Thesis for the degree of Doctor of Philosophy

Efficient Tube Rearrangement Algorithms for Online Video Synopsis

Moonsoo Ra

Graduate School of Hanyang University

August 2019

Thesis for the degree of Doctor of Philosophy

Efficient Tube Rearrangement Algorithms for Online Video Synopsis

Thesis supervisor: Whoi-Yul Kim

A Thesis submitted to the graduate school of
Hanyang University in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Moonsoo Ra

August 2019

Department of Electronics and Computer Engineering
Graduate School of Hanyang University

This thesis, written by Moonsoo Ra,
has been approved as a thesis for the *Doctor of Philosophy.*

August 2019

Committee Chairman: Prof. Jechang Jeong (Signature)

Committee member: Prof. Whoi-Yul Kim (Signature)

Committee member: Prof. Jong-Il Park (Signature)

Committee member: Prof. Euee S. Jang (Signature)

Committee member: Prof. Joon-Hyuk Chang (Signature)

Graduate School of Hanyang University

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Video synopsis allows us to analyze security videos efficiently by condensing a long video into a short one. To generate a condensed video, moving objects are extracted from the input video in the form of object tubes. Then, these tubes are rearranged in the temporal domain using a predefined objective function. It consists of several energy terms which play important roles in making a visually appealing condensed video. Among them, collision energy creates a bottleneck in the computation because it requires two object tubes as input arguments; in other words, the computational complexity is proportional to square of the number of objects. Existing approaches try to reduce the computation time of the collision energy calculation by reducing the number of tubes processed at once. However, computational complexity of the approaches are not sufficiently low to generate the condensed video when the number of object tubes is large.

This dissertation presents efficient tube rearrangement algorithms targeted for online video synopsis. The proposed algorithms reduce the computational complexity of the collision energy calculation itself by using fast Fourier transform (FFT). For the first step of the computational complexity reduction, occupation matrices,

foreground masks having low resolution, have been introduced to represent coarse locations of the object tubes. Then, collision energy can be computed as a series of element-wise multiplications between the occupation matrices. This process is same as conducting 1D cross-correlations between two sets of signals. Therefore, the computational complexity of the collision energy calculation can be reduced by applying FFT. Moreover, parallel processing can be utilized for further improvements of computation time.

To evaluate and analyze the performance of the proposed algorithms, total 10 hours long videos are captured at four different places of Hanyang University, Seoul Korea. For all the test sequences, the proposed algorithm with FFT can rearrange object tubes within 1.89 seconds in average, and this is at least 2.26 times faster result than existing algorithms. Moreover, the result can be accelerated in conjunction with parallel processing. In this case, it only takes 0.75 seconds in average for the tube rearrangement task. In addition, resulting synopsis videos of the proposed algorithms have less collisions than comparing algorithms. For better understanding of the proposed algorithm, effectiveness of two speed up techniques (FFT and parallel processing) has been analyzed through the ablation study, and the parameter analysis of the proposed algorithm has been conducted extensively.

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 [9–11] prior to video synopsis [7,12,13] 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 [7, 12, 13] 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.* [13], the tube rearrangement problem is formulated as Markov Random Fields (MRFs) [14] 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 [15] or greedy optimization algorithm [16]. During the optimization process, calculating pairwise energy terms (in this case, collision and temporal consistency) becomes a bottleneck for the 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.* [17] 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 [5,6, 18–21], which solves a stepwise optimization problem. In the 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 not necessary to consider temporal and background consistencies. Therefore, most of the 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 [19], a Tetris-like tube rearrangement strategy [18, 20], and a potential collision graph [6, 21]. Even though these existing algorithms have their own virtues, computational complexities of the algorithms are not efficient enough to consider large amount of object tubes, and they have lack of considerations for utilizing multi-core resources to speed up the computation.

This dissertation presents efficient online tube rearrangement algorithms which utilize fast Fourier transform (FFT) [22] and parallel processing. The main role of FFT is to reduce the computational complexity of the collision energy calculation, and the tube rearrangement speed can be further improved by parallel processing. Note that since the proposed algorithm is targeted for online video synopsis, the collision energy is primarily considered during the tube rearrangement like other online tube rearrangement algorithms [5, 6, 18–21].

As a preprocessing step, the proposed algorithm reshapes object tubes into 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 temporal domain. This occupation matrix provides a coarse representation of the foreground masks; therefore $\mathcal{M}$ and $\mathcal{N}$ are much smaller than height and width of the input video. By using the occupation matrix,

the collision energy between two object tubes is defined as element-wise multiplications of two matrices. To determine a staring label of the incoming object tube, collision energies for different starting labels are computed first; then, the starting label having a minimum collision energy is selected as the optimum value. Interestingly, this process is same as conducting $\mathcal{M} \times \mathcal{N}$ 1D cross-correlations between two sets of signals and finding the starting label which produces minimum responses. We can use either ways to solve the tube rearrangement task for online video synopsis. However, the later one which uses 1D cross-correlations has a chance to reduce its computational complexity by using FFT. In addition, applying parallel processing to the algorithm is straightforward. These improvements have been started from using the occupation matrix as a spatial approximation of the foreground mask. More comprehensive explanation will be presented at Chapter 3.

1.2 Related works

After the very first appearance of the video synopsis technique [7], it has been improved in many different aspects. In this dissertation, recent advances in video synopsis are discussed. Especially, details of the online video synopsis framework are presented.

One of the big hurdles for video synopsis is that it works poorly on very crowded scenes. There are two approaches to solve the problem by 1) scaling down object sizes [1] and 2) rearranging objects in both spatial and temporal domain [2]. As illustrated in Figure 1-1, reducing size of the objects can produce the less com-

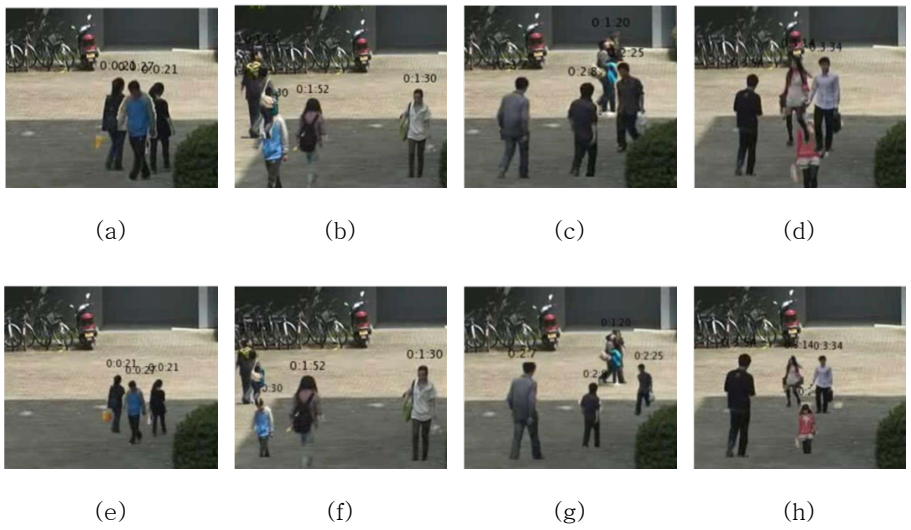


Figure 1-1. Synopsis results by scaling down objects [1]. Because object sizes are not consistent and matched with the context, the results are visually uncomfortable.

plicated condensed video and can have more objects in the scene simultaneously. However, as we can see in the results, scaled objects are visually awkward and some interactions between the objects are hard to understand. On the other hand, Nie *et al.* [2] solve the problem by generating expanded background images and re-arranging object tubes in the spatio-temporal domain. As shown in Figure 1-2, the background images contain synthetically generated regions; width of the sidewalk in the original image has been expanded to triple. Then, the algorithm utilizes such regions to reduce a complexity of the scene. It can produce decent results when the many objects in the scene walk along the same path. However, since generating the synthetic background image is not a straightforward task, the algorithm cannot be applied to videos having complex real-world scenarios. In addition, as similar

Figure 1-2. Result of object rearrangements in both spatial and temporal domains [2]. Images in the first row are from the input videos and synopsis results are presented in the second row. As we can see in the figure, width of the sidewalk has been expanded to triple, so that more objects can be displayed at the single frame simultaneously.

to the work of Li *et al.* [1], the user has a chance to miss importance interactions between the objects.

