TABLE OF CONTENTS

ABSTRACT **1**

1 Introduction **3**

1.1 Motivation 3

1.2 Related works 5

1.3 Dissertation overview 6

2 Problem formulation of video synopsis **7**

2.1 Activity energy 8

2.2 Time-lapse background generation 11

2.3 Background consistency energy 12

2.4 Temporal consistency energy 13

2.5 Collision energy 15

2.6 Computational bottleneck 16

3 Proposed tube rearrangement **18**

3.1 Occupation matrix generation 18
3.1.1 Binary occupation matrix 19
3.1.2 Probabilistic occupation matrix 20
3.2 Objective function 20
3.2.1 Reformulated collision energy 20
3.2.2 Length energy 23
3.3 Optimizing objective function 24
3.3.1 Properties of accumulated occupation matrix 26
3.4 Parallelized optimization 26

4 Online video synopsis framework **29**

4.1 Background modeling 29
4.2 Object tube generation 34
4.3 Object stitching 36
4.4 Discontinuity of motion flow 38

5 Experimental Results **41**

5.1 Performance metrics 41
5.2 Test video sequences 43
5.3 Performance analysis 45
5.3.1 Optimum parameters 45
5.3.2 Ablation study for speed up techniques 61
5.3.3 Performance comparisons 64

6 Conclusion **69**

BIBLIOGRAPHY **80**

LIST OF FIGURES

2-1 Concept diagram of video synopsis [1]. The bird and man appeared at different time in the original video are rearranged in temporal domain, and then displayed simultaneously in the condensed video. 9

2-2 Example of the object tube rearrangement in 2D space. Red arrows indicate some offsets of the starting labels for better understanding of the tube rearrangement process. We can see that after the tube rearrangement, the length of the condensed video becomes much shorter than that of the original. . . . 10

2-3 Flowchart of the proposed online video synopsis framework. At the beginning, foreground of the object tube is reshaped into the 3D occupation matrix. We use this matrix representation to calculate the collision energy fast in conjunction with parallel processing and Fourier transform. Afterwards, we determine optimal starting labels for tubes and stitch tubes with the background to generate a resulting synopsis video. Illustrations of man and vehicle in this figure are created by Lluisa Iborra and Yasser Megahed from the Noun Project. . . 17

3-1 Example of the binary occupation matrix generation when $\mathcal{W}/\mathcal{N} = 16$ and $\mathcal{H}/\mathcal{M} = 9$. The foreground and background of the object are represented in black and white, respectively. Dotted lines in the figure are depicted to show contours of the original object for the readers. Illustrations of the woman and man in this figure are created by Nataliia Lytvyn and Ludovic Gicqueau from the Noun Project, respectively. 21

3-2 Example calculation of reformulated collision energy E_c with two binary occupation matrices. Occupied elements in the matrix are colored in red and blue. After the element-wise multiplication, we can see that the objects have two collided elements colored in magenta. In this figure, $\odot$ is an operator for the element-wise multiplication, also known as Hadamard product. 22

3-3 Two ways of calculating E_c. All of occupation matrices in this figure have $6 \times 8 \times 3$ spatio-temporal resolution. E_c can be calculated by using (a) Hadamard products between two sets of frames, and (b) 1D cross correlations between 48 pairs of 1D signals. Three primitive colors (red, green, and blue) in this figure is used to show example correspondences. 28

4-1 Flowchart of foreground segmentation by Lim *et al.* [2]. It follows the common process of the image segmentation, but its performance has been increased by incorporating feature maps from multiple scales. 30

4-2 Recent framework of foreground segmentation via GAN [3]. It utilizes three generator-discriminator pairs to make foreground segmentation robust against extreme illumination changes. At first, input images undergo the gamma correction to make them in the extreme illumination condition. Red G_b and D_b boxes to generate synthetic bright images from dark images, and to discriminate the synthetic and original ones. On the other hand, blue G_b and D_b boxes do the same task for the bright images. Finally, by using the results of red and blue G_b, G_s and D_s are trained to generate and discriminate segmented foregrounds. 32

4-3 First example of seamless cloning using Poisson image editing. All images in this figure are from the work of Pérez *et al.* [4]. 35

4-4 Second example of seamless cloning using Poisson image edting. Unlike the first example, there are three regions from two source images for one destination image. All images in this figure are from the work of Pérez *et al.* [4]. . . 35

4-5 Notations used in Poisson image editing [4]. 36

4-6 text 37

4-7 Proposed online video synopsis framework. The framework generates a partial condensed video whenever the size of the queue exceeds K. Then, partial videos are merged into the complete synopsis video. 39

4-8 Simple solution for the discontinuity of motion flow problem. When finding optimum starting labels for 2^{nd} iteration, tails of the object tubes rearranged at 1^{st} iteration (patterned region) are considered as obstacles as shown in (b). 40

5-1 Examples of the test sequences. All sequences were captured with three PTZ cameras at Hanyang university, Seoul, Korea. 44
5-2 Result of the experiment conducted by changing λ. 48
5-2 (continued) Result of the experiment conducted by changing λ. 49
5-2 (continued) Result of the experiment conducted by changing λ. 50
5-3 RT of the proposed algorithm measured by changing λ for five sequences. . . 51
5-4 Result of the experiment conducted by changing $\mathcal{M} \times \mathcal{N}$. 53
5-4 (continued) Result of the experiment conducted by changing $\mathcal{M} \times \mathcal{N}$. 54
5-4 (continued) Result of the experiment conducted by changing $\mathcal{M} \times \mathcal{N}$. 56
5-5 RT of the proposed algorithm measured by changing $\mathcal{M} \times \mathcal{N}$ for five sequences. 57
5-6 Result of the experiment conducted by changing K. 59
5-6 (continued) Result of the experiment conducted by changing K. 60
5-6 (continued) Result of the experiment conducted by changing K. 61
5-7 RT of the proposed algorithm measured by changing K for five sequences. . . 62
5-8 Result of the ablation study to compare four different versions of the proposed algorithm. 63
5-9 Result of the experiment regarding RT measured in seconds. 66
5-10 Result of the experiment regarding FR. 67
5-11 Result of the experiment regarding CR. 67
5-12 Result of the experiment regarding OR. 68

LIST OF TABLES

4-1 Performance comparison of different background estimation methods regarding F-measure on LASIESTA dataset [5]. This table is from the work of Patil *et al.* [6]. Among them, MSFgNet [6] is one and only deep learning based approach and it outperforms other methods with a large margin. . . . 33

5-1 List of test sequences used in the experiments. 45

5-2 List of sequences used to find optimum parameters of the proposed tube rearrangement algorithm. 46

5-3 RT of the proposed algorithm measured in seconds by changing $\mathcal{M} \times \mathcal{N}$ for five video clips. 55

5-4 Four versions of the proposed algorithm used for ablation study. 63

ABSTRACT

Video synopsis allows us to analyze security videos efficiently by condensing or shortening a long video into a short one. To generate a condensed video, moving objects (a.k.a. object tubes) in the video are rearranged in the temporal domain using a predefined objective function. The objective function consists of several energy terms which play important roles in making a visually appealing condensed video. One of the energy terms, collision energy, creates a bottleneck in the computation because it requires two object tubes to calculate the degree of collision between them. Existing approaches try to reduce the computation time of the collision energy calculation by reducing the number of tubes processed at once. However, those approaches are not sufficient to generate condensed video when the number of object tubes becomes large.

In this letter, we propose a fast Fourier transform (FFT)-based parallelized tube rearrangement algorithm. To take advantage of both parallel processing and FFT, we represent object tubes as 3D binary matrices (occupation matrices). An objective function of the tube rearrangement problem is defined on the occupation matrix, and a starting position for each tube in the temporal domain is then determined by optimizing the objective function. Throughout the experiments, the proposed algorithm took a much shorter time to condense

the video than existing algorithms, while other performance metrics were similar.

1 Introduction

1.1 Motivation

The field of security video summarization has been studied for decades to reduce burdens of browsing large amount of video footages. Earlier approaches [7–9] prior to video synopsis [10–12] suffered from several disadvantages including low frame condensation ratio (FR) or missing information, when the frame length of the input video was long. Fundamental building blocks of such approaches were image frames, which means that they tried to select a subset of image frames representing the original video best. On the other hand, building blocks of video synopsis [10–12] are moving objects extracted from the scene, called *object tubes*. In the video synopsis framework, the object tubes are rearranged in the temporal domain and stitched back with background images to generate a short and condensed video. This difference allows video synopsis to efficiently utilize the spatial domain of the video and to drastically improve the FR as compared to the earlier approaches.

Among the diverse research topics in video synopsis, solving the optimization problem for determining starting positions (starting labels) of the object tubes in the temporal domain greatly affects the system performance regarding computation time. This problem is simply

denoted as *a tube rearrangement problem*.

In the pioneering work of video synopsis by Pritch *et al.* [12], the tube rearrangement problem is formulated as Markov Random Fields (MRFs) [13] with four energy terms: activity, collision, temporal consistency, and background consistency. The starting label for each object tube is then determined by minimizing the energy function of MRFs with a simulated annealing [14] or greedy optimization algorithm [15]. During the optimization process, calculating pairwise energy terms (in this case, collision and temporal consistency) becomes a bottleneck for computation speed, because such calculation has $O(TK^2)$ complexity, where T is the number of time steps and K is the total number of object tubes.

In order to cope with the problem, Pritch *et al.* [1] suggest a clustering based optimization algorithm. It divides object tubes into several subsets; then, the optimization algorithm is conducted on each subset. Since the number of object tubes belonging to each subset is much smaller than K, execution time of the optimization algorithm is greatly reduced. However, its condensation result depends on the performance of the clustering algorithm which has a chance to generate inappropriate clusters.

An alternative approach to tube rearrangement is an online video synopsis [16–20], which solves a stepwise optimization problem. In stepwise optimization, instead of considering entire object tubes at the same time, the starting labels of the object tubes are determined one by one. Therefore, it requires less computational power and memory space than batch or offline video synopsis. In addition, since online video synopsis optimizes the object tubes in chronological order, it is inessential to consider temporal and background consistencies. Therefore, the most of online video synopsis frameworks mainly consider the collision energy during the optimization.

Based on such advantages, recent studies of online video synopsis focused on finding efficient ways of solving a stepwise optimization problem: for example, the maximum a posteriori estimation [16], a Tetris-like tube rearrangement strategy [17, 18], and a potential collision graph [19]. Even though these existing algorithms have their own virtue, they have lack of considerations of multi-core environment, which means that there are still rooms for improvements.

In this dissertation, the tube rearrangement problem is reformulated as a suitable form for parallel processing, and the novel concurrent optimization algorithm based on 1D convolutions is proposed to accelerate the optimization process. As in other online tube rearrangement algorithms [16–20], the collision energy is primarily considered during the tube rearrangement. As a preprocessing step, the proposed algorithm reshapes object tubes into probabilistic occupation matrices of $\mathcal{M} \times \mathcal{N} \times \mathcal{T}$ dimension, where $\mathcal{M}$ and $\mathcal{N}$ represent the spatial domain, and $\mathcal{T}$ represents the time domain. This occupation matrix becomes a fundamental building block of the proposed algorithm. Then, the collision energy between two object tubes can be computed as element-wise multiplications of two occupation matrices. This process can be accelerated by utilizing Fast Fourier transform (FFT) [21] in conjunction with parallel computing; therefore, the proposed algorithm can effectively determine the starting labels of numerous object tubes in very short amount of time.

1.2 Related works

A summary of the recent advances in video synopsis is presented as follows. Nie *et al.* [22] rearrange object tubes in both temporal and spatial domain to generate more condensed

videos. Zhu *et al.* [23] and Mahapatra *et al.* [24] extend the concept of video synopsis to the multi-camera network. Wang *et al.* [25] and Zhong *et al.* [26] utilize the compressed domain to generate synopsis videos efficiently. X. Li *et al.* [27] scale down object sizes to reduce collisions in the synopsis video. Z. Li *et al.* [28] and K. Li *et al.* [29] introduce a seam carving method to remove redundant information from the original video.

1.3 Dissertation overview

The rest of the dissertation is organized as follows. Chapter 2 introduces the problem formulation of video synopsis and details of the proposed tube rearrangement algorithm are described in Chapter 3. Chapter 4 contains explanations of other components that the online video synopsis consists of. Chapter 5 presents experimental results, and the dissertation is concluded in Chapter 6.

2 Problem formulation of video synopsis

In this chapter, the problem formulation of video synopsis introduced in the pioneering works [10–12] is described to show which part of the formulation has to be changed to apply parallel processing. In addition, the reason why online video synopsis mainly considers collision energy is explained in detail.

As in Figures 2-1 and 2-2, a principal objective of video synopsis is shortening length of the input video by relocating object tubes in temporal domain. In other words, we try to find the best combination of object tubes' starting positions in temporal domain (starting labels). In the field of video surveillance, a definition of the best combination can be different from specific applications. However, based on the paper of Pritch *et al.* [12], the condensed video with the best starting label combination should have following characteristics.

- Objects of interests should be appeared in the condensed video.
- Rearranged object tubes should seamlessly rendered in the condensed video.
- The condensed video has significantly shorter length than the input video.
- Dynamics of objects or interactions between the objects should be understood in the condensed video.

To achieve the characteristics, the batch video synopsis [12] utilizes four energy terms as described in Chapter 1: activity, background consistency, collision, and temporal consistency. The order of the energy terms are matched with that of the characteristics.

Assume that $L = \{l_0, ..., l_N\}$ is a set of starting labels for N object tubes; then, an objective function $E(L)$ can be defined as

$$E(L) = \sum_{l_i \in L} \left(E_a(l_i) + \gamma E_s(l_i) \right) + \sum_{l_i, l_j \in L} \left(\alpha E_t(l_i, l_j) + \beta E_c(l_i, l_j) \right), \tag{2.1}$$

where E_a, E_s, E_t, and E_c are activity, background consistency, temporal consistency, and collision energies, respectively. In addition, α, β, and γ are weighting parameters for controlling importance between the energies.

2.1 Activity energy

At first, E_a defines which object tubes should be appeared in the condensed video. One example of E_a is

$$E_a(l_i) = \begin{cases} \sum_{x,y,t} \chi_i(x, y, t) & l_i \in L_e \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where l_i and $\chi_i(x, y, t)$ are the starting label and the characteristic function of the i^{th} object tube, respectively. Due to the condition ($l_i \in L_e$) in (2.2), the only characteristic function of the object tube whose starting label belongs to L_e is added to E_a. The set L_e contains starting labels of the objects not included in the condensed video. Therefore, the role of

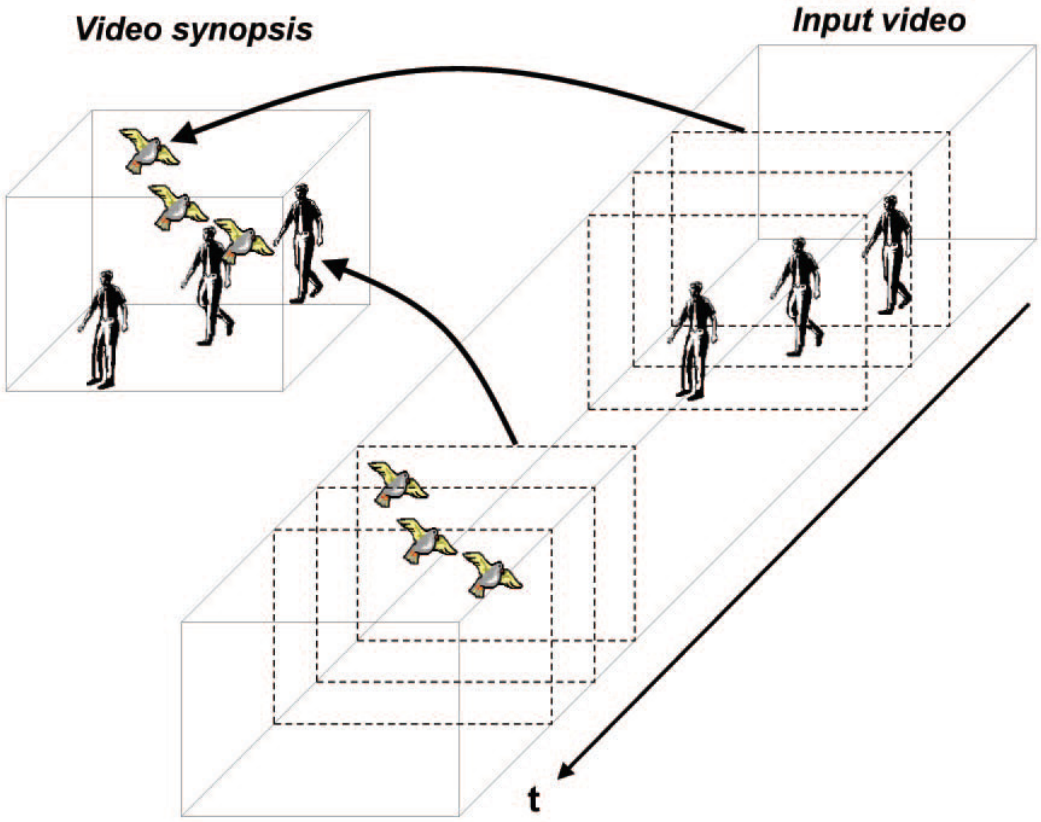


Figure 2-1. Concept diagram of video synopsis [1]. The bird and man appeared at different time in the original video are rearranged in temporal domain, and then displayed simultaneously in the condensed video.

