015). Summarizing surveillance videos with local-patch-learning-based abnormality detection, blob sequence optimization, and type-based synopsis. *Neurocomputing, 155*, 84–98.

Liu, Z., Nie, Y., Long, C., Zhang, Q., & Li, G. (2021). A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *ICCV*, pp. 13588–13597.

Li, X., Wang, Z., & Lu, X. (2015). Surveillance video synopsis via scaling down objects. *IEEE Transactions on Image Processing, 25*(2), 740–755.

Li, X., Wang, Z., & Lu, X. (2018). Video synopsis in complex situations. *IEEE Transactions on Image Processing, 27*(8), 3798–3812.

Lu, M., Wang, Y., & Pan, G. (2013). Generating fluent tubes in video synopsis. In *ICASSP*, pp. 2292–2296. IEEE.

Ma, Y. F., & Zhang, H. J. (2002). A model of motion attention for video skimming. In *ICIP*, vol. 1, p. IEEE

Mahapatra, A., Sa, P. K., Majhi, B., & Padhy, S. (2016). Mvs: A multi-view video synopsis framework. *Signal Processing: Image Communication, 42*, 31–44.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics, 21*(6), 1087–1092.

Moussa, M. M., & Shoitan, R. (2021). Object-based video synopsis approach using particle swarm optimization. *Signal, Image Video Process, 15*(4), 761–768.

Namitha, K., Geetha, M., & Athi, N. (2022). An improved interaction estimation and optimization method for surveillance video synopsis. *IEEE MultiMedia*, 1–13.

Namitha, K., Narayanan, A., & Geetha, M. (2022). Interactive visualization-based surveillance video synopsis. *Applied Intelligence, 52*(4), 3954–3975.

Narayanan, A., et al. (2020). Preserving interactions among moving objects in surveillance video synopsis. *Multimedia Tools and Applications, 79*(43), 32331–32360.

Negi, A., Kumar, K., & Saini, P. (2023). Object of interest and unsupervised learning-based framework for an effective video summarization using deep learning. *IETE Journal of Research, 70*(5), 5019–5030.

Nie, Y., Li, Z., Zhang, Z., Zhang, Q., Ma, T., & Sun, H. (2019). Collision-free video synopsis incorporating object speed and size changes. *IEEE Transactions on Image Processing, 29*, 1465–1478.

Nie, Y., Xiao, C., Sun, H., & Li, P. (2012). Compact video synopsis via global spatiotemporal optimization. *IEEE Transactions on Visualization and Computer Graphics, 19*(10), 1664–1676.

Nimmagadda, P., Sudhakar, K., Rajasekar, P., & et al. (2023). Perceptual video summarization using keyframes extraction technique. In *ICIPTM*, pp. 1–4. IEEE.

Pappalardo, G., Allegra, D., Stanco, F., & Battiato, S. (2019). A new framework for studying tubes rearrangement strategies in surveillance video synopsis. In *ICIP*, pp. 664–668. IEEE.

Pritch, Y., Ratovitch, S., Hendel, A., & Peleg, S. (2009). Clustered synopsis of surveillance video. In *ICAVSS*, pp. 195–200. IEEE.

Pritch, Y., Rav-Acha, A., Gutman, A., & Peleg, S. (2007). Webcam synopsis: Peeking around the world. In *ICCV*, pp. 1–8. IEEE.

Pritch, Y., Rav-Acha, A., & Peleg, S. (2008). Nonchronological video synopsis and indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(11), 1971–1984.

Priyadharshini, S., & Mahapatra, A. (2023a). Mohasa: A dynamic video synopsis approach for consumer-based spherical surveillance video. *IEEE Transactions on Consumer Electronics*.

Priyadharshini, S., & Mahapatra, A. (2023b). A personalized video synopsis framework for spherical surveillance video. *CSSE, 46*(1), 2603–2616.