One straightforward extension of video synopsis is applying it to the multicamera network. Hoshen *et al.* [3] presents a concept of live video synopsis (LVS) in the multicamera network, where cameras belonging to the network have master-slave relationships. When some objects appear in the master camera, their past activities in the slave cameras are given to the users in the form of synopsis videos. Therefore, the users can understand objects' behaviors better and are easy to make a critical decision for the object. Figure 1-3 shows a result of LVS, where past activ-

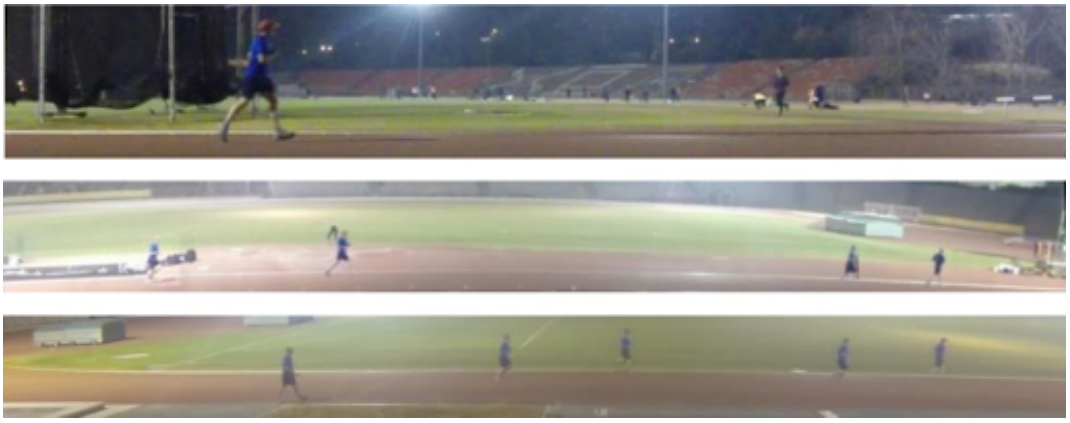


Figure 1-3. Result of LVS [3]. The image in the first row is from the master camera, and remaining images are snapshots of synopsis videos for the slave cameras containing past activities of the objects appearing in the master camera.

ities of the objects in the master camera are simultaneously displayed in the views of the slave cameras. Zhu *et al.* [4] introduces a new cost to keep chronological orders between the object tubes from different cameras. To formulate the cost function, they define four types of key time stamps (KTSs) based on the sticky tracking [18]: start time, merging time, and splitting time, and end time. Then, the object tube is divided into several tracklets according to KTSs. These tracklets become building blocks of calculating the chronological disorder cost function. Figure 1-4 shows the framework to rearrange object tubes in the multicamera network and the result is illustrated in Figure 1-5.

For the online video synopsis, five recently published studies [5, 6, 18, 20, 21] will be discussed. At first, the tube rearrangement algorithm inspired by the video game Tetris [18] has been developed. In the subsequent study [20], the high-

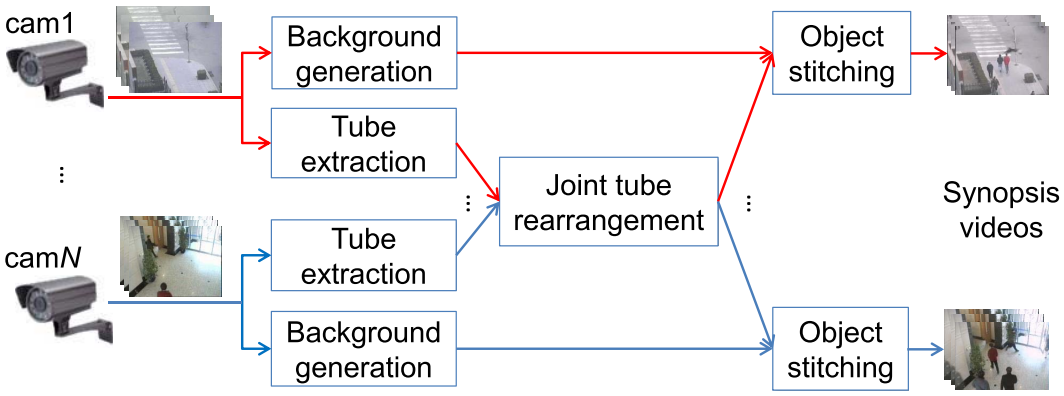


Figure 1-4. Flowchart of the multicamera joint video synopsis [4]. Object tubes extracted from multiple cameras are gathered and optimized together with the joint tube rearrangement. Then, several synopsis videos are generated with rearranged object tubes.

performance online video synopsis framework which utilizes GPU and parallel processing to improve a throughput of the system has been introduced. To apply the concept of Tetris to solve the video synopsis problem, two level cache condensed spaces (L1 and L2) are utilized. The L1 cache is the space, where starting labels of the objects are optimized. If the L1 cache space is filled enough with rearranged object tubes, the algorithm generates a portion of the synopsis video. On the other hand, the L2 space is to hold tails of the object tubes which cannot be placed at L1 space completely. Since the portion of the condensed video is generated only when the L1 space is packed with objects, the length of the resulting video is changed according to the contents of the original video.

Aside from the interesting tube filling strategy, it utilizes a simple greedy opti-