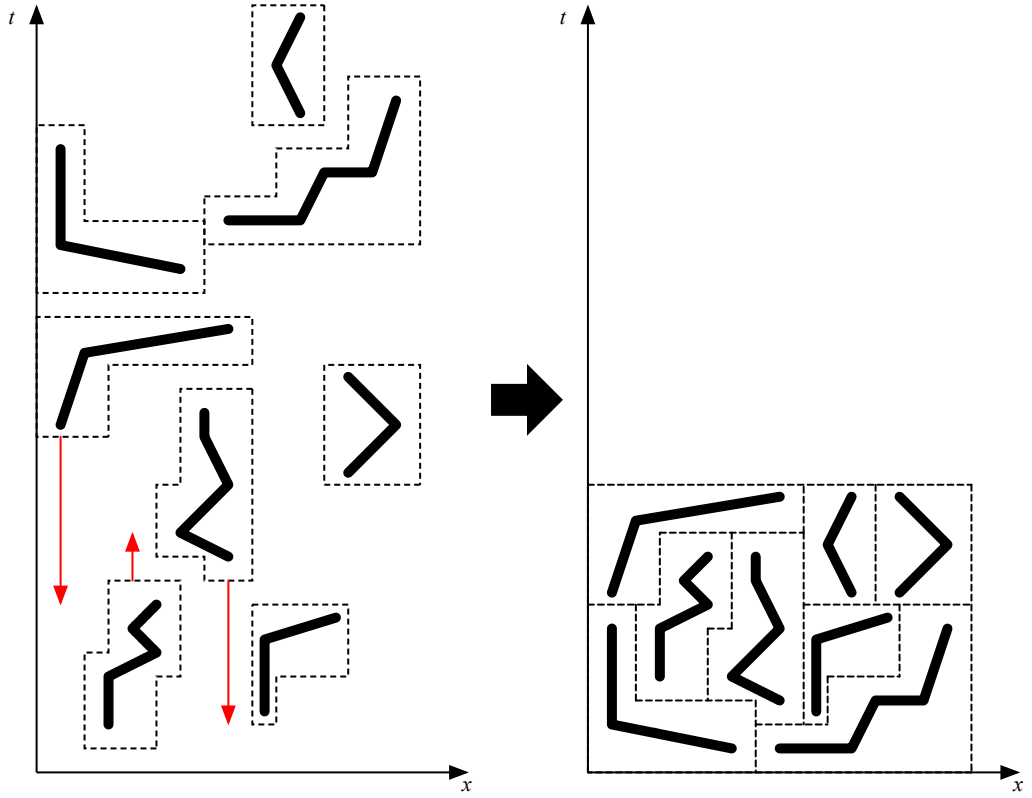


Figure 2-2. Example of the object tube rearrangement in 2D space. Red arrows indicate some offsets of the starting labels for better understanding of the tube rearrangement process. We can see that after the tube rearrangement, the length of the condensed video becomes much shorter than that of the original.

E_a is penalizing exclusions of the object tubes. On the other hand, $\chi(x, y, t)$ represents the importance of the object tube. If the characteristic function of one object has larger values than that of the others, the object is more likely to be included in the resulting video. In the original work of video synopsis [10–12], $\chi_i(x, y, t)$ is defined as

$$\chi_i(x, y, t) = \begin{cases} \left\| I_i(x, y, t) - B(x, y, t) \right\| & t \in t_i \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

where $I_i(x, y, t)$ is a foreground pixel of i^{th} object and $B(x, y, t)$ is a respective background pixel, and t_i is a period of time in frames indicating the appearance of the object. Based on (2.3), the condensed video prefers the object tubes having distinctive colors as compared with the background.

Defining a proper E_a is important for processing the query of the video synopsis users, since it determines which objects will be included in the resulting video. However, we do not have to directly optimize E_a because object filtering step prior to the optimization with specific conditions (e.g., colors, trajectories, object types, and etc) can do the same functionality.

2.2 Time-lapse background generation

Before moving on to the next energy term, how to generate time-lapse background is briefly explained. Since the main objective of video synopsis is condensing the contents of the original video, background information as well as foreground has be condensed too. If

the input video is 12 hours long and the condensed video is 10 minutes long, time-lapse background can be generated by uniformly subsampling every 720th of original background images or we can use the adaptive sampling rate proportional to (or inverse proportional to) the number of objects in the current frame [12]. An example of the time-lapse background generation is illustrated in **??**.

2.3 Background consistency energy

The role of the second energy term in (2.1), E_s, is to seamlessly render the object tubes with the time-lapse background images. In the video synopsis framework, foreground pixels of the object tubes are stitched with the background images to generate the condensed video. During the stitching process, image blending algorithms (e.g., Poisson image editing [4]) can be used to smoothly blend the foreground and background pixels. However, inaccurate segmentation results of the foreground or foreground and background pixels from different time of day can cause visually unappealing results as shown in **??**. E_s is defined to penalize such situation.

$$E_s(l_i) = \sum_{x,y \in \sigma_i, t} \left\| I_i(x,y,t) - B_t(x,y,t) \right\|, \tag{2.4}$$

where σ_i is a set of boundary pixels for the i^{th} object and $B_t(x,y,t)$ is a pixel of the time-lapse background. To obtain σ_i, we can apply morphological dilation to the foreground mask of the i^{th} object and subtract it from the original. Based on (2.4), the object appeared in the midnight are more likely to be appeared at night-part of the time-lapse background.

In the online video synopsis framework, object tube extraction, time-lapse background

generation, and foreground-background stitching are conducted in real-time; therefore, foreground and background pixels are from the similar time of day. Therefore, online video synopsis has less reason to consider E_s during the optimization.

2.4 Temporal consistency energy

The temporal consistency energy, E_t, is designed to keep chronological orders between the object tubes in the original video. If the condensed video contains chronological disorders between the tubes, we may miss the important interaction between the objects presented in the original video. Prior to further discussion about E_t, we need to define a probability of the interaction between the two object tubes first. If the objects share common time periods in the original video ($t_i \cap t_j \neq \emptyset$), the probability becomes

$$p_I(i,j) = \exp\left(-\min_{t \in t_i \cap t_j} \frac{d(i,j,t)}{\sigma_s}\right), \tag{2.5}$$

where $d(i,j,t)$ is a Euclidean distance between the closest pixels of i^{th} and j^{th} objects in frame t and σ_s is a parameter for adjusting a spatial range of the interaction. Based on (2.5), a pair of the objects spatially adjacent to each other is more likely to have interactions between them.

On the other hand, if the objects do not have any overlap in the temporal domain of the original video, $p_I(i,j)$ is defined as

$$p_I(i,j) = \exp\left(-\frac{l_j - (l_i + T_i)}{\sigma_t}\right), \tag{2.6}$$

where T_i is the number of frames in the i^{th} object tube and σ_t determines a temporal proximity between the objects. In addition, (2.6) is defined on the assumption that the i^{th} object appears earlier than the j^{th} object in the input video ($l_i + T_i < l_j$). Therefore, the object tubes located far from each other in temporal domain are less likely to have interactions.

In summary, (2.5) and (2.6) encode the idea that objects close in spatio-temporal domain have strong interactions. Based on the two equations, we can define E_t to keep chronological orders between the objects when generating the condensed video.

$$E_t(i,j) = p_I(i,j) \cdot \begin{cases} 0 & \hat{l}_i - \hat{l}_j = l_i - l_j \\ C & \text{otherwise,} \end{cases} \tag{2.7}$$

where $\hat{l}$ indicates a starting label of the object in the input video and C is a large constant value to penalize the objects having temporal inconsistencies.

Since the behavior of the equation (2.7) is not straightforward, detail explanations will be given through examples. Assume that two objects are close in spatio-temporal domain of the original video. In this case, E_t of two objects becomes very large (due to C), when their relative starting label in the condensed video ($l_i - l_j$) is not exactly same as in the input video ($\hat{l}_i - \hat{l}_j$). Conversely, the objects far from each other in spatio-temporal domain have a low penalty for violating the condition ($\hat{l}_i - \hat{l}_j = l_i - l_j$), because their p_I has a small value.

As similar to E_s, the role of E_t is not significant in online video synopsis. Recent online video synopsis frameworks [19, 20] maintain a queue of object tubes and the queue grows as

new object tube is extracted in the input video. When the size of the queue exceeds a certain threshold K, framework generates a partial condensed video with K object tubes, and then removes the first K objects from the queue. Based on the framework, chronological disorders only can be presented when the objects are in the same part of the condensed video. Even if the objects are optimized together to generate a same part of the resulting synopsis video, their temporal inconsistencies are negligible, because their relative spatio-temporal distance is small. In consequence, the one and only energy term to optimize in online video synopsis is the collision energy.

2.5 Collision energy

The key role of E_c is to prevent the resulting synopsis video from becoming crowded. During the video synopsis process, the objects from different time periods in the input video are displayed simultaneously in the same scene of the condensed video. In this case, pixel overlaps between the objects make us difficult to understand the context of the synopsis video. To penalize such situation through E_c, a degree of collision between the objects is defined as

$$E_c(l_i, l_j) = \sum_{x,y,t \in t_i \cap t_j} \chi_i(x, y, t)\chi_j(x, y, t). \tag{2.8}$$

Based on (2.8), a collision between two objects having distinctive colors from the background is considered more seriously. However, this definition of E_c is computationally expensive due to $\chi(x, y, t)$. Therefore, in this dissertation, the multiplication of two characteristic functions is replaced with the intersection over union (IoU) between two bounding boxes of

the objects.

$$E_c(l_i, l_j) = \sum_{x,y,t \in t_i \cap t_j} IoU\left(B_i(t), B_j(t)\right), \tag{2.9}$$

$$IoU(B_i, B_j) = \frac{B_i \cap B_j}{B_i \cup B_j}, \tag{2.10}$$

where $B_i(t)$ and $B_j(t)$ are bounding boxes of i^{th} and j^{th} objects at frame t, respectively. Since the bounding box does not represent an exact location of the object, (2.9) can be thought as an approximated version of (2.8).

2.6 Computational bottleneck

In (2.1), we should note that energies can be categorized into two groups regarding the number of required parameters: unary and pairwise. Activity and background consistency only requires a single object tube to calculate the energies; on the other hand, remaining energies require two object tubes for the calculation. When the number of objects to optimize increases, pairwise energy terms become a bottleneck of the computation. Since E_s is not the main concern of online video synopsis, E_c becomes the one and only issue for the computational burden. As described in Section 1.2, recent studies of video synopsis [19] utilize different calculations of E_c, but their definitions of the collision energy are not suitable for parallel processing. In the following section, a new representation of the object tube named as an occupation matrix which has a suitable form for concurrent computation of E_c will be introduced.

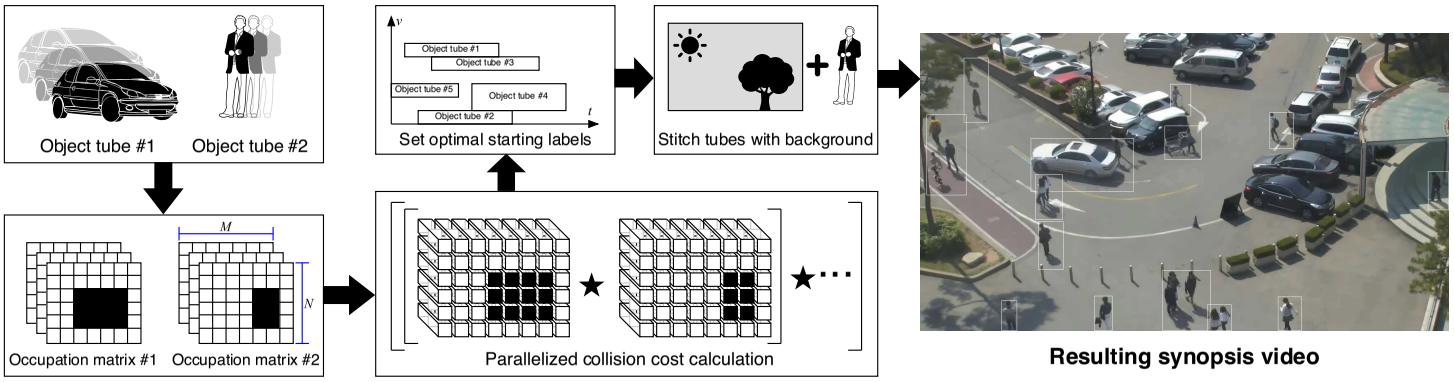


Figure 2-3. Flowchart of the proposed online video synopsis framework. At the beginning, foreground of the object tube is reshaped into the 3D occupation matrix. We use this matrix representation to calculate the collision energy fast in conjunction with parallel processing and Fourier transform. Afterwards, we determine optimal starting labels for tubes and stitch tubes with the background to generate a resulting synopsis video. Illustrations of man and vehicle in this figure are created by Lluisa Iborra and Yasser Megahed from the Noun Project.

3 Proposed tube rearrangement

In this chapter, E_c is reformulated using the occupation matrix and an efficient tube rearrangement algorithm for optimizing the objective function is proposed. In addition, two types of the occupation matrix (binary and probabilistic) are introduced and their characteristics are explained in detail. A flowchart of the proposed online video synopsis framework including the tube rearrangement algorithm is illustrated in Figure 2-3.

3.1 Occupation matrix generation

Each element of the occupation matrix $\mathbf{M}_i(u, v, t)$ is either from Boolean or continuous domain, and represents the probability of existence for i^{th} object tube at position (u, v) and time t of a video whose spatial resolution is $\mathcal{H} \times \mathcal{W}$. The i^{th} occupation matrix $\mathbf{M}_i$ is then formed by stacking resized binary foreground masks of the object over multiple frames. The resized foreground mask has $\mathcal{M} \times \mathcal{N}$ resolution, where $\mathcal{M}$ and $\mathcal{N}$ have much smaller values than the width and height of the original video ($\mathcal{M} \ll \mathcal{H}$ and $\mathcal{N} \ll \mathcal{W}$). In this dissertation, two strategies of resizing will be introduced in following subsections and they determine the type of resulting occupation matrix: binary and probabilistic.

3.1.1 Binary occupation matrix

The binary occupation matrix $\mathbf{M}^b$ does not allow gray area value to represent the existence of objects; it can only have 1s and 0s. Assume that the foreground mask of the i^{th} object is denoted as $\mathbf{F}_i(x, y, t) \in \mathbb{B}$; then, $\mathbf{M}_i^b(u, v, t)$ is defined as

$$\mathbf{M}_i^b(u, v, t) = \begin{cases} 1 & \sum_{(x,y)\in C(u,v)} \mathbf{F}_i(x, y, t) \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

where $C(u, v)$ is a set of 2D coordinates (x, y). Based on (3.1), to calculate a single element of $\mathbf{M}_i^b$, we need to examine the values of $\mathbf{F}_i$ for every coordinate in $C(u, v)$. The definition of $C(u, v)$ is given by

$$C(u, v) = \left\{(x, y) \mid x \in X(u), y \in Y(v)\right\}, \tag{3.2}$$

where $X(u)$ and $Y(v)$ are sets of x and y coordinates, respectively.

$$X(u) = \left\{ x \;\middle|\; \left\lfloor \frac{\mathcal{W}}{\mathcal{N}} u \right\rceil \leq x < \left\lfloor \frac{\mathcal{W}}{\mathcal{N}} (u + 1) \right\rceil \right\}, \tag{3.3}$$

$$Y(v) = \left\{ y \;\middle|\; \left\lfloor \frac{\mathcal{H}}{\mathcal{M}} v \right\rceil \leq y < \left\lfloor \frac{\mathcal{H}}{\mathcal{M}} (v + 1) \right\rceil \right\}. \tag{3.4}$$

Due to the condition $\left(\sum_{(x,y)\in C(u,v)} \mathbf{F}_i(x, y, t) \neq 0\right)$ in (3.1), even a single pixel of $\mathbf{F}_i(x, y, t)$ can produce a response in $\mathbf{M}_i^b(u, v, t)$. Therefore, $\mathbf{M}_i^b$ exaggerates the occupation region of the object tube in the video sequence. An example of the binary occupation matrix gener-

ation is depicted in Figure 3-1.