Ra, M., & Kim, W.-Y. (2018). Parallelized tube rearrangement algorithm for online video synopsis. *IEEE Signal Processing Letters, 25*(8), 1186–1190.

Rav-Acha, A., Pritch, Y., & Peleg, S. (2006). Making a long video short: Dynamic video synopsis. In *CVPR*, vol. 1, pp. 435–441. IEEE.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.0276.

Rochan, M., & Wang, Y. (2019). Video summarization by learning from unpaired data. In *CVPR*, pp. 7902–7911.

Rodriguez, M. (2010). Cram: Compact representation of actions in movies. In *CVPR*, pp. 3328–3335. IEEE.

Ruan, T., Wei, S., Li, J., & Zhao, Y. (2019). *Rearranging online tubes for streaming video synopsis: A dynamic graph coloring approach, 28*(8), 3873–3884.

Shoitan, R., Moussa, M. M., Gharghory, S. M., Elnemr, H. A., Cho, Y.-I., & Abdallah, M. S. (2023). User preference-based video synopsis using person appearance and motion descriptions. *Sensors, 23*(3), 1521.

Sun, P., Cao, J., Jiang, Y., Yuan, Z., Bai, S., Kitani, K., & Luo, P. (2022). Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20993–21002.

Sun, M., Farhadi, A., Taskar, B., & Seitz, S. (2014). Salient montages from unconstrained videos. In *ECCV*, pp. 472–488. Springer.

Thirumalaiah, G., & Immanuel Alex Pandian, S. (2023). An optimized complex motion prediction approach based on a video synopsis. *IJIUS11*(1), 88–95.

Tian, Q., Zhu, Z., Wang, C., Wang, P., Guo, J., & Wang, Y. (2021). A video synopsis method for object interactive preservation combined with face orientation. In *ISKE*, pp. 491–496. IEEE.

Wang, Z., Zheng, L., Liu, Y., Li, Y., & Wang, S. (2020). Towards real-time multi-object tracking. In *ECCV*, pp. 107–122. Springer.

Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *ICIP*, pp. 3645–3649. IEEE.

Xu, M., Li, S.Z., Li, B., Yuan, X. T., & Xiang, S. M. (2008). A set theoretical method for video synopsis. In *MIR*, pp. 366–370.

Yang, Y., Kim, H., Choi, H., Chae, S., & Kim, I.-J. (2021). Scene adaptive online surveillance video synopsis via dynamic tube rearrangement using octree. *IEEE Transactions on Image Processing, 30*, 8318–8331.

Zhang, Y., Guo, K., & Zheng, T. (2023). Surveillance video synopsis based on spatio-temporal offset. *Journal of Electronic Imaging, 32*(1), 013013–013013.

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 Transactions on Image Processing, 29*, 971–985.

Zhang, Y., & Zheng, T. (2023). Object interaction-based surveillance video synopsis. *Applied Intelligence, 53*, 4648–4664.

Zhao, B., Li, X., & Lu, X. (2018) Hsa-rnn: Hierarchical structure-adaptive RNN for video summarization. In *CVPR*, pp. 7405–7414.

Zhong, R., Hu, R., Wang, Z., & Wang, S. (2014). Fast synopsis for moving objects using compressed video. *IEEE Signal Processing Letters, 21*(7), 834–838.

Zhong, S.-H., Lin, J., Lu, J., Fares, A., & Ren, T. (2022). Deep semantic and attentive network for unsupervised video summarization. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 18*(2), 1–21.

Zhu, J., Feng, S., Yi, D., Liao, S., Lei, Z., & Li, S. Z. (2014). High-performance video condensation system. *IEEE Transactions on Circuits and Systems for Video Technology, 25*(7), 1113–1124.

Zhu, J., Liao, S., & Li, S. Z. (2015). Multicamera joint video synopsis. *IEEE Transactions on Circuits and Systems for Video Technology, 26*(6), 1058–1069.

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