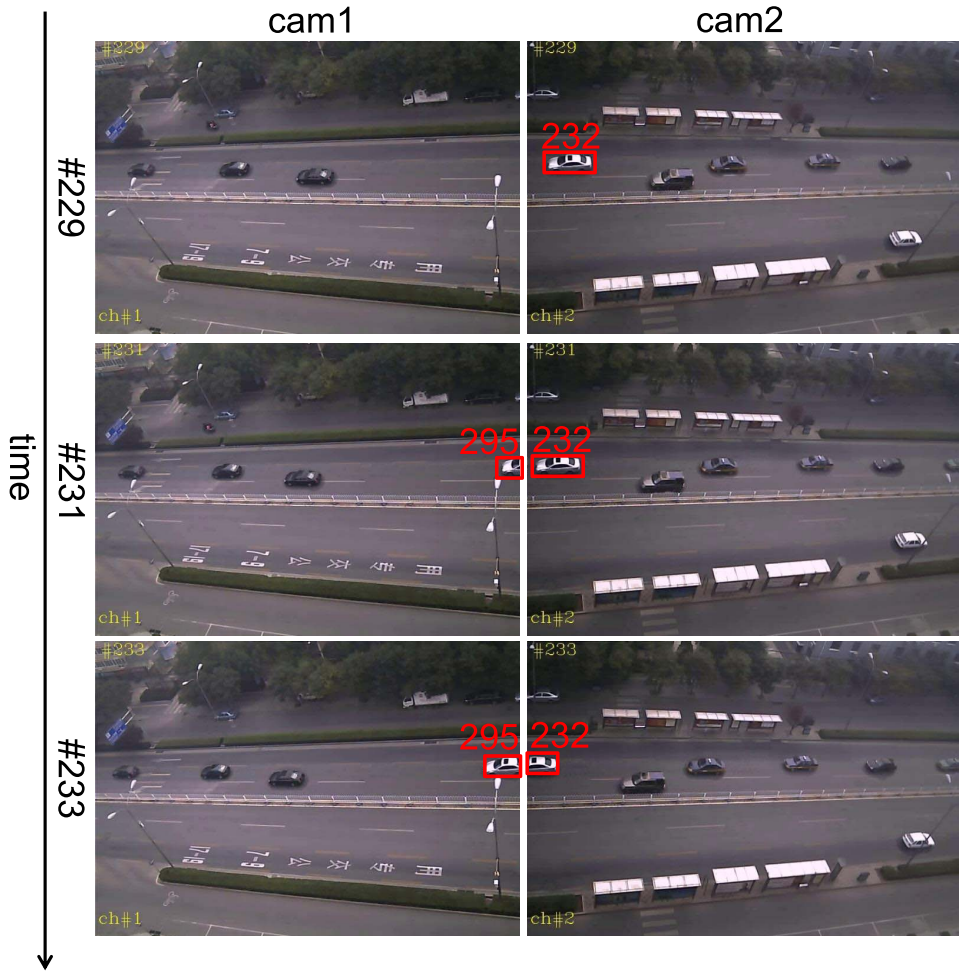


Figure 1-5. Result of the multicamera joint video synopsis [4] with two cameras placed along with the road. Since the white sedan with the red rectangle is driving from cam2 to cam1, it is more natural to see the object in the synopsis video of cam2 first. After that the object is shown in the condensed video of cam1 with the label of 295.

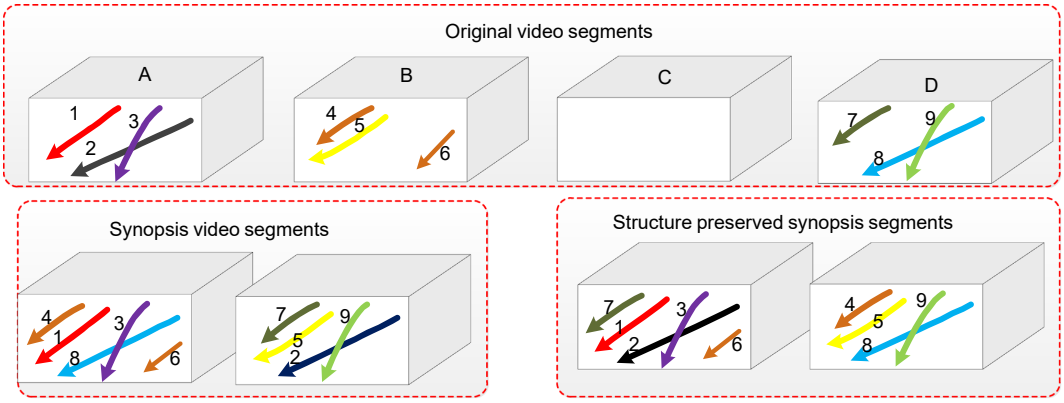


Figure 1-6. Concept of the structure preserved video synopsis [5]. Objects have strong interactions in the original videos are also grouped together in the synopsis video.

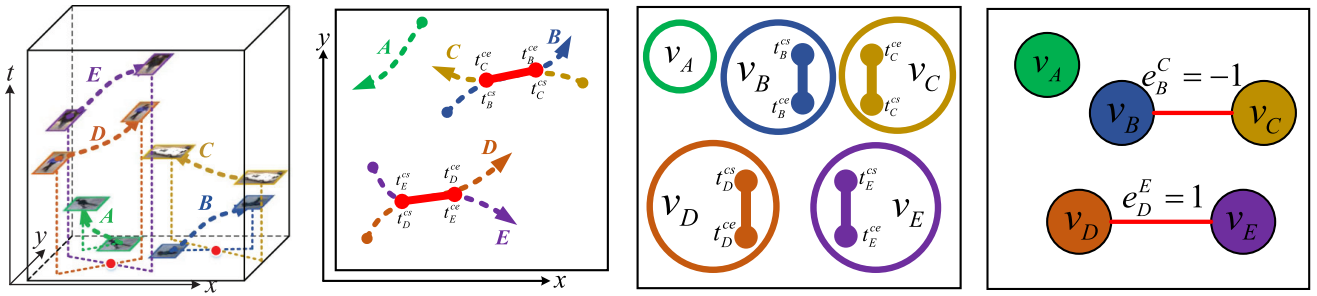
mization to select a location of the object tube. Because a solution of the greedy optimization is always local optimum, its result after few iterations may be far from the global optimum solution. To reduce the gap between them, roulette wheel selection [23] has been adopted, where it selects starting labels of the objects based on their probabilities. In consequence, the object has a chance to be rearranged in a better location than the deterministic approach. This roulette wheel selection can be applied to any online video synopsis which suffers from local optimum solutions.

Fu *et al.* [5] try to keep interactions between the objects from being broken in the condensed video generated by the online framework. To achieve the objective, they consider motion proximity and interaction of objects. Simply say, the objects which are close each other and have similar motions are more likely to have low pairwise energies. The conceptual diagram of the structure preserved synopsis

is illustrated in Figure 1-6. Apart from the importance of preserving the motion structure between the objects, it adds computational burden for calculating pairwise energy terms. Even though they utilize a hierarchical optimization which resembles the clustered synopsis [17] to speed up the tube rearrangements and the online video synopsis frameworks has fewer energy terms to consider than the offline framework, such drawback is not desirable to improve the throughput or latency of the system.

He *et al.* [6] construct a potential collision graph (PCG) to represent collision relationships between the objects. Each pair of objects belongs to one of two relationships: collision free (CF) and collision potential (CP). CP is further divided into two cases: colliding in the same direction (CSD) and colliding in the opposite direction (COD). Example of the potential collision graph construction is illustrated in Figure 1-7. After the graph construction, the smallest starting label satisfying simple constraints induced by the relationships is selected as a rearranged starting label of the object tube. Then, each starting label of the object tube is determined one by one through iterations. This tube rearrangement process can be computed very fast; however, the constraints cannot handle complex relationships between the objects (e.g., objects collided two or more times or objects have both CSD and COD relationships). Also, PCG cannot encode how severe the collision between the object is. Such drawbacks could make the algorithm select suboptimal solutions during the tube rearrangement task.

In the following study of He *et al.* [21], the PCG construction has been improved to consider one more case of the relationship, an intersection of object tubes. Then,



(a) (b) (c) (d)

Figure 1-7. Example of constructing PCG [6]. Since the object tube *A* has no potential collisions, the corresponding vertex has no connected edges to other vertices. Otherwise, object tubes having collision potentials (*B*-*C* and *D*-*E*) are connected to each others, and the edges have either positive (CSD relationship) or negative (COD relationship) values.

they solve the tube rearrangement problem by applying $L(q)-coloring$ on the graph. Aside from the improvements, since the PCG only contains too abstract information of the collisions (start and end time of the collision, and time of intersection between the objects), its ability of avoiding collisions relies on the value of the hyper-parameter q.