3.1.2 Probabilistic occupation matrix

Since the probabilistic occupation matrix $\mathbf{M}_i^p$ represents the existence of the object tube with continuous values, it can provide more precise information than $\mathbf{M}_i^b$. Each element of $\mathbf{M}_i^p$ is calculated as

$$\mathbf{M}_i^p(u,v,t) = \frac{\sum_{(x,y)\in C(u,v)} \mathbf{F}_i(x,y,t)}{|C(u,v)|}. \tag{3.5}$$

where $|C(u,v)|$ is a cardinality of $C(u,v)$. In most cases, where $\mathcal{W}/\mathcal{N} \in \mathbb{N}$ and $\mathcal{H}/\mathcal{M} \in \mathbb{N}$, $|C(u,v)|$ becomes a constant value.

3.2 Objective function

For the next step, the collision energy is reformulated with the occupation matrix and a new energy term E_l is introduced to penalize a long condensed video. Then, the final objective function of the proposed tube rearrangement algorithm is defined by considering both E_c and E_l.

3.2.1 Reformulated collision energy

The motivation behind the reformulation of E_c is that the degree of collision between the objects at a certain frame can be calculated as a sum of the element-wise multiplication of

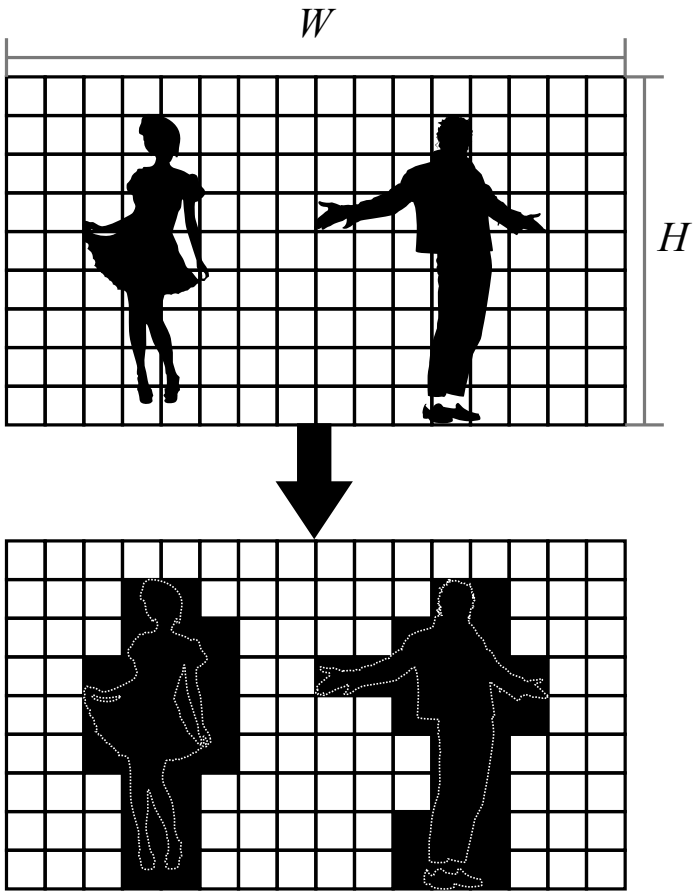


Figure 3-1. Example of the binary occupation matrix generation when $\mathcal{W}/\mathcal{N} = 16$ and $\mathcal{H}/\mathcal{M} = 9$. The foreground and background of the object are represented in black and white, respectively. Dotted lines in the figure are depicted to show contours of the original object for the readers. Illustrations of the woman and man in this figure are created by Nataliia Lytvyn and Ludovic Gicqueau from the Noun Project, respectively.

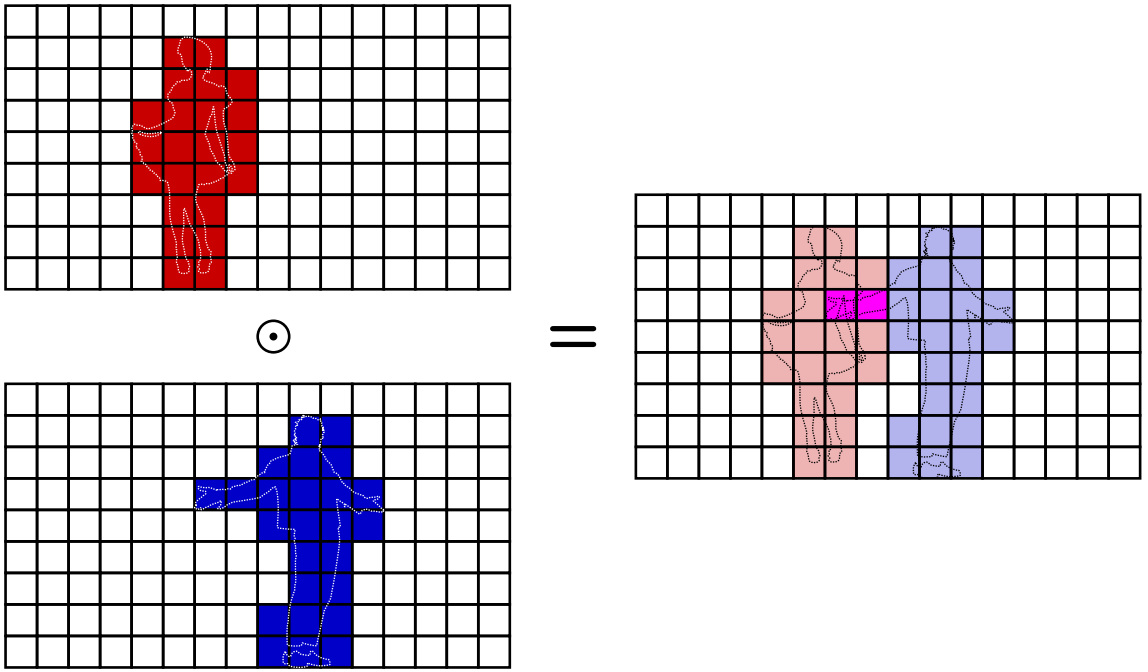


Figure 3-2. Example calculation of reformulated collision energy E_c with two binary occupation matrices. Occupied elements in the matrix are colored in red and blue. After the element-wise multiplication, we can see that the objects have two collided elements colored in magenta. In this figure, $\odot$ is an operator for the element-wise multiplication, also known as Hadamard product.

two occupation matrices. An example of this computation is depicted in Figure 3-2 and the redefined $E_c(l_i, l_j)$ is given by

$$E_c(l_i, l_j) = \sum_{u=1}^{\mathcal{M}} \sum_{v=1}^{\mathcal{N}} \sum_{t=t_{\min}}^{t_{\max}} \mathbf{M}_i(u, v, t)\mathbf{M}_j(u, v, t), \tag{3.6}$$

where $t_{\min}$ and $t_{\max}$ are minimum and maximum values of the overlapped temporal domain. Detailed calculations of $t_{\min}$ and $t_{\max}$ are

$$t_{\min} = \max(l_i, l_j), \tag{3.7}$$

$$t_{\max} = \min(T_i + l_i, T_j + l_j), \tag{3.8}$$

where T_i and T_j are frame lengths of the i^{th} and j^{th} object tubes, respectively.

3.2.2 Length energy

Apart from the existing video synopsis frameworks using the fixed length of the synopsis video [], the proposed framework adaptively adjusts the length of the condensed video by considering both compactness and complexity. In this regard, the length energy $E_l(l_i, l_j)$ is defined as the frame length of the synopsis video when two object tubes have starting labels of l_i and l_j.

$$E_l(l_i, l_j) = \max(T_i + l_i, T_j + l_j) - \min(l_i, l_j). \tag{3.9}$$

An objective function $E(l_i, l_j)$ is calculated as a weighted sum of the collision and length energies.

$$E(l_i, l_j) = E_c(l_i, l_j) + \lambda E_L(l_i, l_j), \tag{3.10}$$

where λ is a weighting parameter adjusting the importance of the length energy. In general, the larger λ generates the shorter but more complex synopsis video; on the other hand, the smaller λ produces the longer but less confused condensed video.

3.3 Optimizing objective function

As in other online video synopsis algorithms [18–20], the proposed tube rearrangement algorithm adopts the stepwise optimization strategy; therefore, starting labels of the object tubes are determined one by one through iterations. At the i^{th} iteration of the optimization, the starting label of i^{th} object tube l_i is determined as

$$l_i = \arg\min_{l} E(l, L_{i-1}) \text{ subject to } l_i \geq 0, \tag{3.11}$$

where $L_{i-1} = \{l_1, ..., l_{i-1}\}$ is a set of starting labels determined after $i-1$ iterations. A constraint to the optimization $l_i \geq 0$ is used to alleviate chronological disorder in the synopsis video. In other words, since a negative l_i means that i^{th} tube appear prior to the first tube in the synopsis video, preventing such case increases a chance to keep chronological order of the tubes.

Due to the stepwise optimization strategy, one of two input arguments for E in (3.11)

becomes L_{i-1} instead of a single label as described in (3.10). In consequence, slight modifications of (3.6) and (3.9) are necessary. For the stepwise optimization, the calculation of E_c is modified as

$$E_c(l_i, L_{i-1}) = \sum_{u=1}^{\mathcal{M}} \sum_{v=1}^{\mathcal{N}} \sum_{t=t^*_{\min}}^{t^*_{\max}} \mathbf{M}_i(u, v, t)\mathbf{M}^*_{i-1}(u, v, t), \tag{3.12}$$

where $\mathbf{M}^*_{i-1}$ is an accumulated occupation matrix for $i-1$ iterations, and $t^*_{\min}$ and $t^*_{\max}$ are minimum and maximum bounds of shared temporal domain between $\mathbf{M}_i$ and $\mathbf{M}^*_{i-1}$. Moreover, each element of $\mathbf{M}^*_{i-1}$ is defined in the recurrence relation as

$$\mathbf{M}^*_{i-1}(u, v, t) = \mathbf{M}_{i-1}(u, v, t - l_{i-1}) + \mathbf{M}^*_{i-2}(u, v, l^*_{i-2}), \tag{3.13}$$

where $l^*_{i-2} = \min L_{i-2}$. For the initial condition of (3.13), $\mathbf{M}^*_1 = \mathbf{M}_1$ and $l^*_1 = l_1 = 0$ are used. Formal definitions of $t^*_{\min}$ and $t^*_{\max}$ are

$$t^*_{\min} = \max(l_i, l^*_{i-1}) \tag{3.14}$$

and

$$t^*_{\max} = \min(T_i + l_i, T^*_{i-1} + l^*_{i-1}), \tag{3.15}$$

where T^*_{i-1} is a frame length of $\mathbf{M}^*_{i-1}$. The length energy for the stepwise optimization is defined as

$$E_l(l_i, L_{i-1}) = \max(T_i + l_i, T^*_{i-1} + l^*_{i-1}) - \min(l_i, l^*_{i-1}). \tag{3.16}$$

3.3.1 Properties of accumulated occupation matrix

For better understanding of the stepwise optimization process, we will discuss about properties of the accumulated occupation matrix $\mathbf{M}^*$. According to the type, the occupation matrix $\mathbf{M}$ can have either Boolean or continuous values in the range from 0 to 1. On the other hand, $\mathbf{M}^*$ is computed by adding two matrices as described in (3.13); therefore, each element of $\mathbf{M}^*$ belongs to either $\mathbb{N}_0$ or $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\}$. By utilizing $\mathbf{M}^*$, we can represent occupation and collision states of more than two objects on the single matrix.

3.4 Parallelized optimization

Even though $\mathbf{M}$ provides an efficient way of representing the object tubes and E_c can be computed easily with the element-wise multiplication, optimization of E_c can further be accelerated by using both parallel processing and cross-correlation of two occupation matrices in the temporal domain. Prior to define the cross-correlation, assume that two occupation matrices overlap by at least one frame in the temporal domain. Without this restriction, E needs to be evaluated for every possible l value. Then, the parallelized version of E_c in (3.12) is defined as

$$\begin{aligned} E_c(l_i, L_{i-1}) &= \sum_{u=1}^{\mathcal{M}} \sum_{v=1}^{\mathcal{N}} \mathbf{M}_i \star \mathbf{M}^*_{i-1}(u, v, l_i - l^*_{i-1}) \\ &= \sum_{u=1}^{\mathcal{M}} \sum_{v=1}^{\mathcal{N}} \sum_{t=-\infty}^{\infty} \mathbf{M}_i \mathbf{M}^*_{i-1}(u, v, t + l_i - l^*_{i-1}), \end{aligned} \tag{3.17}$$

where $\star$ is an operator for the cross-correlation.

Algorithm 1 Proposed tube rearrangement algorithm

Input: $\mathbf{M}_i, i = 1, ..., N$

Output: $L_N = \{l_1, ..., l_N\}$

$\mathbf{M}_1^* = \mathbf{M}_1$, $l_1^* = l_1 = 0$, $L_1 = \{l_1\}$

for $i = 2$ to N **do**

Calculate $\mathbf{M}_i \star \mathbf{M}_{i-1}^*$ using FFT and parallel processing

Find a local optimum starting label l_i by using (3.11)

Calculate $\mathbf{M}_i^*$ from $\mathbf{M}_i$ and $\mathbf{M}_{i-1}^*$ by using (3.13)

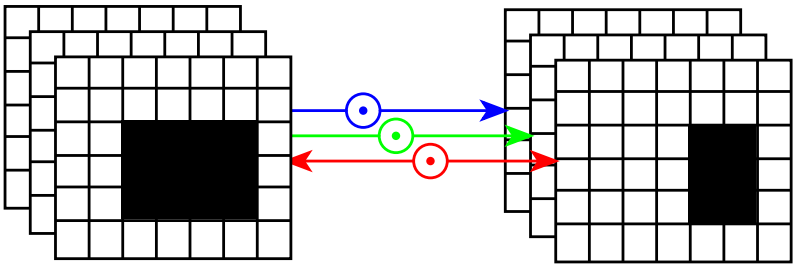
$L_i = L_{i-1} \cup l_i$

$l_i^* = \min L_i$

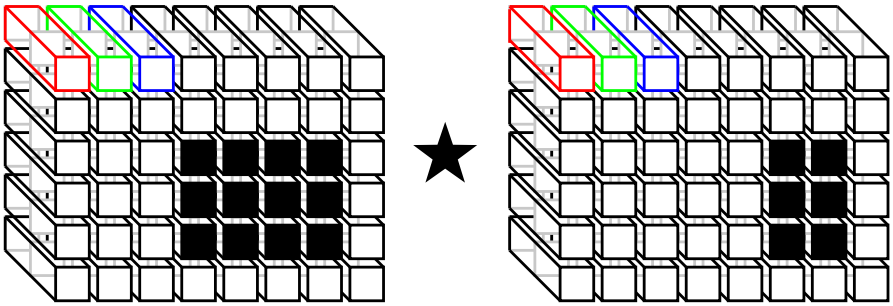
end for

return L_N

The motivation behind the conversion from (3.12) to (3.17) is illustrated in Figure 3-3. From (3.12), if we take the spatial coordinate into the consideration first, the 3D element-wise multiplication can be thought as a series of 2D Hadamard products in temporal domain as shown in Figure 3-3a. On the other hand, if we consider the temporal domain first, the operation becomes $\mathcal{M} \times \mathcal{N}$ 1D cross-correlations as illustrated in Figure 3-3b. This difference may seem to be minor but it is important when we consider some tricks to accelerate the operation. The computational burden of multiple 1D cross correlations can be reduced by Fast Fourier Transform (FFT) [21] in conjunction with parallel processing. A detailed procedure of the proposed tube rearrangement algorithm is presented in Algorithm 1.



(a) 2D Hadamard products



(b) 1D cross-correlations

Figure 3-3. Two ways of calculating E_c. All of occupation matrices in this figure have $6 \times 8 \times 3$ spatio-temporal resolution. E_c can be calculated by using (a) Hadamard products between two sets of frames, and (b) 1D cross correlations between 48 pairs of 1D signals. Three primitive colors (red, green, and blue) in this figure is used to show example correspondences.

4 Online video synopsis framework

The proposed tube rearrangement algorithm is based on the online framework. Similar to existing online frameworks [18–20], the proposed framework consists of four stages: background modeling, object tube generation, tube rearrangement, and object stitching. Among them, three components, except for the tube rearrangement, will be explained in detail.

4.1 Background modeling

Since this field of research has been studied for decades, there are numerous choices for modeling the background. According to the recent review literature [30], apart from well-established statistical background modelings [31–38] and neural network based approaches [39, 40], deep learning based approaches become a main stream of the research and they can be categorized into two broad groups: convolutional neural networks (CNN) [2, 6] and generative adversarial network (GAN) [3, 41, 42].

Figure 4-1 shows a process of segmenting the foreground from the background introduced by Lim *et al.* [2]. This process has a lot in common with image segmentation using CNN architecture [43–48]; encoder module is for extracting feature maps and decoder module is

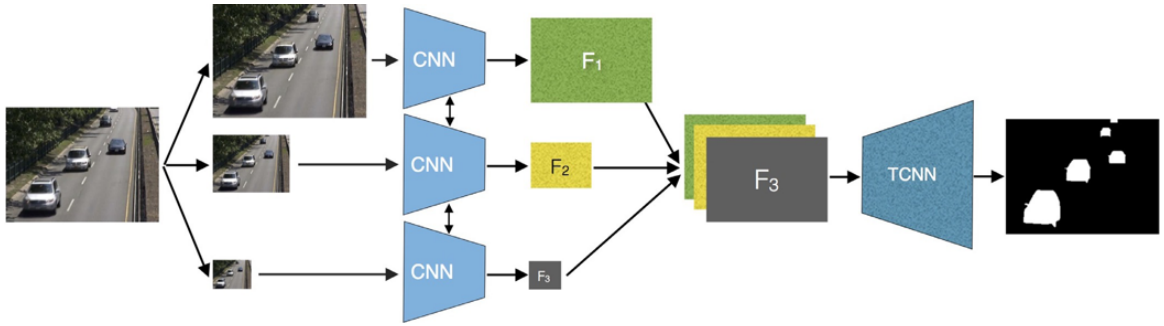


Figure 4-1. Flowchart of foreground segmentation by Lim *et al.* [2]. It follows the common process of the image segmentation, but its performance has been increased by incorporating feature maps from multiple scales.

for compensating reduced spatial resolution. The key difference between the foreground and image segmentation is the number of output labels; the former produces only two labels while the latter discerns more than 20 labels for renowned PASCAL VOC 2012 dataset [49] and 30 labels for Cityscapes dataset targeted for the autonomous driving application [50].

GAN is one of the most actively researched topics in the computer vision. Due to its intriguing idea, many researchers in different field of studies try to solve their problems by using GAN and the problem of modeling background is one of them [3, 41, 42]. Figure 4-2 shows a framework of segmenting the foreground with GAN introduced by Sakkos *et al.* [3]. They focus on solving the varying illumination problem in the background modeling and achieve the goal by using a triple multi-task GAN which jointly optimizes the GAN and segmentation losses.

As shown in Table 4-1 of Patil *et al.* [6], F-measure of the deep learning based approach (MSFgNet) is superior than those of the non deep learning approaches. However, their computational burden is hard to be ignored for the video synopsis application. Since the main objective of video synopsis is to make users browse videos quickly, computation time is one of the most important things to consider. Therefore, in this dissertation, the proposed framework utilizes a well-known Gaussian mixture model [31, 32] to separate the foreground of the objects from the background and this modeling process can be accelerated by using the GPGPU.

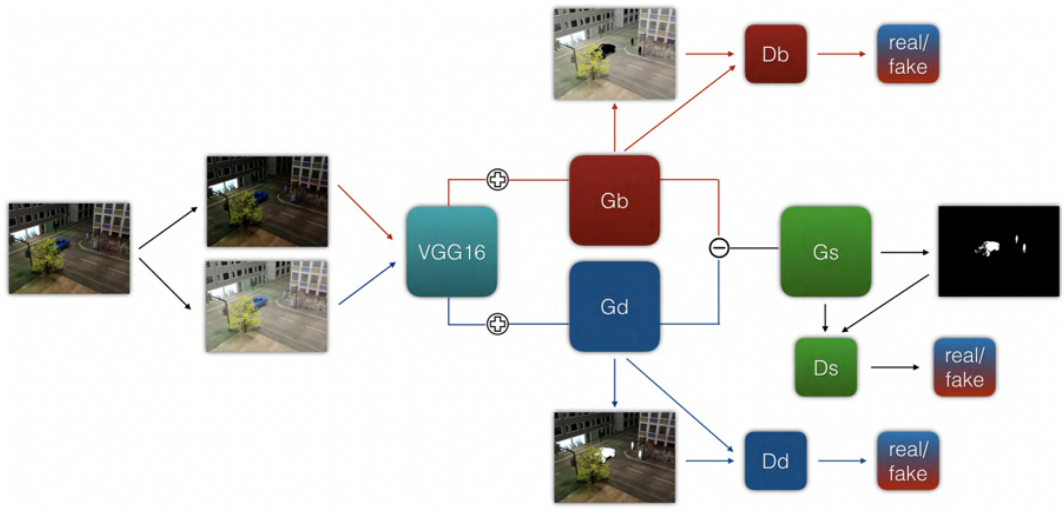


Figure 4-2. Recent framework of foreground segmentation via GAN [3]. It utilizes three generator-discriminator pairs to make foreground segmentation robust against extreme illumination changes. At first, input images undergo the gamma correction to make them in the extreme illumination condition. Red G_b and D_b boxes to generate synthetic bright images from dark images, and to discriminate the synthetic and original ones. On the other hand, blue G_b and D_b boxes do the same task for the bright images. Finally, by using the results of red and blue G_b, G_s and D_s are trained to generate and discriminate segmented foregrounds.

Methods	I_SL	I_CA	I_OC	I_IL	I_MB	I_BS	O_CL	O_RA	O_SN	O_SU	Average
Zivkovic [32]	0.9053	0.8320	0.9507	0.2391	0.8668	0.5308	0.8764	0.8235	0.3804	0.7105	0.7125
Maddalena [39]	0.8696	0.8463	0.9134	0.6142	0.7617	0.4244	0.8766	0.8412	0.5781	0.8015	0.7525
Maddalena [40]	0.9484	0.8573	**0.9540**	0.2105	0.9122	0.4017	0.8709	0.8472	0.8105	**0.8795**	0.7692
Cuevas [51]	0.7859	0.7361	0.8527	0.7915	0.7288	0.5836	0.8638	0.8085	0.4555	0.7305	0.7335
Haines [52]	0.8876	0.8938	0.9223	0.8491	0.8441	0.6809	0.8267	0.8592	0.1735	0.8586	0.7791
Berjón [53]	0.8805	0.8444	0.7807	0.6487	0.8873	0.6642	0.8776	0.8165	0.7765	0.7215	0.7914
MSFgNet [6]	**0.9264**	**0.9213**	0.9163	**0.8967**	**0.9143**	**0.7157**	**0.8806**	**0.8659**	**0.8952**	0.7869	**0.8717**

Table 4-1. Performance comparison of different background estimation methods regarding F-measure on LASIESTA dataset [5]. This table is from the work of Patil *et al.* [6]. Among them, MSFgNet [6] is one and only deep learning based approach and it outperforms other methods with a large margin.

4.2 Object tube generation

After the background modeling stage, we can get foreground masks of objects. To generate the object tubes, the masks that belong to the same object must be associated over temporal domain. This association task is identical to the assignment problem. Solving the assignment problem can be seen as finding a matching, where the sum of edge weights is maximized in the bipartite graph. If the one set in the graph contains foreground masks in i^{th} frame, the other set has the binary masks belong to $(i+1)^{\text{th}}$ frame. This problem can be solved by simple yet efficient Hungarian algorithm [54–56]. The original version of the algorithm requires a condition that cardinalities of two sets are equal; in other words, the number of agents and the number of tasks to be assigned are same. We say that the assignment problem with such condition is linear. However, in real world environment, it is common that cardinalities of two sets are not equal; thus, the extended version of Hungarian algorithm [57] is utilized in this dissertation. Moreover, an intersection of HSV color histograms between two foreground regions is used as a similarity function of the bipartite graph [58].

For online video synopsis, generated object tubes are stored and maintained in a queue. When the size of th queue exceeds K, the starting labels of first K object tubes in the queue are determined by the proposed tube rearrangement algorithm. Then, corresponding tubes are removed from the queue and prepared to be stitched.

(a) Source image (b) Destination image (c) Cloning (d) Seamless cloning

Figure 4-3. First example of seamless cloning using Poisson image editing. All images in this figure are from the work of Pérez *et al.* [4].

(a) Sources/destinations (b) Cloning (c) Seamless cloning

Figure 4-4. Second example of seamless cloning using Poisson image edting. Unlike the first example, there are three regions from two source images for one destination image. All images in this figure are from the work of Pérez *et al.* [4].

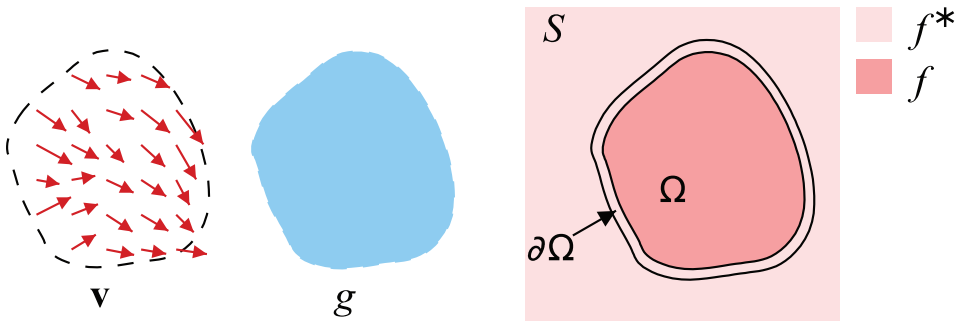


Figure 4-5. Notations used in Poisson image editing [4].

4.3 Object stitching

To make a condensed video, foregrounds of the rearranged object tubes are stitched with the background images by utilizing Poisson image editing [4]. What we can do with Poisson image editing is inserting some part of the source image to the destination image seamlessly as shown in Figure 4-3 and Figure 4-4.

Before explaining the mathematics behind this editing, some notations need to be defined first. In Figure 4-5, $\boldsymbol{S}$ is a spatial domain of the destination image and belongs to $\mathbb{R}^2$, $\boldsymbol{\Omega}$ is the domain to be edited and has a boundary $\partial\boldsymbol{\Omega}$, g and f^* are scalar functions of source and destination images, respectively, f is an unknown function, and $\mathbf{v}$ is a gradient field which will be explained later. In addition, f^* is defined over $S - (\boldsymbol{\Omega} - \partial\boldsymbol{\Omega})$ and f is defined over $\boldsymbol{\Omega}$; therefore, $\partial\boldsymbol{\Omega}$ indicates an overlapped region between $\boldsymbol{S}$ and $\boldsymbol{\Omega}$.

As you can see in Figure 4-5, the objective of the editing is to find a proper f satisfying the boundary condition on $\boldsymbol{\Omega}$. One example of achieving the objective is minimizing the

Figure 4-6. text

following equation.

$$\min_{f} \iint_{\mathbf{\Omega}} |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}. \tag{4.1}$$

The unique solution of (4.1) can be obtained by solving following Poisson equation with Dirichlet boundary condition.

$$\Delta f = \text{div}\mathbf{v} \quad \text{over} \quad \mathbf{\Omega} \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}, \tag{4.2}$$

where div$\cdot$ is a divergence operator; hence $\text{div}\mathbf{v} = \left(\frac{\partial u}{\partial x}, \frac{\partial v}{\partial y}\right)$, when $\mathbf{v} = (u, v)$. In (4.1) and (4.2), $\mathbf{v}$ is used as a guidance field; therefore, how to choose $\mathbf{v}$ can change the purpose of the editing. One possible choice of $\mathbf{v}$ to seamlessly insert one image to another is ∇g. Then, (4.2) is changed to

$$\Delta f = \Delta g \quad \text{over} \quad \mathbf{\Omega} \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}. \tag{4.3}$$

The equation (4.3) means that the Laplacian of the inserted region is identical to that of the source while pixel intensities of the destination over the inserted region's boundary remain unchanged.

Figure 4-6 shows example frames of the condensed video after applying either naïve alpha blending or Poisson image editing for the stitching process. As you can see in the figure, results using Poisson image editing are more visually natural but it takes more computation time than the naïve approach.

4.4 Discontinuity of motion flow

As shown in Figure 4-7, online video synopsis generates a small portion of the condensed video containing K object tubes after each tube rearrangement step. If we have $20 \times K$ object tubes, there will be 20 portions of the condensed video. At the time of the user request, these portions are merged into the complete synopsis video. During the merging process, if we could not properly handle the transitions between the one portion to another, the users might notice the abrupt changes in the scene. This problem is called as a discontinuity of motion flow. One simple yet efficient way to prevent such discontinuity is considering tails of the object tubes in the previous iteration during the current step [20]. Figure 4-8 shows diagrams to explain the solution.

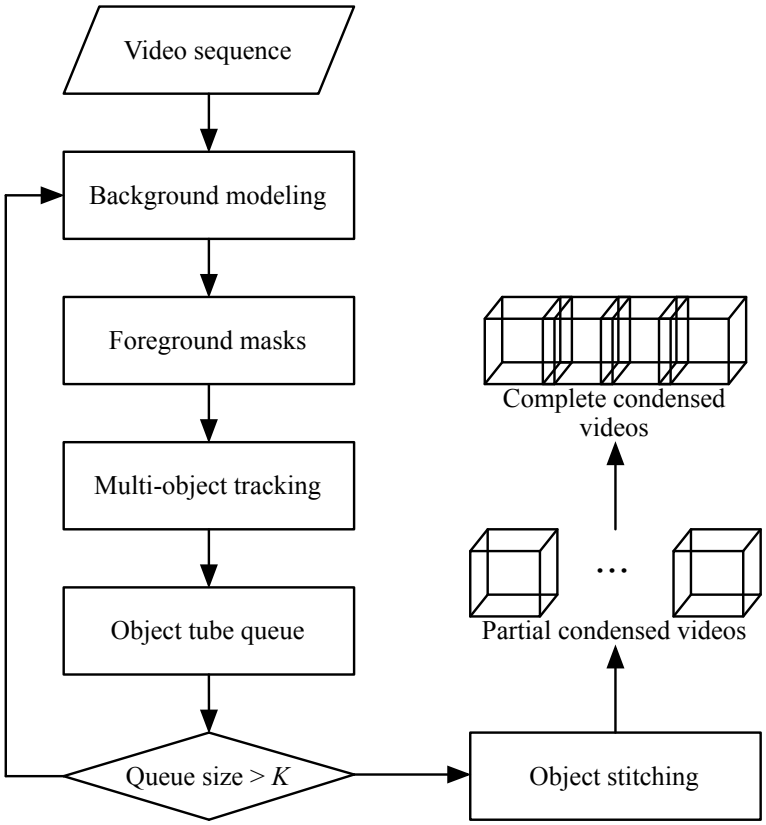
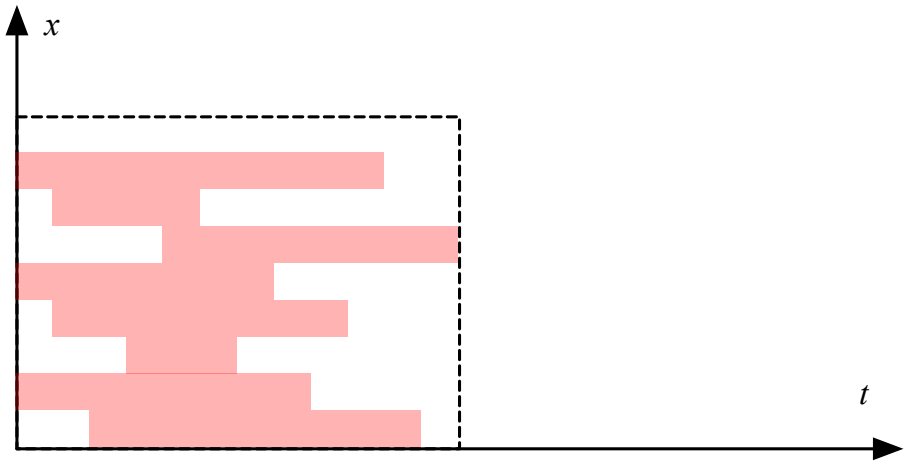
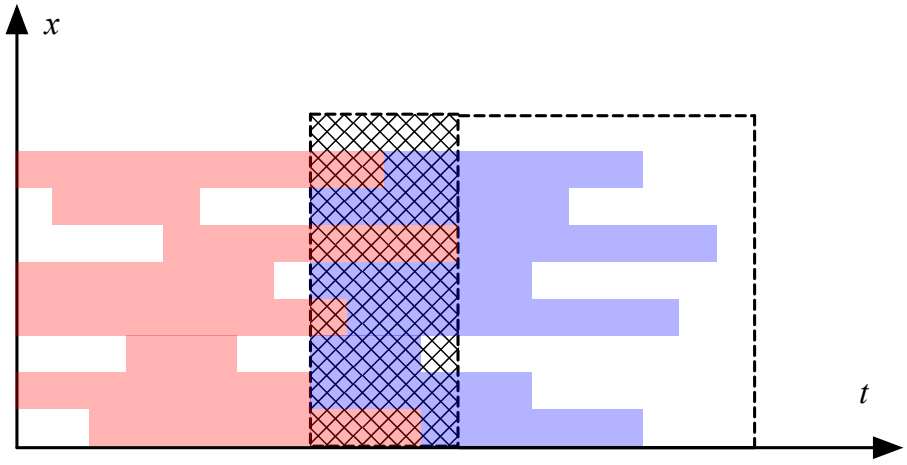


Figure 4-7. Proposed online video synopsis framework. The framework generates a partial condensed video whenever the size of the queue exceeds K. Then, partial videos are merged into the complete synopsis video.