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 algorithms are described in Chapter 3. Chapter 4 contains explanations of other components that the online video synopsis framework 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 [7, 12, 13] is described to show which part of the formulation has to be changed to reduce the computational complexity of the tube rearrangement task. In addition, the reason why online video synopsis mainly considers the 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 the temporal domain. In other words, we try to find the best combination of object tubes' starting positions in the 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.* [13], 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 [13] 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

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

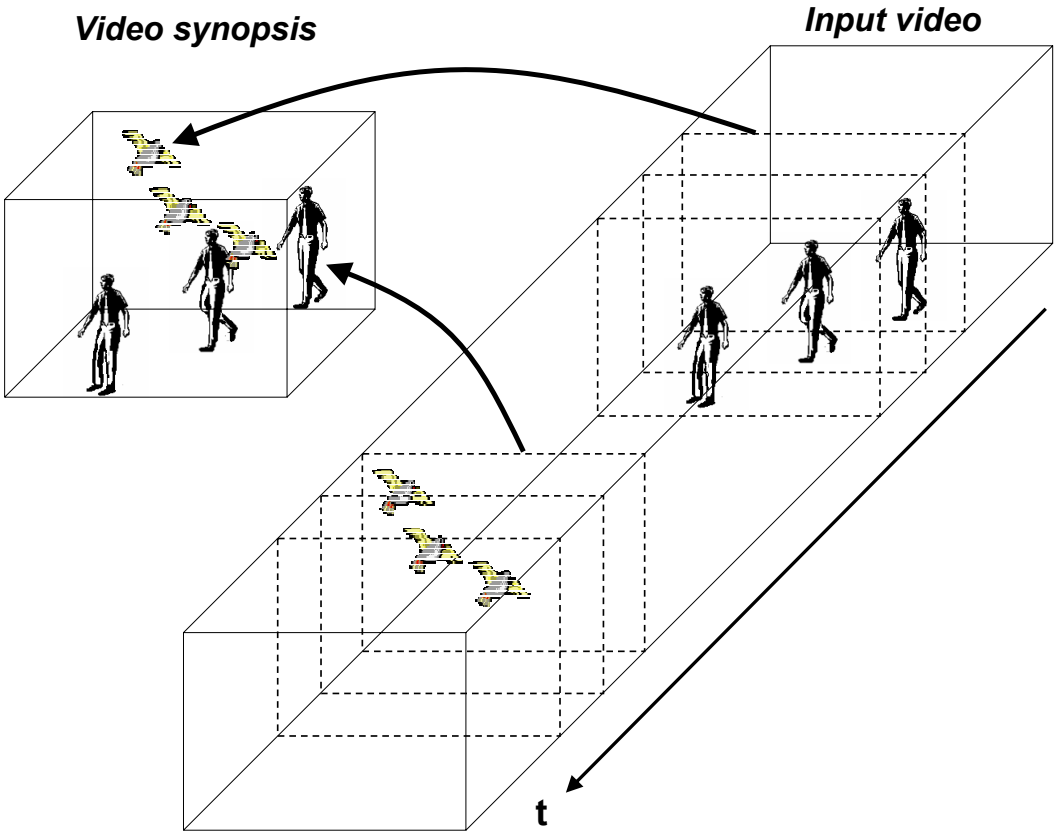


Figure 2-1. Concept diagram of video synopsis [7]. The bird and man appeared at different time in the original video are rearranged in the temporal domain, and then displayed simultaneously in the condensed video.

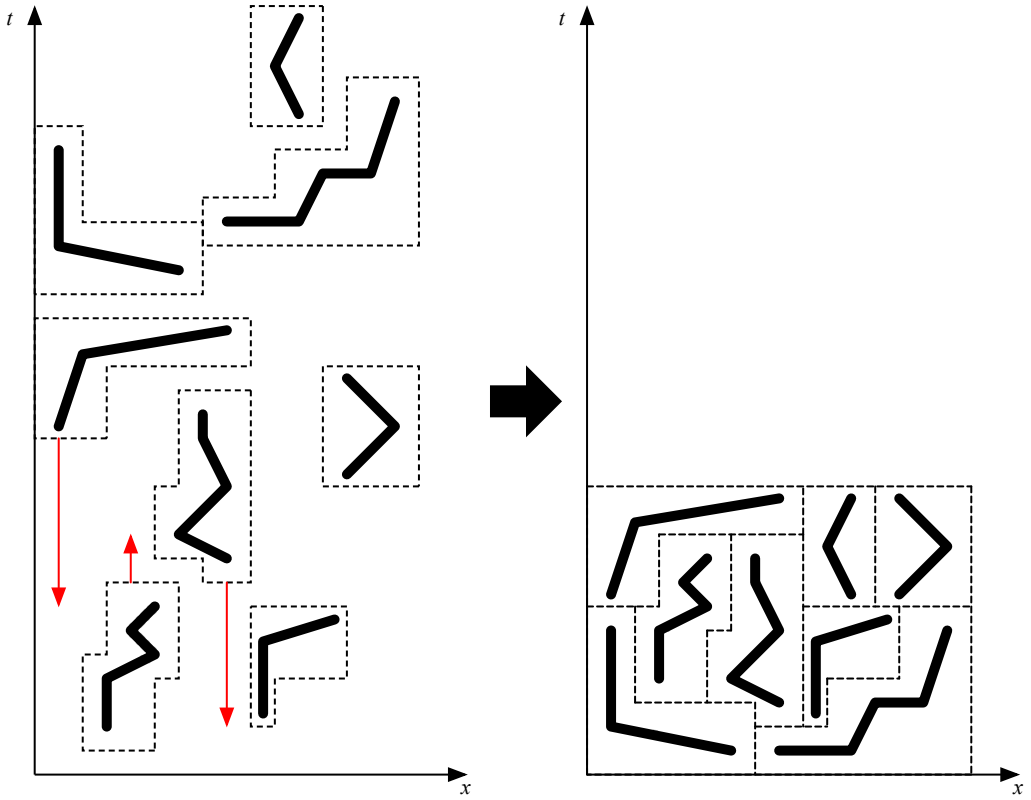


Figure 2-2. Example of the object tube rearrangement in 2D space. Red arrows indicate some offsets of the starting labels for better understanding of the tube rearrangement process. We can see that after the tube rearrangement, the length of the condensed video becomes much shorter than that of the original.

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 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 works of video synopsis [7, 12, 13], $\chi_i(x,y,t)$ is defined as

$$\chi_i(x,y,t) = \begin{cases} \left\| I_i(x,y,t) - B(x,y,t) \right\|^2 & t \in t_i \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

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 to 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 720^{th} 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 [13]. To generate background images, background modeling methods such as well known Gaussian Mixture Models (GMM) [24-26], or simple temporal median of input images over several frames can be used. Some examples of the time-lapse background are depicted in Figure 2-3.

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 [8]) can be used to smoothly blend the foreground and background pixels. However, inaccurate foreground segmentation results or

(a) (b)

(c) (d)

(e) (f)

Figure 2-3. Some time-lapse background images generated from the input video captured from 7 pm to 8 pm.

foreground and background pixels from different time of day can cause visually unappealing results. E_s is defined to penalize such situation.

$$E_s(l_i) = \sum_{x,y \in \sigma_i, t} \left\| I_i(x,y,t) - B_t(x,y,t) \right\|^2, \tag{2.4}$$

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 the morphological dilation to the foreground mask of the i^{th} object and subtract it from the original. Based on (2.4), the objects appeared in the midnight are more likely to be displayed 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 the temporal domain are less likely to have interactions.

In summary, (2.5) and (2.6) encode the idea that objects close in the 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 the 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 [5, 6, 21] maintain a queue of object tubes and the queue grows as a new object tube is extracted in the input video. When the size of the queue exceeds a certain threshold K, the 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.

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

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

where $B_i(t)$ and $B_j(t)$ are bounding boxes of i^{th} and j^{th} objects at frame t, respec-

tively. 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 consistencies only require 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 [5, 6, 18, 20, 21] do not calculate E_c efficiently. 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. In addition, the occupation matrix allows us to reduce the computational complexity of E_c by using FFT [22].

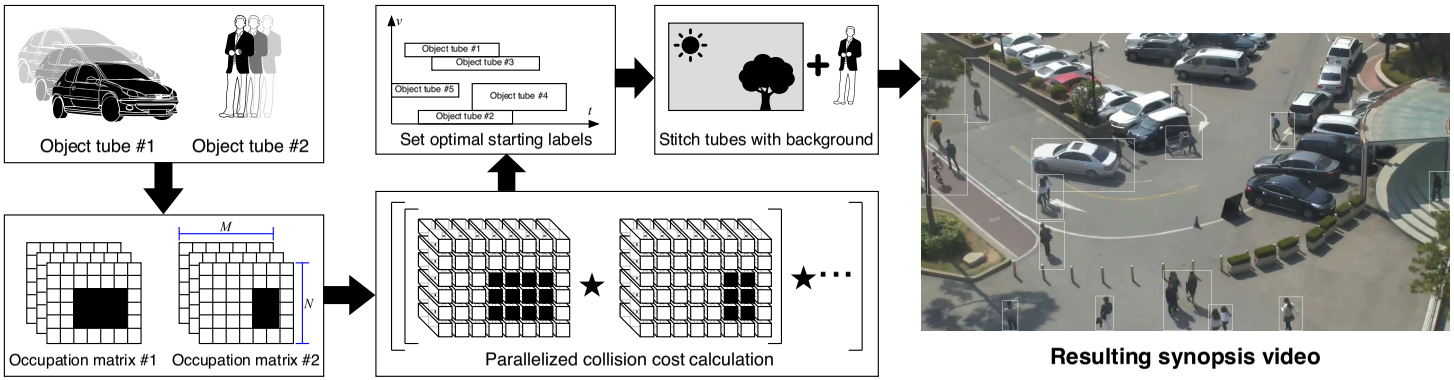


Figure 2-4. Flowchart of the proposed online video synopsis framework. At the beginning, foreground of the object tube is reshaped into the 3D occupation matrix. This matrix representation is used to calculate the collision energy fast in conjunction with FFT [22] and parallel processing. Afterwards, optimum starting labels of the tubes are determined, and the rearranged tubes are then stitched back 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 2-4.

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 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 values 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

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

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

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

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

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 generation 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.

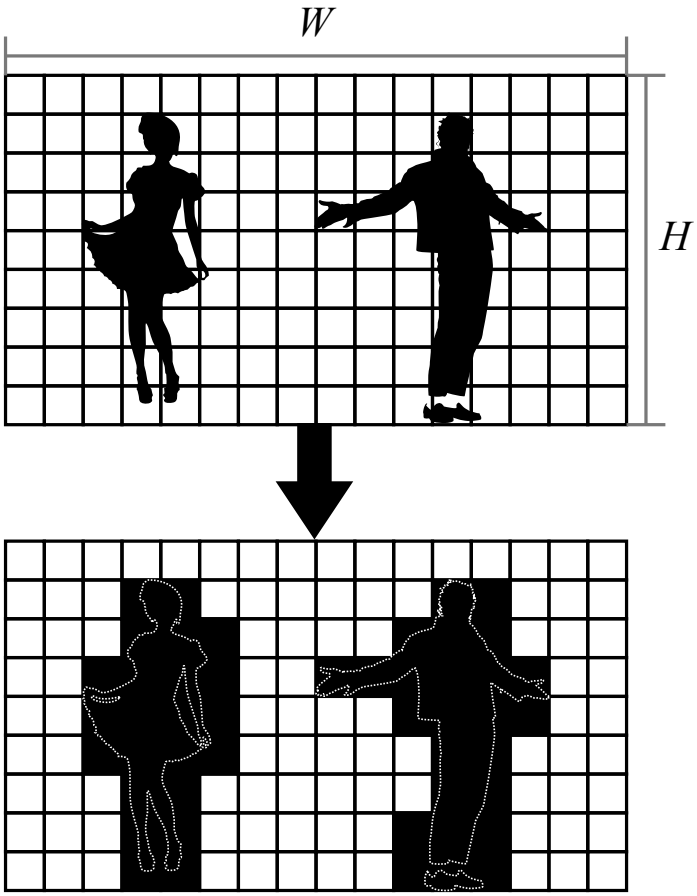


Figure 3-1. Example of the binary occupation matrix generation when $\mathcal{M} \times \mathcal{N} = 9 \times 16$. The foreground and background of the object are represented in black and white, respectively. Dotted lines in the figure are depicted to show contours of the original object for the readers.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

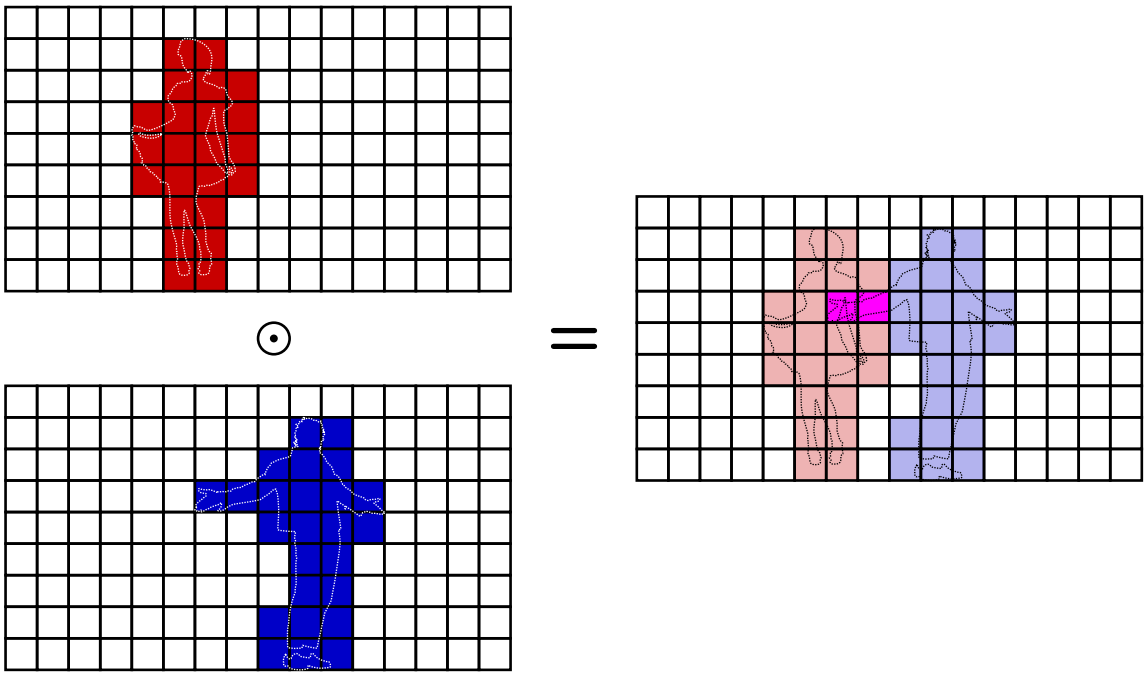


Figure 3-2. Example calculation of reformulated collision energy E_c with two binary occupation matrices. Occupied elements in the matrix are colored in red and blue. After the element-wise multiplication, we can see that the objects have two collided elements colored in magenta. In this figure, ⊙ is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