(a) Rearranged tubes after 1st iteration



(b) Rearranged tubes after 2nd iteration

Figure 4-8. Simple solution for the discontinuity of motion flow problem. When finding optimum starting labels for 2nd iteration, tails of the object tubes rearranged at 1st iteration (patterned region) are considered as obstacles as shown in (b).

5 Experimental Results

5.1 Performance metrics

In this chapter, the performance of the proposed tube rearrangement algorithm is evaluated by using four metrics: frame condensation ratio (FR), compact ratio (CR), overlap ratio (OR), and running time (RT). The detail of each performance metric is presented as follows.

FR is defined as a ratio of the condensed video length to the original video.

$$\mathrm{FR} = \frac{\mathcal{T}^*}{\mathcal{T}}, \tag{5.1}$$

where $\mathcal{T}^*$ and $\mathcal{T}$ are lengths of the condensed and original videos, respectively. Smaller FR is better for reducing time consumption of browsing contents of the video.

CR indicates that how many pixels in the condensed video are occupied by the objects and is defined as

$$\mathrm{CR} = \frac{1}{\mathcal{W}\mathcal{H}\mathcal{T}^*} \sum_{x=1}^{\mathcal{W}} \sum_{y=1}^{\mathcal{H}} \sum_{t=1}^{\mathcal{T}^*} F^*(x,y,t) = \frac{|F^*|}{\mathcal{W}\mathcal{H}\mathcal{T}^*}, \tag{5.2}$$

where F^* is a foreground volume of the condensed video. Each element of F^* is defined as

$$F^*(x,y,t) = \begin{cases} 1 & I^*(x,y,t) \neq B_t(x,y,t) \\ 0 & \text{otherwise,} \end{cases} \tag{5.3}$$

where $I^*(x,y,t)$ is a pixel of the condensed video and $B_t(x,y,t)$ is a pixel of the time-lapse background before the stitching process. A large CR indicates that the tube rearrangement algorithm effectively utilizes the spatio-temporal domain of the synopsis video.

OR is proportional to the number of overlapped foreground pixels in the condensed video and defined as

$$\text{OR} = \frac{1}{|F^*|} \sum_{x=1}^{\mathcal{W}} \sum_{y=1}^{\mathcal{H}} \sum_{t=1}^{\mathcal{T}^*} O(x,y,t) = \frac{|O|}{|F^*|}, \tag{5.4}$$

where $O(x,y,t)$ is activated to 1, when $I^*(x,y,t)$ is a result of blending foreground pixels of two or more objects with $B_t(x,y,t)$; otherwise, it produces 0. If OR is small, we can easily distinguish rearranged objects in the synopsis video; therefore, we can understand the summarized information better.

The last but not least RT is a metric to compare computational complexity of the algorithm and measured in seconds. This metric is important when the framework responds to the requests of users. Smaller RT is better for reducing a latency of the system. All experiments in following sections are conducted on a 4-core 4.0 GHz computer with 32 GB of memory.

5.2 Test video sequences

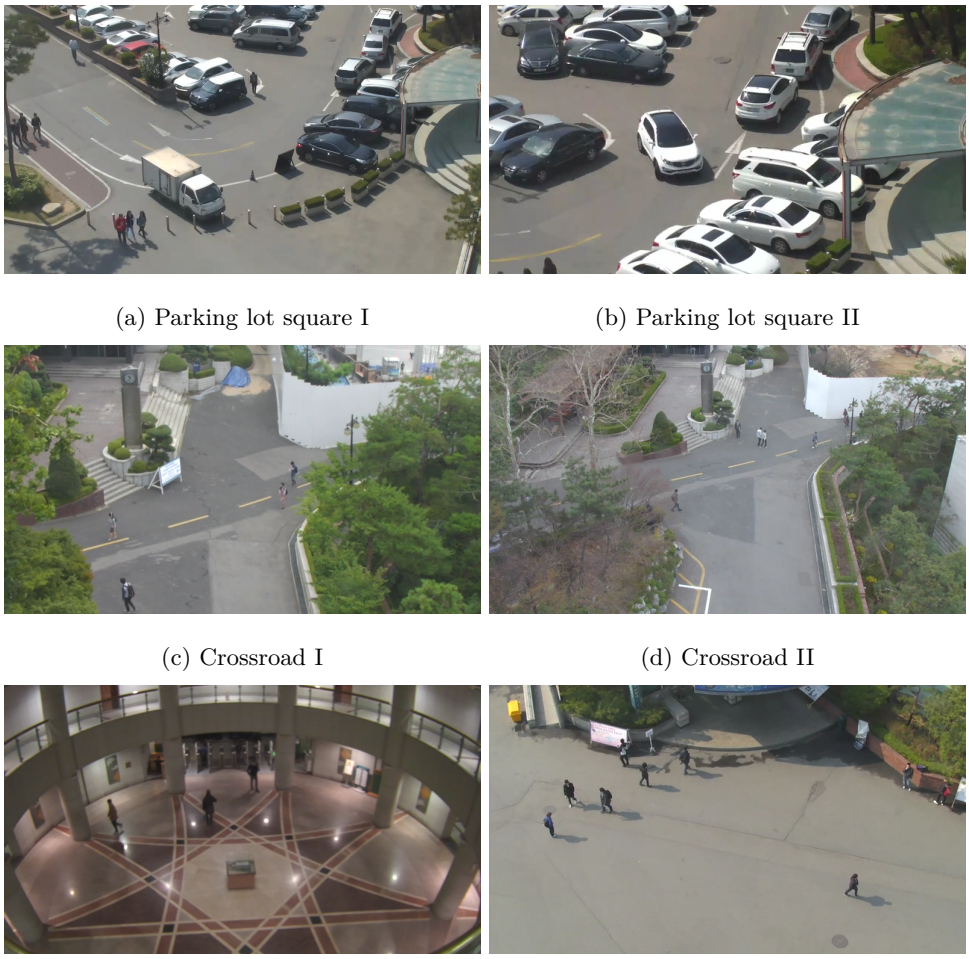
For the test sequences, six video clips are captured at four different places: a parking lot square, a crossroad, a library lobby, and a subway station plaza. Detail characteristics of the test sequences are summarized in Table 5-1. Some examples of the test sequences are depicted in Figure 5-1.

Parking lot square I sequence mainly focuses on the entrance of the parking lot but there is a sidewalk with red bricks on the left. Since this place is a main road to most of buildings in Hanyang university, the scene is very crowded with people. The scene of Parking lot square II is similar to that of Parking lot square I, but the most of moving objects are vehicle not pedestrian.

Crossroad I and II sequences are captured at the same place with different seasons and camera's zoom parameter. Crossroad I is captured at summer with more zoom, while Crossroad II is captured at fall with less zoom. Most of people appeared in this scene walk in either left or right directions.

Library lobby sequence is captured by the indoor security camera mounted on 2nd floor of the building. There is a gateway of the library at top of the scene and stairs (not visible in the scene) are located at left and right side of the building; therefore, people move from either left or right to the top, or vice versa.

Subway station plaza is an open place in front of the subway station entrance. Since there are many ways to get to the building from here, there is no dominant walking direction of people.



(a) Parking lot square I (b) Parking lot square II

(c) Crossroad I (d) Crossroad II

(e) Library lobby I (f) Subway station plaza I

Figure 5-1. Examples of the test sequences. All sequences were captured with three PTZ cameras at Hanyang university, Seoul, Korea.

Symbol	Video clip name	Resolution	# Frame	# Tube
VC1	Parking lot square I	1280×720	44,057	650
VC2	Parking lot square II	640×360	107,946	271
VC3	Crossroad I	640×360	85,766	291
VC4	Crossroad II	640×360	106,459	937
VC5	Library lobby I	1280×720	49,679	316
VC6	Subway station plaza I	640×360	107,876	1038

Table 5-1. List of test sequences used in the experiments.

5.3 Performance analysis

The experiments in this section are designed to 1) find optimum parameters for the proposed tube rearrangement algorithm, 2) conduct an ablation study for two speed up techniques, and 3) compare performances of several different online tube rearrangement algorithms.

5.3.1 Optimum parameters

There are four necessary parameters for the proposed tube rearrangement algorithm: the weight parameter of the length energy λ, the size of the occupation matrix $\mathcal{M} \times \mathcal{N}$, the type of the occupation matrix, and the size of the queue K. Apart from six test sequences used for the performance evaluation, additional five video clips are prepared to find optimum parameters and detail characteristics of the videos are summarized in Table 5-2.

Symbol	Video clip name	Resolution	# Frame	# Tube
VC7	Crossroad III	1280×720	106,558	831
VC8	Library lobby II	1280×720	108,091	3,166
VC9	Subway station plaza II	1280×720	134,622	2,311
VC10	Subway station plaza III	1280×720	108,016	2,048
VC11	Subway station plaza IV	1280×720	108,012	1,988

Table 5-2. List of sequences used to find optimum parameters of the proposed tube rearrangement algorithm.

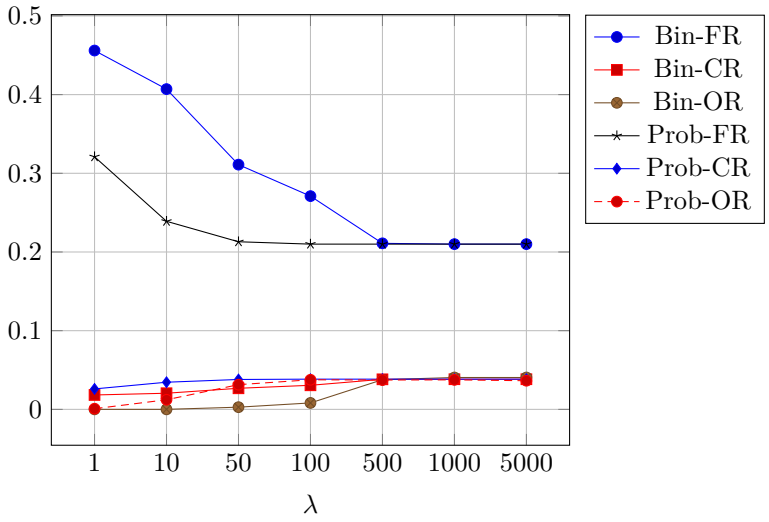
Weight parameter of length energy

First experiment measures the four performance metrics by changing λ from 1 to 5000. Remaining parameters are fixed as $\mathcal{M} \times \mathcal{N} = 9 \times 16$, $K = 20$, and the algorithm produces results for both binary and probabilistic occupation matrices. Except for RT, other three metrics have similar scales; hence, FR, CR and OR are depicted together in Figure 5-2. On the other hand, RT for five sequences are grouped and summarized in Figure 5-3.

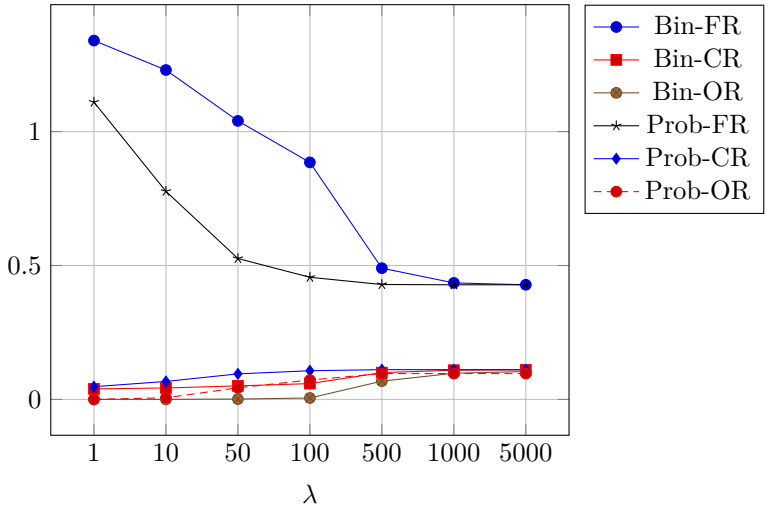
As expected, FR decreases when the proposed algorithm pays more attention to the length energy (increasing λ). On the other hand, CR and OR do not change as much as FR. As shown in Figure 5-3, we can see that RT is proportional to the number of object tubes in the original video, and tends to decrease as λ increases. This result is quite obvious, because small number of object tubes and less frames to consider reduce the computational burden.

One interesting result is shown in Figure 5-2b, where FR is larger than 1 for $\lambda \in \{1, 10, 50\}$ with binary occupation matrix and $\lambda = 1$ with probabilistic occupation matrix. This result is induced by two factors: small λ and large number of object tubes having similar paths. When λ is small, the proposed algorithm focuses on reducing collisions rather than making a short length video. In addition, when large number of objects share the common path in the scene, it is hard to avoid collisions between the objects moving along the path. One straightforward solution for the rearrangement problem under these conditions is minimizing the overlapped time region between the objects. In other words, the algorithm rearranges object tubes like linked sausages and the resulting video may have a longer length than the original one. To avoid such undesirable solution, selecting sufficiently large λ is important for the proposed algorithm. However, when λ exceeds some value, four metrics become saturated, which means that the algorithm primarily considers the length energy. This solution is not desirable neither; therefore, we need to choose a balanced value of λ.

For selecting the value of λ, different behaviors of the binary and probabilistic occupation matrices should be considered. As you can see in Figures 5-2 and 5-3, when K increases, the algorithm utilizing the probabilistic occupation matrix approaches to the saturation point faster than the one using the binary occupation matrix. Therefore, with same λ, the probabilistic occupation matrix allows the algorithm to produce the shorter synopsis video than the binary occupation matrix; it other words, it has better FR and CR, but higher OR than its counterpart. Based on the observation, it is better to select different λ for the probabilistic and binary occupation matrices. However, for easier parameter analysis in following sections, both types of the occupation matrix utilize λ of 100.

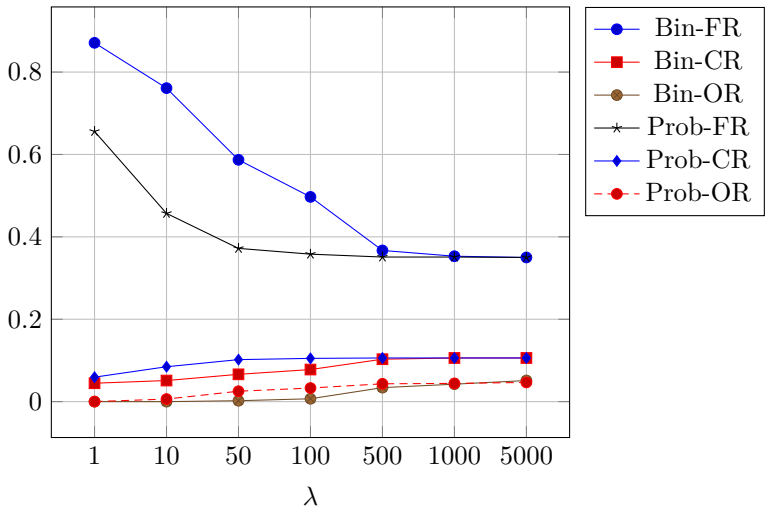


(a) Crossroad III

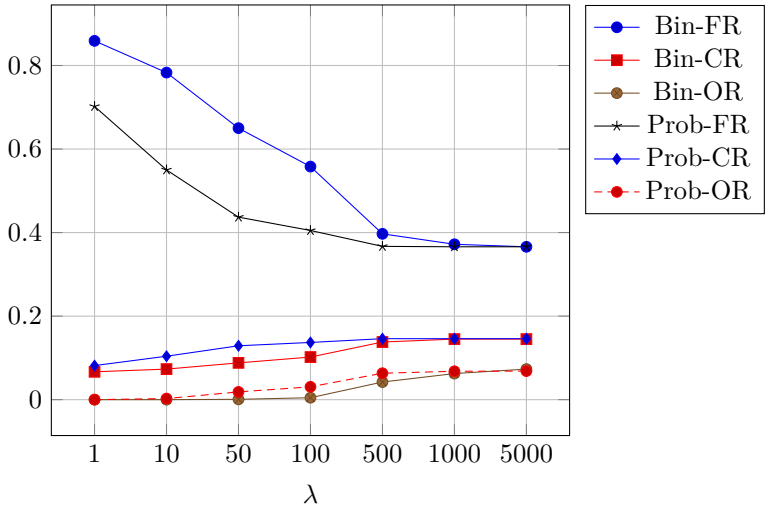


(b) Library lobby II

Figure 5-2. Result of the experiment conducted by changing λ.