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 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 [7, 12, 13], 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 [5,6,20], 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 the 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 optimiza-

tion 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

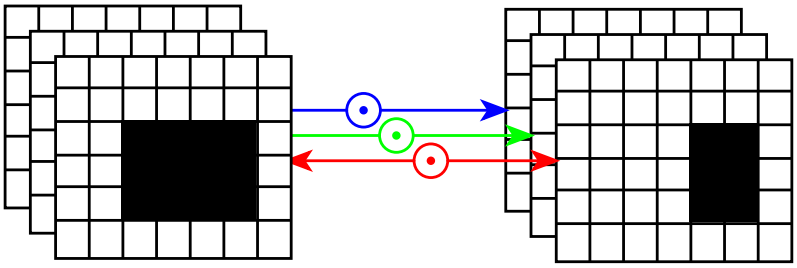
Even though $\mathbf{M}$ provides an efficient way of representing the object tubes and E_c can be computed easily with the element-wise multiplication, optimization of E_c can further be accelerated by using both parallel processing and cross-correlation of two occupation matrices in the temporal domain. 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

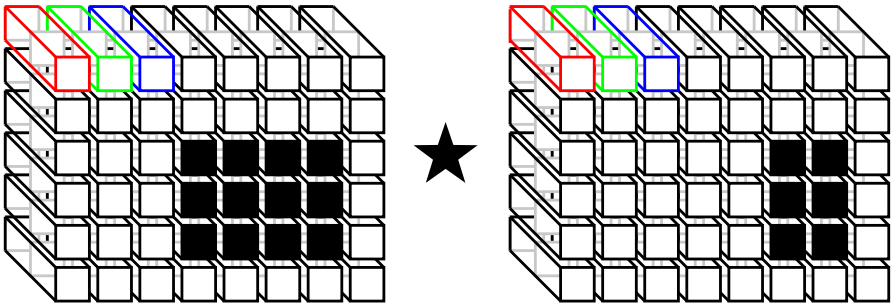
$$\begin{aligned} E_c(l_i, L_{i-1}) &= \sum_{u=1}^{\mathcal{M}} \sum_{v=1}^{\mathcal{N}} \mathbf{M}_i \star \mathbf{M}_{i-1}^*(u, v, l_i - l_{i-1}^*) \\ &= \sum_{u=1}^{\mathcal{M}} \sum_{v=1}^{\mathcal{N}} \sum_{t=-\infty}^{\infty} \mathbf{M}_i \mathbf{M}_{i-1}^*(u, v, t + l_i - l_{i-1}^*), \end{aligned}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

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

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 the 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) [22] 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 $6\times8\times3$ 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 some correspondences.

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

4 Online video synopsis framework

The proposed tube rearrangement algorithm is based on the online framework. Similar to existing online frameworks [5,6,20], the proposed framework consists of four stages: foreground segmentation, object tube generation, tube rearrangement, and object stitching. Among them, three components, except for the tube rearrangement, will be explained in detail.

4.1 Foreground segmentation

Since this field of research has been studied for decades, there are numerous choices for extracting foreground from the background. Very limited list of the methods are GMM [24–26], ViBe variants [27–30], SOBS [31,32], non-parametric background modelings [33, 34], deep CNN based approaches [35, 36], and GAN based approaches [37–39].

To select the best foreground segmentation algorithm for the proposed framework, six publicly available implementations are plugged into the framework and compared regarding segmentation quality, background quality, and computation time:

GMM [25], ViBe [27-30], CNT [40], GSOC [41], PAWCS [42], and LSBP [43]. Brief introductions to the algorithms are given as follows.

GMM of Zivkovic [25] is named as MOG2 in the OpenCV library and its computation can be accelerated by using GPU. According to benchmarks for CDnet 2014 [44], performance of GMM is hard to compete with recently published algorithms. However, its advantage comes from the low computational complexity. ViBe [27-30] is a renowned non-parametric background subtraction algorithm and has been improved over 5 years. CNT [40] is designed to subtract background fast in low spec hardware. GSOC [41] is the background subtraction algorithm developed during OpenCV Google Summer of Code and named after it. PAWCS [42] is one of the top five unsupervised background subtraction algorithms achieving decent performance in CDnet 2014 [44].

The foreground segmentation result is generated by subtracting background from the input image and applying a morphological close operation with the 3×3 square structuring element and finding connected components. Then, connected components with less than 0.1% or greater than 25% of the input image area are removed from the scene. Computation time (CT) of the algorithm is defined as a time consumption for extracting object tubes from the 1 hour long 640×360 video. Result of the different foreground segmentation algorithms is summarized in Table 4-1.

In the proposed framework, GMM [25], GSOC [41], and LSBP [43] can produce sufficiently high quality foreground segmentation results. This result is not matched with the benchmark [44] because the proposed framework utilizes a few post-processing steps to prune noisy segmentation results. For background im-

ages, CNT [40] and PAWCS [42] generate grayscale or low quality backgrounds, which are not suitable for the video synopsis application. These background subtraction methods require an additional background modeling step to generate high quality time-lapse background images. LSBP [43] also produces an awkward background image, but it can be improved with parameter adjustments. Regarding CT, CNT [40], GMM [25], and ViBe [27–30] have advantages over other algorithms. They can process the 1 hour long video within 6 minutes; in other words, extracting object tubes from the input video only takes 1/10 of the video length. By considering all factors, GMM [25] is the most preferred background subtraction method for the proposed framework. It can produce decent quality of background images as well as foreground segmentation results, and have a low computational complexity, which leads to a high throughput of the online video synopsis framework.

4.2 Object tube generation

After the foreground segmentation 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 the 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 [45–47]. The original version of the algorithm requires a condition

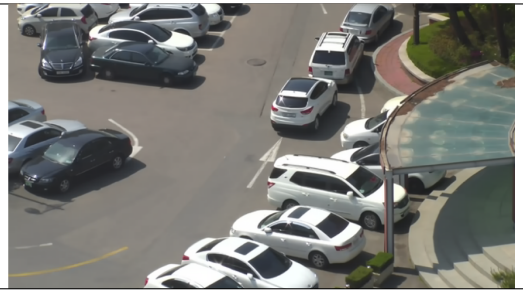
Methods	Foreground	Background	CT (min)
GMM [25]			4.23
ViBe			5.46

Table 4-1. Result of different background subtraction algorithms.

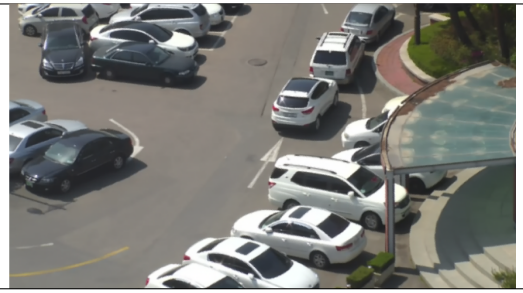
Methods	Foreground	Background	CT (min)
GSOC			20.69
CNT			3.99

Table 4-1. (continued) Result of different background subtraction algorithms.

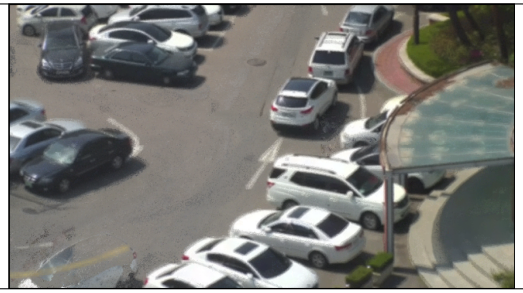
Methods	Foreground	Background	CT
PAWCS			322.01
LSBP			36.77

Table 4-1. (continued) Result of different background subtraction algorithms.

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 [48] 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 [49].

For online video synopsis, generated object tubes are stored and maintained in a queue. When the size of the 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.

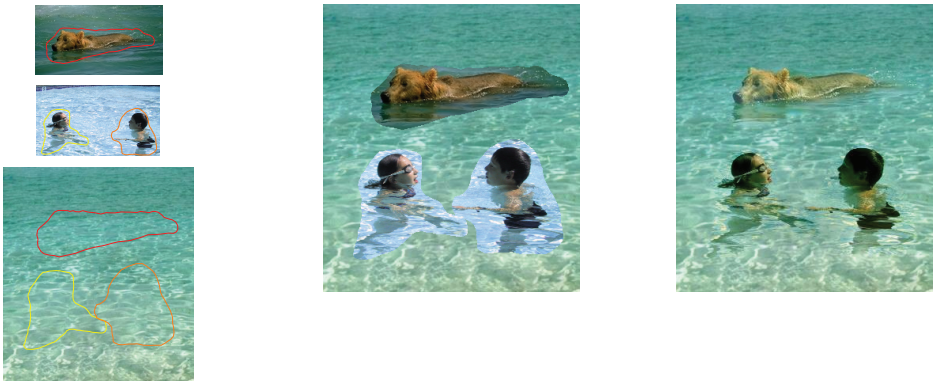
4.3 Object stitching

To make a condensed video, foregrounds of the rearranged object tubes are stitched with the background images by utilizing Poisson image editing [8]. 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 4-1 and Figure 4-2.

Before explaining the mathematics behind this editing, some notations need to be defined first. In Figure 4-3, $\boldsymbol{S}$ is the 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

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

Figure 4-1. First example of seamless cloning using Poisson image editing. All images in this figure are from the work of Pérez *et al.* [8].



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

Figure 4-2. 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.* [8].

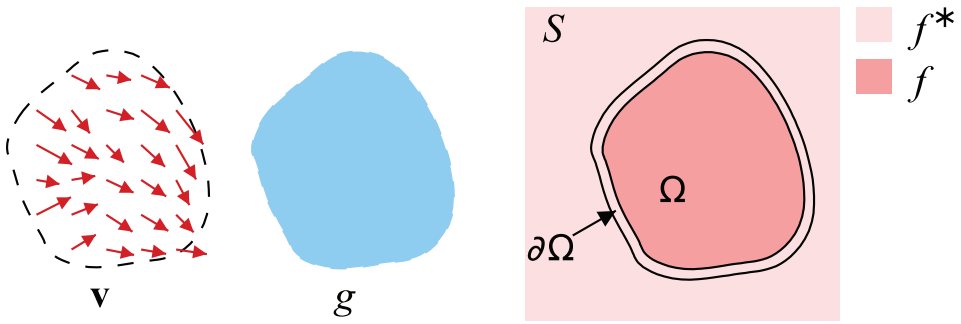


Figure 4-3. Notations used in Poisson image editing [8].

$S-(\mathbf{\Omega}-\partial\mathbf{\Omega})$ and f is defined over $\mathbf{\Omega}$; therefore, $\partial\mathbf{\Omega}$ indicates an overlapped region between $\boldsymbol{S}$ and $\mathbf{\Omega}$.

As you can see in Figure 4-3, the objective of the editing is to find a proper f satisfying the boundary condition on $\mathbf{\Omega}$. One example of achieving the objective is minimizing the 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.

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

where div· is a divergence operator; hence $\mathrm{div}\mathbf{v}=\left(\frac{\partial u}{\partial x},\frac{\partial v}{\partial y}\right)$, 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 4-4 shows an example 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, the result using Poisson image editing is more visually natural but it takes more computation time than the naïve approach.

4.4 Discontinuity of motion flow

As shown in Figure 4-5, 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 [5]. Figure 4-6 shows diagrams to

(a) Alpha blending

(b) Poisson image editing

Figure 4-4. Result of two different object stitching algorithms. The blending ratio of the foreground and background in the alpha blending is 1:1.

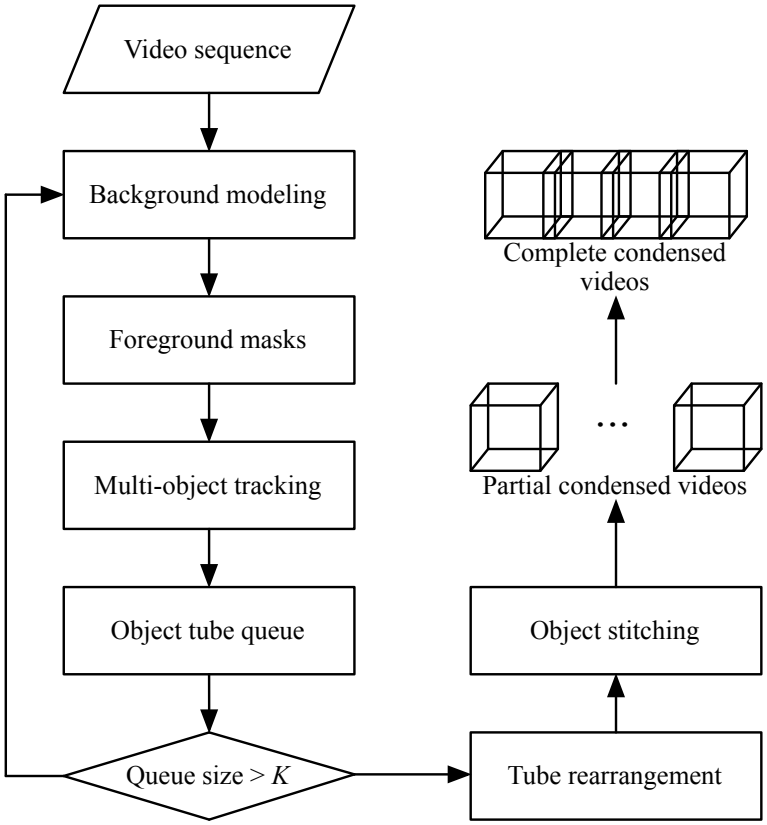
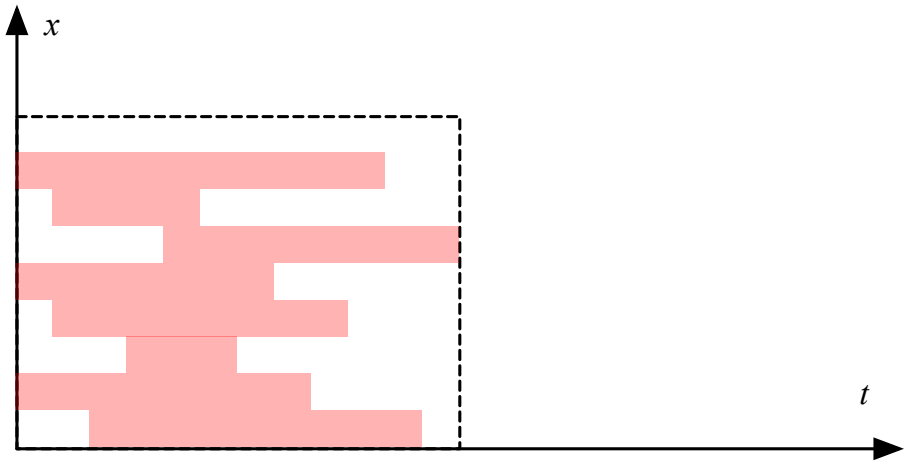
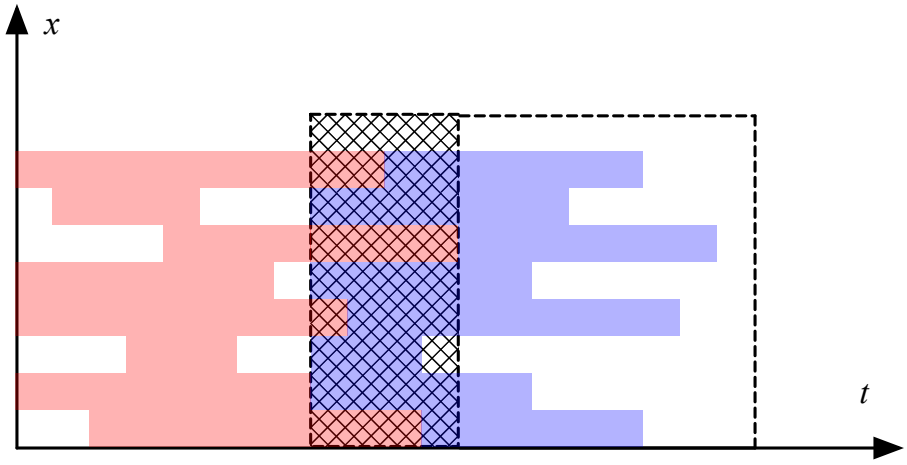