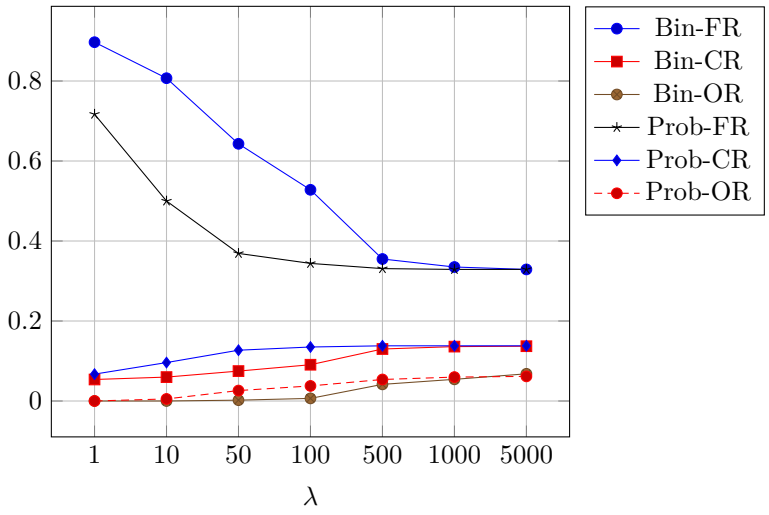


(c) Subway station plaza II



(d) Subway station plaza III

Figure 5-2. (continued) Result of the experiment conducted by changing λ.



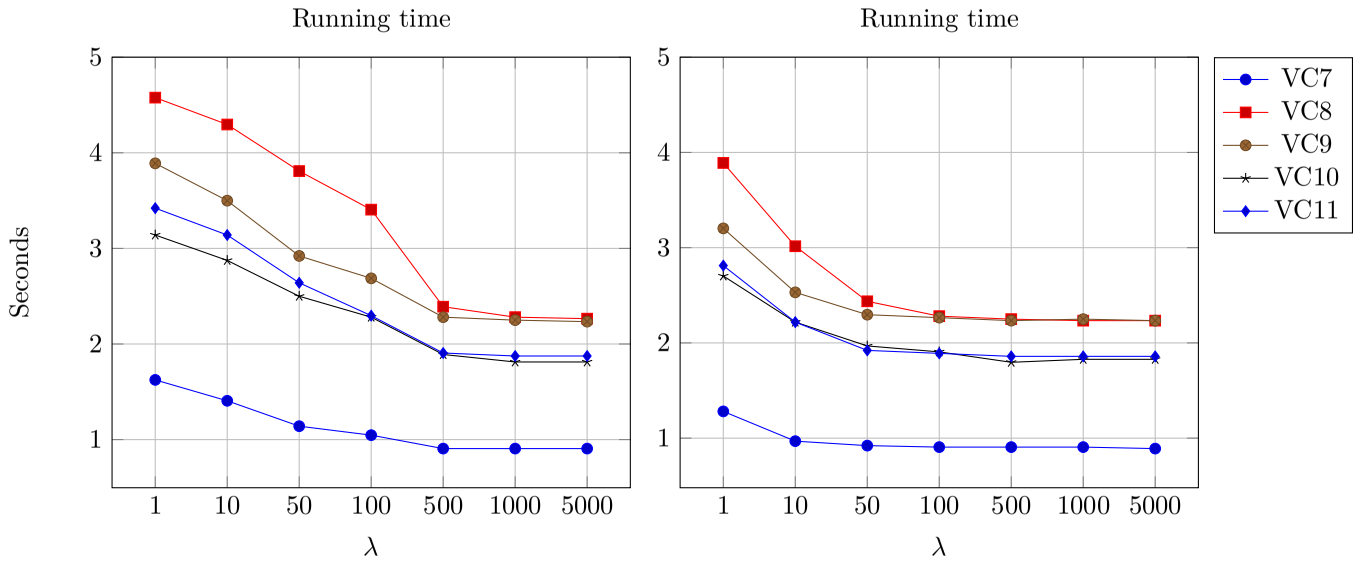
(e) Subway station plaza IV

Figure 5-2. (continued) Result of the experiment conducted by changing λ.

Size of occupation matrix

Second experiment is conducted by changing spatial resolution $\mathcal{M} \times \mathcal{N}$ of the occupation matrix. Assume that the aspect ratio of the input video is 16:9 (the ratio of the width to the height). Based on the assumption, four candidates of $\mathcal{M} \times \mathcal{N}$ are considered in this experiment: 9×16, 18×32, 36×64, and 72×128. Remaining parameters are fixed as $\lambda = 100$ and $K = 20$ and the algorithm utilizes both binary and probabilistic occupation matrices.

The larger binary occupation matrix has a better ability to encode the locations of the objects; therefore, the algorithm can finely adjust starting labels to avoid collisions. Then, the resulting video has lower FR and higher CR than the one using the smaller occupation



(a) Binary occupation matrix

(b) Probabilistic occupation matrix

Figure 5-3. RT of the proposed algorithm measured by changing λ for five sequences.

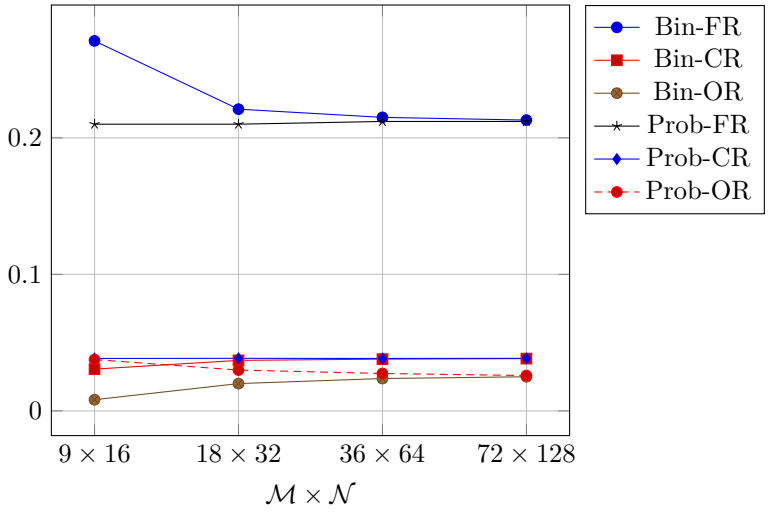
matrix as shown in Figure 5-4. However, interestingly, increasing the resolution of the probabilistic occupation matrix does not affect to the performance regarding three metrics. This indicates that even the low resolution probabilistic occupation matrix has already got enough capability to express fine locations of the objects.

In terms of RT, reducing spatial resolution for both types of the occupation matrix drastically increases the computation speed of the algorithm as depicted in Figure 5-5. According to Table 5-3, in overall, the probabilistic occupation matrix can compute rearranged starting labels faster than the binary occupation matrix; however, the tendency of the computation time for different $\mathcal{M} \times \mathcal{N}$ is almost identical. Especially, the computation time for both types of the occupation matrix is reduced by at least 1/40, when the width and height of the matrix are scaled by 1/8 (from 72×128 to 9×16).

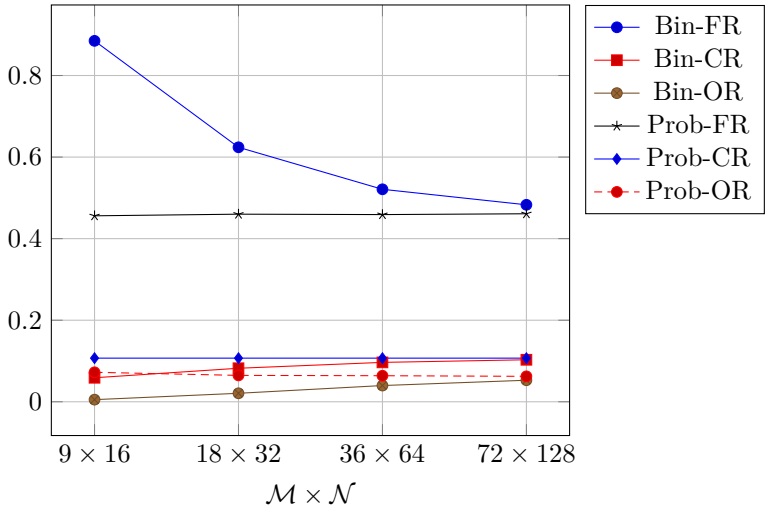
In summary, there are definite advantages in FR and CR for increasing the spatial resolution of the binary matrix; however, the degradation of the computation speed is too serious to be neglected. On the other hand, large probabilistic occupation matrix does not have any advantages regarding the performance metrics, and even with the low resolution matrix, the algorithm can have a better condensation ability than the one uses the high resolution binary matrix. Therefore, for both types of the occupation matrix, the smallest resolution (9×16) is preferred in this dissertation.

Size of queue

Adjusting the size of the queue K determines how many object tubes are considered during each tube rearrangement step. The experiment is conducted as same manner as in Sec-

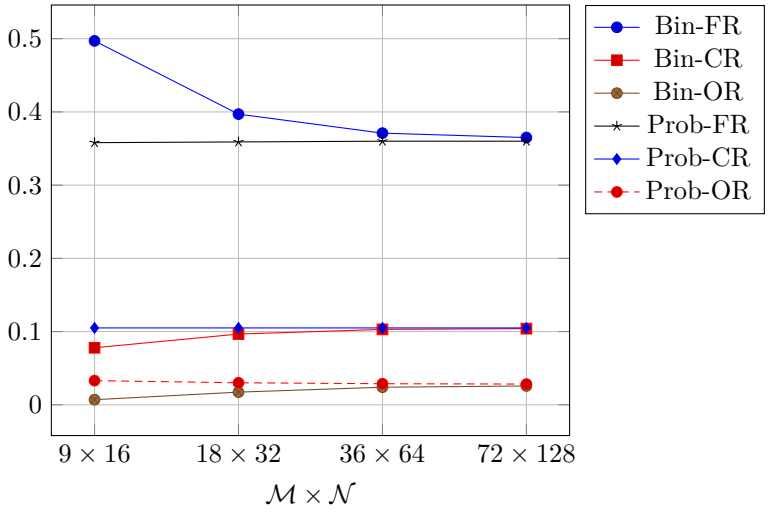


(a) Crossroad III

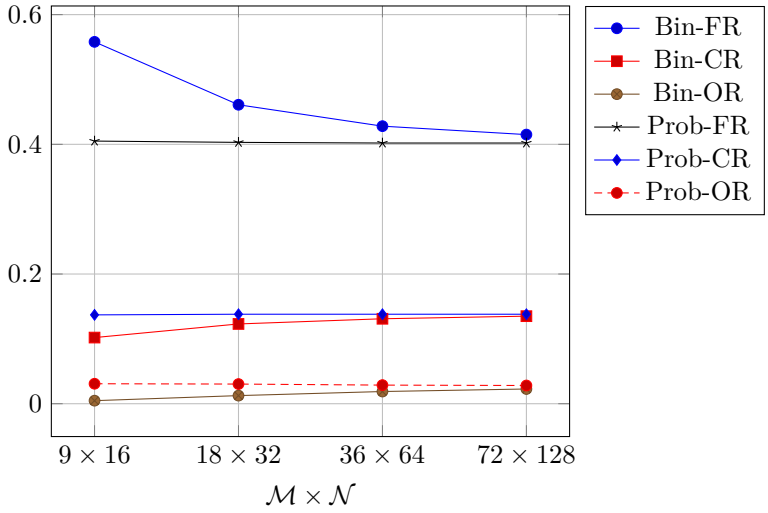


(b) Library lobby II

Figure 5-4. Result of the experiment conducted by changing $\mathcal{M} \times \mathcal{N}$.



(c) Subway station plaza II

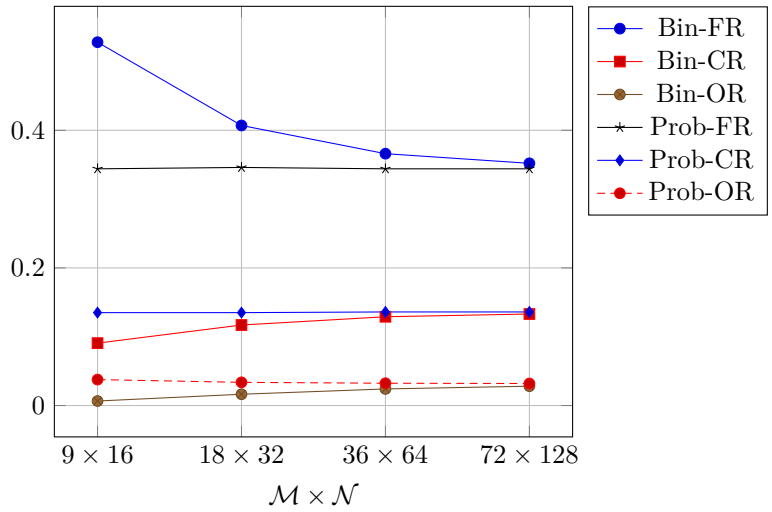


(d) Subway station plaza III

Figure 5-4. (continued) Result of the experiment conducted by changing $\mathcal{M} \times \mathcal{N}$.

	VC7		VC8		VC9		VC10		VC11	
	Bin.	Prob.	Bin.	Prob.	Bin.	Prob.	Bin.	Prob.	Bin.	Prob.
9×16	1.11	0.91	3.36	2.31	2.66	2.3	2.28	1.92	2.28	1.91
18×32	3.56	3.7	9.89	8.54	8.59	8.33	7.47	7	7.33	6.95
36×64	14.64	14.54	34.79	32.65	32.7	32.63	27.79	28.42	27.43	27.1
64×128	61.3	60.19	139.7	135.67	145.51	144.45	121.28	119.57	118.13	117.89

Table 5-3. RT of the proposed algorithm measured in seconds by changing $\mathcal{M} \times \mathcal{N}$ for five video clips.

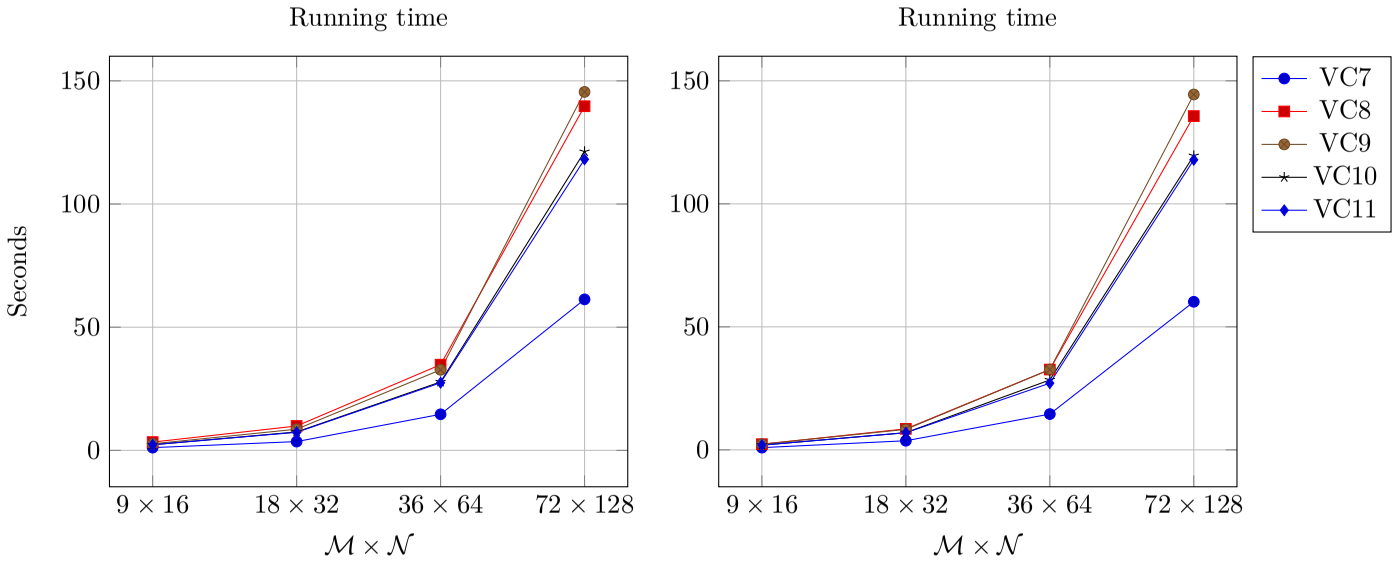


(e) Subway station plaza IV

Figure 5-4. (continued) Result of the experiment conducted by changing $\mathcal{M} \times \mathcal{N}$.

tion 5.3.1, while K is changed from 10 to 100 with 10 interval. Other parameters are fixed to $\lambda = 100$ and $\mathcal{M} \times \mathcal{N} = 9 \times 16$, and both binary and probabilistic occupation matrix are used as spatial approximations of the foreground pixels. Results of the experiments are presented in Figures 5-6 and 5-7.

For the binary occupation matrix, Figure 5-6 shows that, for increasing K, FR is decreased by at least 0.2 and CR is slightly increased, while OR remains almost unchanged. On the other hand, in general, the algorithm using the probabilistic matrix produces shorter synopsis videos with higher CR and OR. The key difference between the binary and occupation matrices is that the algorithm using the binary matrix produces synopsis videos having constant OR regardless of K. In other words, the resulting synopsis video never becomes more crowded even if K is large.



(a) Binary occupation matrix

(b) Probabilistic occupation matrix

Figure 5-5. RT of the proposed algorithm measured by changing $\mathcal{M} \times \mathcal{N}$ for five sequences.

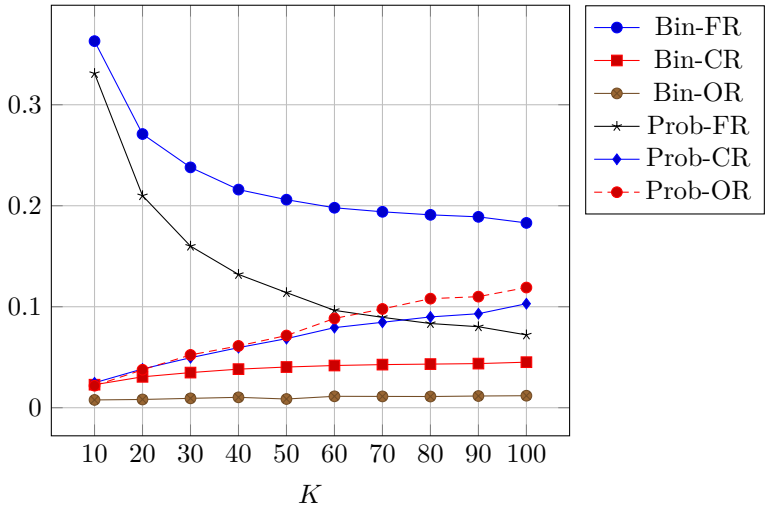
It is obvious that the algorithm takes more time to get rearranged labels for increasing K as shown in Figure 5-7. Additionally, the computational advantage of the probabilistic matrix over the binary matrix is clearly seen in the result. As discussed many times throughout Section 5.3.1, since the algorithm using the probabilistic matrix produces shorter synopsis videos than the one using the binary matrix, the number of frames to consider during the tube rearrangement stage becomes smaller. In consequence, slopes of Figure 5-7b are less steeper than that of Figure 5-7a.

Similar to selecting λ, there is a trade-off between RT and other metrics for selecting K. Empirically, for both binary and probabilistic matrices, a median of the candidate values, 50, is used throughout the subsequent experiments.

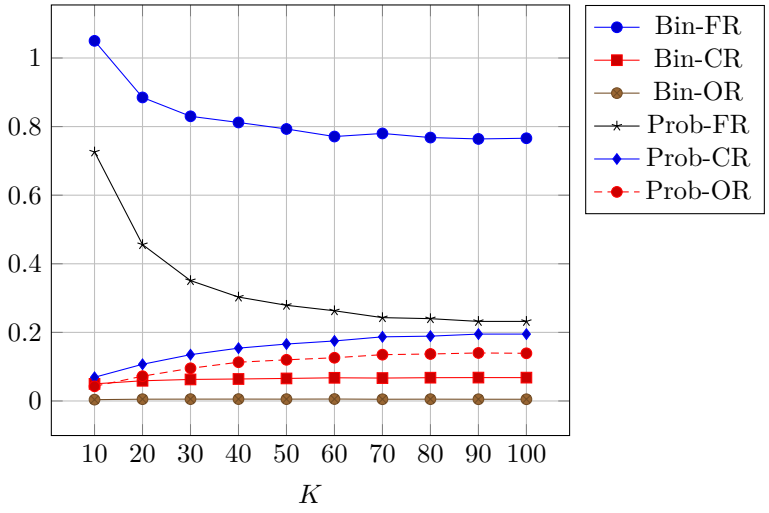
Type of occupation matrix

Based on the observations so far, with same parameters, the binary occupation matrix has a benefit for OR over the probabilistic occupation matrix, which means that the resulting synopsis video is less crowded and the objects in the video can be easily distinguished from each other. On the other hand, the algorithm using the probabilistic matrix outperforms the one utilizing the binary matrix for every performance metric except OR. Therefore it produces more crowded but compact synopsis video than the binary equivalent.

In conclusion, using the probabilistic occupation matrix is beneficial for the most of situations. If the user wants a less crowded condensed video, using the binary occupation matrix can be an option. However, instead of changing the type of the occupation matrix, adjusting λ is more simple and desirable solution.

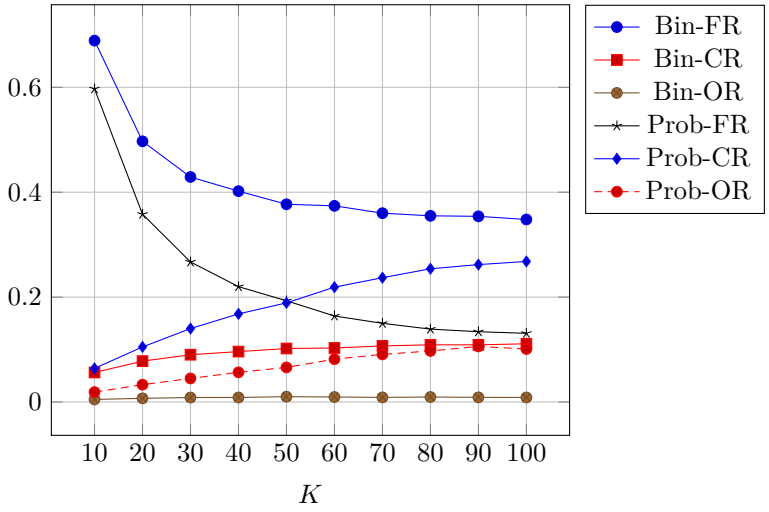


(a) Crossroad III

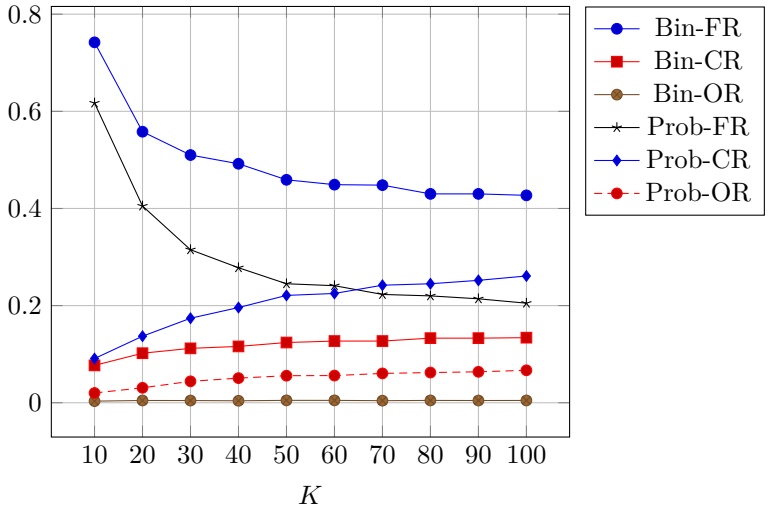


(b) Library lobby II

Figure 5-6. Result of the experiment conducted by changing K.

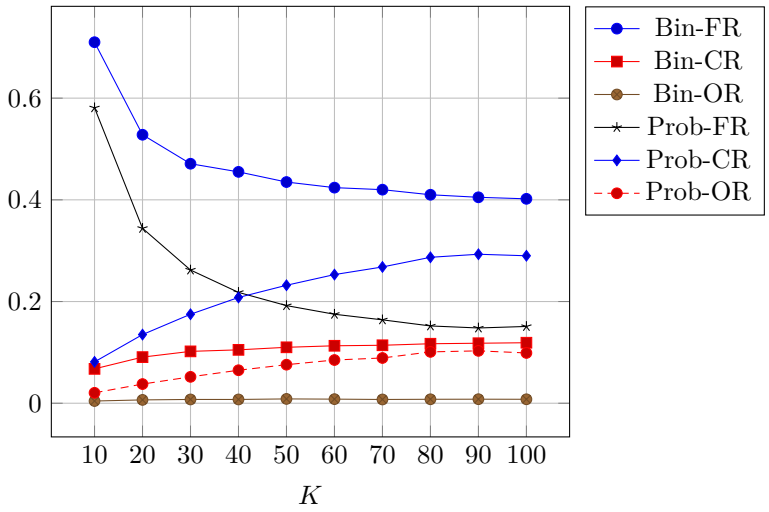


(c) Subway station plaza II



(d) Subway station plaza III

Figure 5-6. (continued) Result of the experiment conducted by changing K.



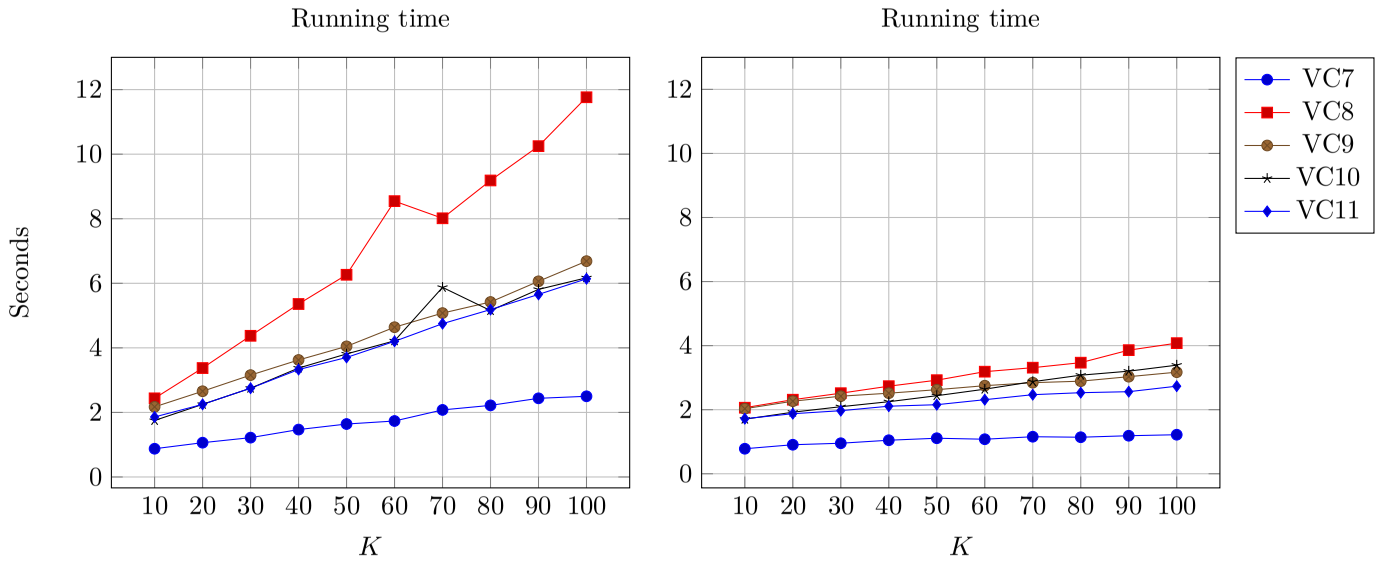
(e) Subway station plaza IV

Figure 5-6. (continued) Result of the experiment conducted by changing K.

5.3.2 Ablation study for speed up techniques

In this section, ablation study for the two speed up techniques used in the proposed algorithm is conducted: FFT and parallel processing. Total four versions of the algorithm are compared regarding RT as summarized in Table 5-4. The baseline of the algorithm denoted as Occ utilizes the occupation matrix and the cross-correlation of the signals in time domain to calculate collisions between the objects. For all versions of the algorithm, the probabilistic occupation matrix of 9×16 resolution is utilized and other parameters are fixed as $\lambda = 100$ and $K = 50$. As similar to Section 5.3.1, five video sequences from VC7 to VC11 are utilized for the evaluation.

According to the result in Figure 5-8, FFT reduces the computational burden of the



(a) Binary occupation matrix

(b) Probabilistic occupation matrix

Figure 5-7. RT of the proposed algorithm measured by changing K for five sequences.

Symbol	FFT	Parallel processing
Occ	-	-
Occ+F	✓	-
Occ+P	-	✓
Occ+FP	✓	✓

Table 5-4. Four versions of the proposed algorithm used for ablation study.

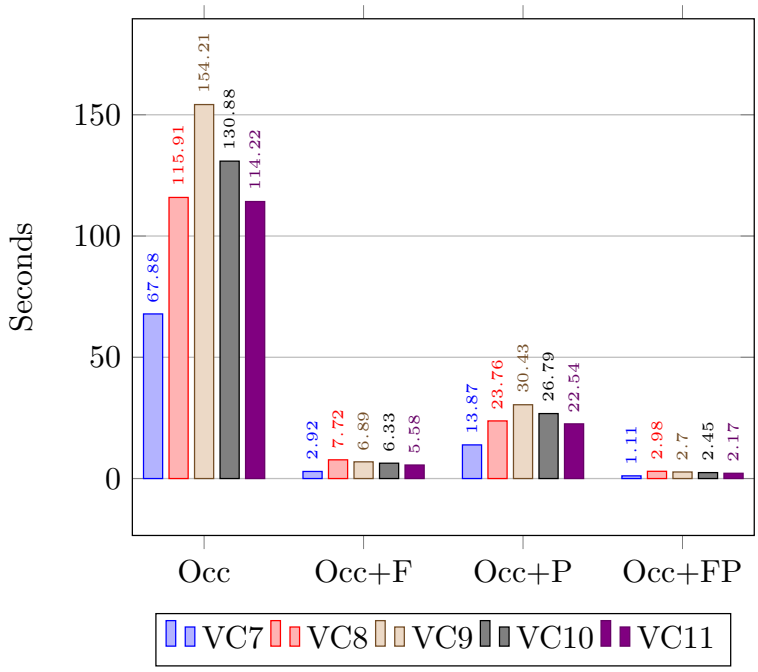


Figure 5-8. Result of the ablation study to compare four different versions of the proposed algorithm.

proposed algorithm by at least 1/20. On the other hand, applying parallel computing to the algorithm increases its computation speed by 5 times in average. Utilizing both speed up techniques (Occ+FP) produces the best result in RT; however, the performance gain from Occ+F to Occ+FP is less drastic than the one from Occ to Occ+P; the proposed algorithm with Occ+FP only produces approximately 2.5 faster results than the non-parallelized version.

5.3.3 Performance comparisons

To compare the performance of the proposed algorithm with others, three recently introduced online tube rearrangement algorithms [18–20] are reproduced by using C/C++ languages. For fair comparisons, the existing algorithms utilize the same values of the required parameters as described in the original papers. The values of the parameters required for the proposed algorithm are summarized as follows: $\lambda = 100$, $\mathcal{M} \times \mathcal{N} = 9 \times 16$, $K = 50$, and the probabilistic occupation matrix. Except for the Tetris like optimization [18] which utilizes its own online tube filling strategy, all algorithms run on the same online video synopsis framework which maintains a fixed sized queue of K for containing moving objects. Note that existing algorithms have not employed any spatial subsampling.

Regarding RT, as shown in Figure 5-9, the proposed algorithm outperforms other algorithms with a large margin. Since the algorithm of Fu *et al.* [20] tries to group object tubes possible to have interactions and put them into the same portion of the condensed video, it takes much more time in computing rearranged labels even though it has benefits from the online framework. For other two algorithms, we can see that the key concepts of the

optimization (greedy search [18] and PCG [19]) definitely contributes to the performance, but their difference is not significant.

According to Figure 5-10, average FR for each algorithm can be listed as 0.0951, 0.0853, 0.0686, and 0.102 in order; therefore, the best performance is achieved by the algorithm of Zhu *et al.* [18]. On the other hand, performances of other three algorithms are not remarkably different. Note that even though the proposed algorithm utilizes the spatial approximation of the foreground, its FR is comparable to other algorithms.

For CR, as we can see in Figure 5-11, Zhu *et al.* [18] performs the best. This result can be expected, because FR and CR have a weak positive correlation; the algorithm with better FR are more likely to have better CR as well. However, the proposed algorithm achieves the second best performance in average, even though it has been ranked at the 3rd place for FR.

Typically, FR and OR have a weak negative correlation, because higher FR means that the spatio-temporal domain of the condensed video is smaller than the one with lower FR; in consequence, the objects are more likely to have collisions between them. However, the performance ranking for OR is very different from that for FR. According to Figure 5-12, the list of average OR for each algorithm is 0.1467, 0.285, 0.2017, and 0.21 in order. The proposed algorithm has been ranked at the 1st position and the Tetris like optimization [18] follows it. The algorithm of Fu *et al.* [20] takes the 3rd position and the PCG baesd algorithm [19] performs the worst.

Except for the algorithm considering the structured motion [20], the objective of other three algorithms is focused on improving the computation speed of the online video synopsis framework. Thanks to the concepts they are using, all of the algorithms can rearrange start-

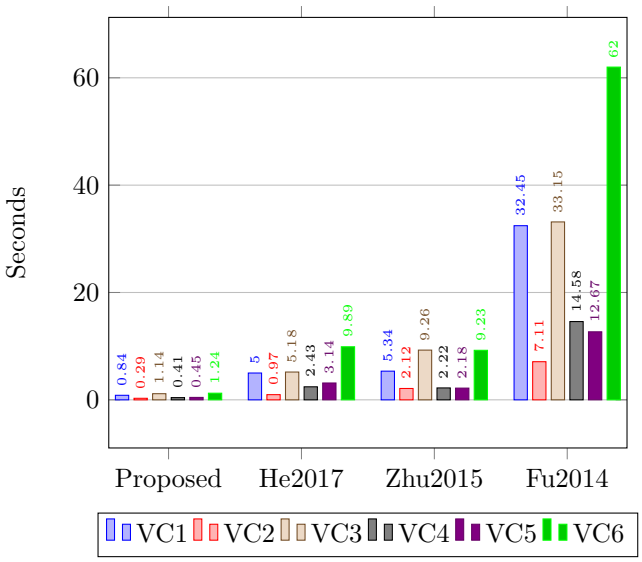


Figure 5-9. Result of the experiment regarding RT measured in seconds.

ing labels within 10 seconds at maximum for the test sequences. However, the algorithms have their own distinctive characteristics. For the Tetris like optimization [18], it definitely has benefits for both FR and CR, which means that it can generate the shorter condensed video and can utilize the target spatio-temporal domain efficiently. For the algorithm of He *et al.* [19], the concept of PCG takes an advantage in RT and FR, where it achieves the 2nd best performance for both metrics. However, as compared to FR, its CR and OR are less than expected, which indicates that it produces suboptimal solutions for the rearrangement task. On the other hand, thanks to the occupation matrix with two speed up techniques (Fourier transform and parallel processing), the proposed algorithm is outmatched other algorithms regarding RT and OR. This indicates that the user can get the visually untangled synopsis video with a low latency through the proposed online video synopsis framework.

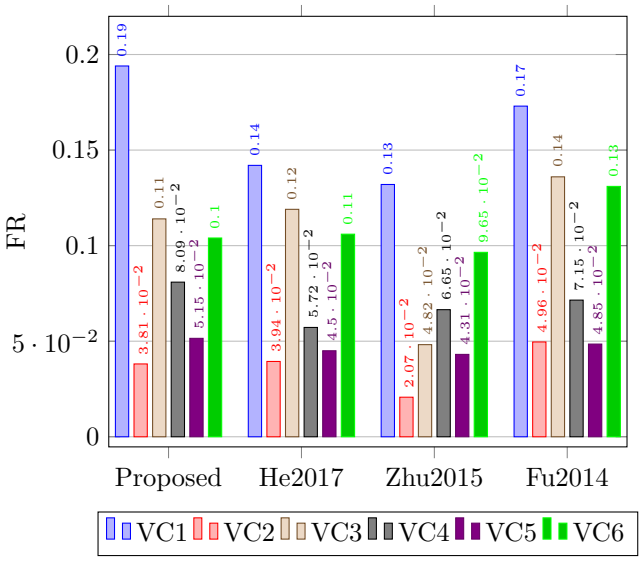


Figure 5-10. Result of the experiment regarding FR.

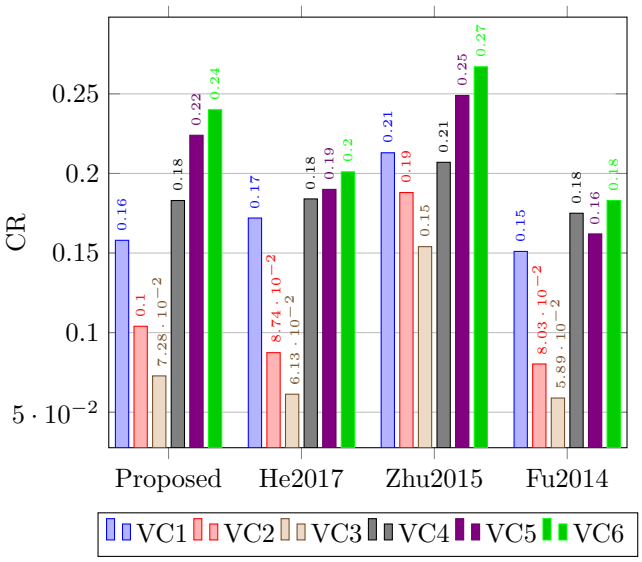


Figure 5-11. Result of the experiment regarding CR.

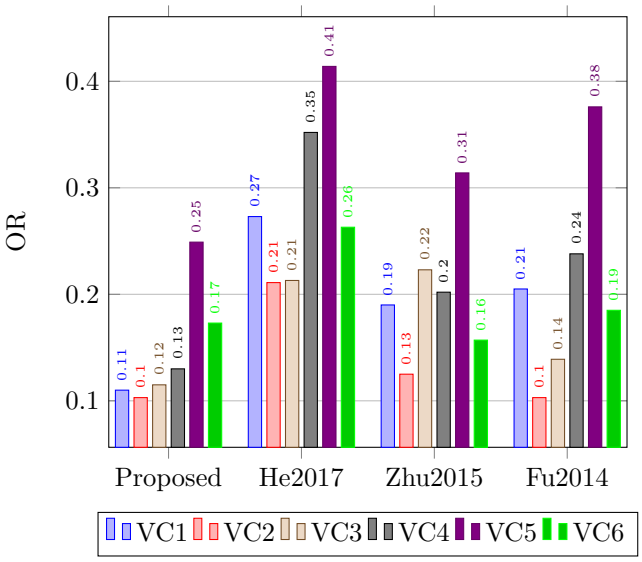


Figure 5-12. Result of the experiment regarding OR.

6 Conclusion

This letter proposes the parallelized tube rearrangement algorithm for online video synopsis. For reducing the computational bottleneck caused by pairwise energy terms, the proposed algorithm concurrently computes the collision energy defined as multiplications of 3D occupation matrices by using an FFT. Throughout the experiments, the proposed algorithm outperformed existing algorithms in terms of computation time and overlap ratio, while its other performance metrics were comparable with those of other algorithms. Our future work will involve running the proposed algorithm on a GPU.

BIBLIOGRAPHY

[1] Y. Pritch, S. Ratovitch, A. Hendel, and S. Peleg, "Clustered Synopsis of Surveillance Video," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, Sep. 2009, pp. 195–200. [Online]. Available: http://ieeexplore.ieee.org/document/5280098/

[2] L. A. Lim and H. Yalim Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, pp. 256–262, Sep. 2018. [Online]. Available: https://doi.org/10.1016/j.patrec.2018.08.002

[3] D. Sakkos, E. S. L. Ho, and H. P. H. Shum, "Illumination-Aware Multi-Task GANs for Foreground Segmentation," *IEEE Access*, vol. 7, pp. 10 976–10 986, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8606933/

[4] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 313–318, Jul. 2003. [Online]. Available: http://portal.acm.org/citation.cfm?doid=882262.882269

[5] C. Cuevas, E. M. Yáñez, and N. García, "Labeled dataset for integral evaluation of moving object detection algorithms: Lasiesta," *Computer Vision and Image Understanding*, vol. 152, pp. 103–117, Nov. 2016. [Online]. Available: https://doi.org/10.1016/j.cviu.2016.08.005

[6] P. W. Patil and S. Murala, "MSFgNet: A Novel Compact End-to-End Deep Network for Moving Object Detection," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, Nov. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8546771/

[7] M. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 1997, pp. 775–781. [Online]. Available: http://ieeexplore.ieee.org/document/609414/

[8] N. Petrovic, N. Jojic, and T. S. Huang, "Adaptive Video Fast Forward," *Multimedia Tools and Applications*, vol. 26, no. 3, pp. 327–344, Aug. 2005. [Online]. Available: https://doi.org/10.1007/s11042-005-0895-9

[9] B. Höferlin, M. Höferlin, D. Weiskopf, and G. Heidemann, "Information-based adaptive fast-forward for visual surveillance," *Multimedia Tools and Applications*, vol. 55, no. 1, pp. 127–150, Oct. 2011. [Online]. Available: https://doi.org/10.1007/s11042-010-0606-z

[10] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a Long Video Short: Dynamic Video Synopsis," in *2006 IEEE Computer Society Conference on Computer Vision*

and Pattern Recognition, vol. 1, Jun. 2006, pp. 435–441. [Online]. Available: http://ieeexplore.ieee.org/document/1640790/

[11] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg, "Webcam Synopsis: Peeking Around the World," in *2007 IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/4408934

[12] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological Video Synopsis and Indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1971–1984, Nov. 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4444355/

[13] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, Feb. 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1262177/

[14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983. [Online]. Available: http://www.sciencemag.org/cgi/doi/10.1126/science.220.4598.671

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, 3rd ed. MIT press, 2009.

[16] C.-R. Huang, P.-C. J. Chung, D.-K. Yang, H.-C. Chen, and G.-J. Huang, "Maximum *a Posteriori* Probability Estimation for Online Surveillance Video Synopsis," *IEEE*

Transactions on Circuits and Systems for Video Technology, vol. 24, no. 8, pp. 1417–1429, Aug. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6748870/

[17] Shikun Feng, Zhen Lei, Dong Yi, and S. Z. Li, "Online content-aware video condensation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 2082–2087. [Online]. Available: http://ieeexplore.ieee.org/document/6247913/

[18] Jianqing Zhu, Shikun Feng, Dong Yi, Shengcai Liao, Zhen Lei, and S. Z. Li, "High-Performance Video Condensation System," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1113–1124, Jul. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/6928452/

[19] Y. He, Z. Qu, C. Gao, and N. Sang, "Fast Online Video Synopsis Based on Potential Collision Graph," *IEEE Signal Processing Letters*, vol. 24, no. 1, pp. 22–26, Jan. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7762044/

[20] W. Fu, J. Wang, L. Gui, H. Lu, and S. Ma, "Online video synopsis of structured motion," *Neurocomputing*, vol. 135, pp. 155–162, Jul. 2014. [Online]. Available: https://doi.org/10.1016/j.neucom.2013.12.041

[21] A. V. Oppenheim and R. W. Schafer, *Discrete Time Signal Processing*, 3rd ed. Pearson, 2010.

[22] Y. Nie, H. Sun, P. Li, C. Xiao, and K.-L. Ma, "Object Movements Synopsis via Part Assembling and Stitching," *IEEE Transactions on Visualization and*

Computer Graphics, vol. 20, no. 9, pp. 1303–1315, Sep. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6702519/

[23] X. Zhu, J. Liu, J. Wang, and H. Lu, "Key observation selection-based effective video synopsis for camera network," *Machine Vision and Applications*, vol. 25, no. 1, pp. 145–157, Jan. 2014. [Online]. Available: https://doi.org/10.1007/s00138-013-0519-8

[24] A. Mahapatra, P. K. Sa, B. Majhi, and S. Padhy, "MVS: A multi-view video synopsis framework," *Signal Processing: Image Communication*, vol. 42, pp. 31–44, Mar. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.image.2016.01.002

[25] S.-Z. Wang, Z.-Y. Wang, and R.-M. Hu, "Surveillance video synopsis in the compressed domain for fast video browsing," *Journal of Visual Communication and Image Representation*, vol. 24, no. 8, pp. 1431–1442, Nov. 2013. [Online]. Available: http://doi.org/10.1016/j.jvcir.2013.10.001

[26] Rui Zhong, Ruimin Hu, Zhongyuan Wang, and Shizheng Wang, "Fast Synopsis for Moving Objects Using Compressed Video," *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 834–838, Jul. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6805196/

[27] X. Li, Z. Wang, and X. Lu, "Surveillance Video Synopsis via Scaling Down Objects," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 740–755, Feb. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7353185/

[28] Zhuang Li, P. Ishwar, and J. Konrad, "Video Condensation by Ribbon Carving," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2572–2583, Nov. 2009. [Online]. Available: http://ieeexplore.ieee.org/document/5156267/

[29] K. Li, B. Yan, W. Wang, and H. Gharavi, "An Effective Video Synopsis Approach with Seam Carving," *IEEE Signal Processing Letters*, vol. 23, no. 1, pp. 11–14, Jan. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7312927/

[30] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation," 2018.

[31] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, Aug. 2004, pp. 28–31. [Online]. Available: http://ieeexplore.ieee.org/document/1333992/

[32] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, May 2006. [Online]. Available: https://doi.org/10.1016/j.patrec.2005.11.005

[33] O. Barnich and M. Van Droogenbroeck, "ViBe: A powerful random technique to estimate the background in video sequences," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2009, pp. 945–948. [Online]. Available: https://ieeexplore.ieee.org/document/4959741

[34] ——, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011. [Online]. Available: https://ieeexplore.ieee.org/document/5672785

[35] M. Van Droogenbroeck and O. Paquot, "Background subtraction: Experiments and improvements for ViBe," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2012, pp. 32–37. [Online]. Available: https://ieeexplore.ieee.org/document/6238924

[36] M. Van Droogenbroeck and O. Barnich, "ViBe: A disruptive method for background subtraction," in *Background Modeling and Foreground Detection for Video Surveillance*, T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, Eds. Chapman and Hall/CRC, Jul. 2014, ch. 7, pp. 7.1–7.23. [Online]. Available: http://hdl.handle.net/2268/157176

[37] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The Pixel-Based Adaptive Segmenter," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, Jun. 2012, pp. 38–43. [Online]. Available: http://ieeexplore.ieee.org/document/6238925/

[38] K. Muchtar, F. Rahman, T. W. Cenggoro, A. Budiarto, and B. Pardamean, "An Improved Version of Texture-based Foreground Segmentation: Block-based Adaptive Segmenter," *Procedia Computer Science*, vol. 135, pp. 579–586, 2018. [Online]. Available: https://doi.org/10.1016/j.procs.2018.08.228

[39] L. Maddalena and A. Petrosino, "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications," *IEEE Transactions on Image*

Processing, vol. 17, no. 7, pp. 1168–1177, Jul. 2008. [Online]. Available: https://ieeexplore.ieee.org/document/4527178

[40] ——, "The SOBS algorithm: What are the limits?" in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2012, pp. 21–26. [Online]. Available: https://ieeexplore.ieee.org/document/6238922

[41] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puigtt, and Y. Ruichek, "BSCGAN: Deep Background Subtraction with Conditional Generative Adversarial Networks," in *2018 25th IEEE International Conference on Image Processing*, Oct. 2018, pp. 4018–4022. [Online]. Available: https://ieeexplore.ieee.org/document/8451603/

[42] M. Sultana, A. Mahmood, S. Javed, and S. K. Jung, "Unsupervised deep context prediction for background estimation and foreground segmentation," *Machine Vision and Applications*, vol. 30, no. 3, pp. 375–395, Apr. 2019. [Online]. Available: https://doi.org/10.1007/s00138-018-0993-0

[43] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2015, pp. 3431–3440. [Online]. Available: http://ieeexplore.ieee.org/document/7298965/

[44] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," 2014.

[45] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," 2017.

[46] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 6230–6239. [Online]. Available: http://ieeexplore.ieee.org/document/8100143/

[47] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, Apr. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/7913730

[48] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *15th European Conference on Computer Vision*, Sep. 2018, pp. 833–851. [Online]. Available: https://doi.org/10.1007/978-3-030-01234-2_49

[49] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015. [Online]. Available: https://doi.org/10.1007/s11263-014-0733-5

[50] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 3213–3223. [Online]. Available: https://ieeexplore.ieee.org/document/7780719

[51] C. Cuevas and N. García, "Improved background modeling for real-time spatio-temporal non-parametric moving object detection strategies," *Image and Vision Computing*, vol. 31, no. 9, pp. 616–630, Sep. 2013. [Online]. Available: https://doi.org/10.1016/j.imavis.2013.06.003

[52] T. S. Haines and T. Xiang, "Background subtraction with DirichletProcess mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 670–683, Apr. 2014.

[53] D. Berjón, C. Cuevas, F. Morán, and N. García, "Real-time nonparametric background subtraction with tracking-based foreground update," *Pattern Recognition*, vol. 74, pp. 156–170, Feb. 2018. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.09.009

[54] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, Mar. 1955. [Online]. Available: http://doi.org/10.1002/nav.3800020109

[55] ——, "Variants of the Hungarian method for assignment problems," *Naval Research Logistics Quarterly*, vol. 3, no. 4, pp. 253–258, Dec. 1956. [Online]. Available: https://doi.org/10.1002/nav.3800030404

[56] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, Mar. 1957. [Online]. Available: https://doi.org/10.1137/0105003

[57] F. Bourgeois and J.-C. Lassalle, "An extension of the Munkres algorithm for the assignment problem to rectangular matrices," *Communications of the ACM*, vol. 14, no. 12, pp. 802–804, Dec. 1971. [Online]. Available: http://doi.acm.org/10.1145/362919.362945

[58] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-Based Probabilistic Tracking," in *7th European Conference on Computer Vision*, May 2002, pp. 661–675. [Online]. Available: https://doi.org/10.1007/3-540-47969-4_44