Figure 4-5. Proposed online video synopsis framework. The framework generates a partial condensed video whenever the size of the queue exceeds K. Then, partial videos are merged into the complete synopsis video.

explain the solution.



(a) Rearranged tubes after 1st iteration



(b) Rearranged tubes after 2nd iteration

Figure 4-6. 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 length.

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

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

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

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 i7-5700K 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. In addition, condensed videos of the test sequences generated by the proposed framework are summarized in Apppendix A.

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 the 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 most of the moving objects are vehicles not pedestrians.

Crossroad I and II sequences are captured at the same place with different seasons and camera's zoom parameters. 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 the 2nd floor of the building. There is a gateway of the library at top of the scene

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.

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.

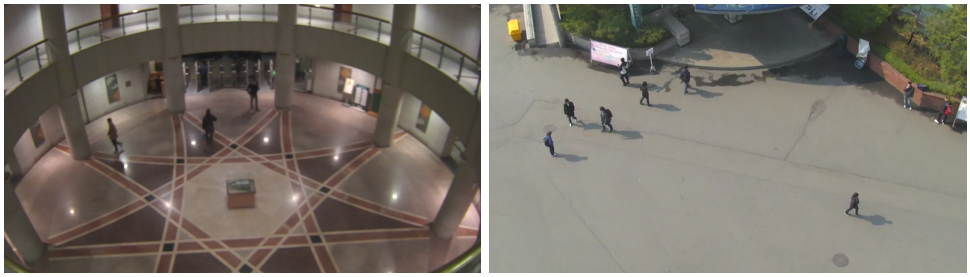
Subway station plaza is an open place in front of the subway station entrance. Since there are many ways to get to the buildings from here, there is no dominant walking direction of people.

5.3 Performance analysis

The experiments in this section are designed to 1) analyze parameters required 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.

(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
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 select appropriate values of the parameters required for the proposed tube rearrangement algorithm.

5.3.1 Parameter analysis

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 appropriate values of parameters and detail characteristics of the videos are summarized in Table 5-2.

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 does. 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 trivial 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

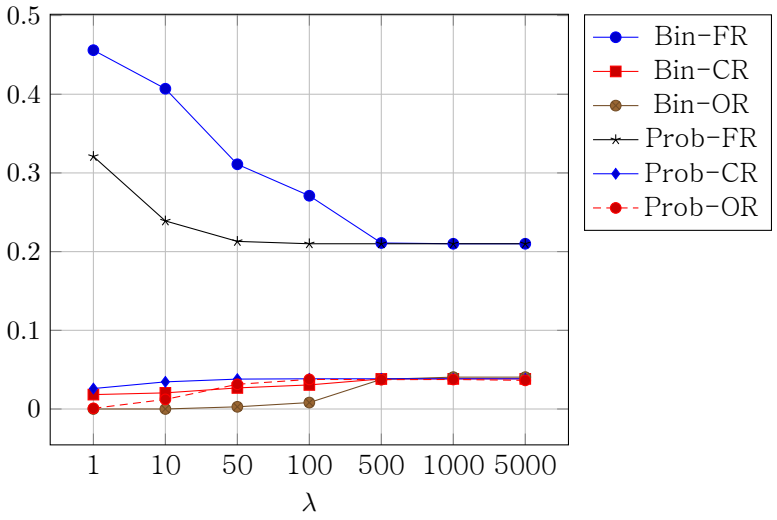
length energy. This solution is not desirable neither; therefore, we need to choose a balanced value of λ.

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 observations, it is better to select different λ for different types of occupation matrices. However, for easier analysis of parameters in following sections, both types of the occupation matrix utilize the same λ value of 100.

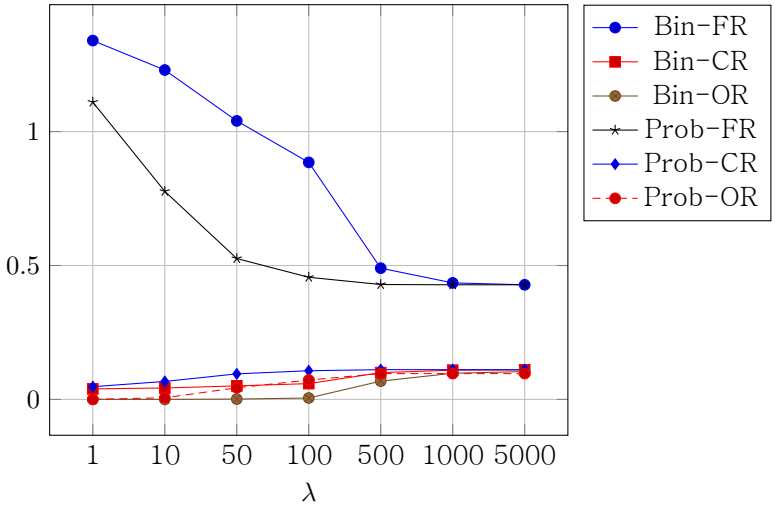
Size of occupation matrix

Second experiment is conducted by changing spatial resolution $\mathcal{M} \times \mathcal{N}$ of the occupation matrix. Assume that the aspect ratio (the ratio of the width to the height) of the input video is 16:9. Under this 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

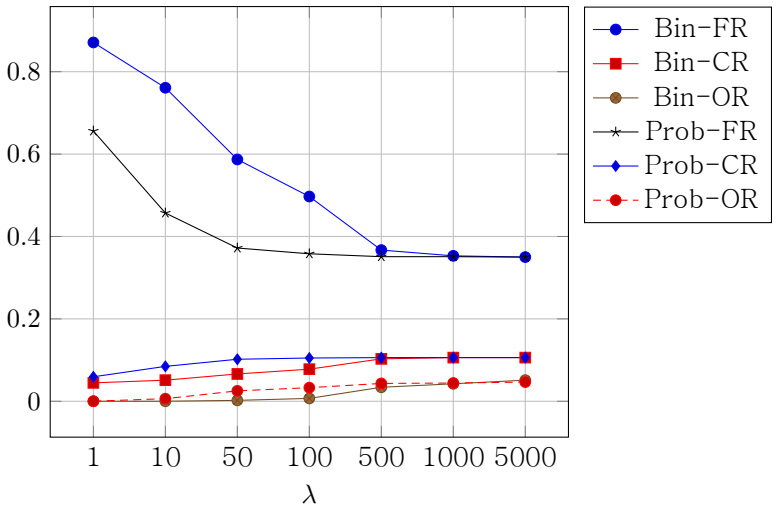


(a) Crossroad III

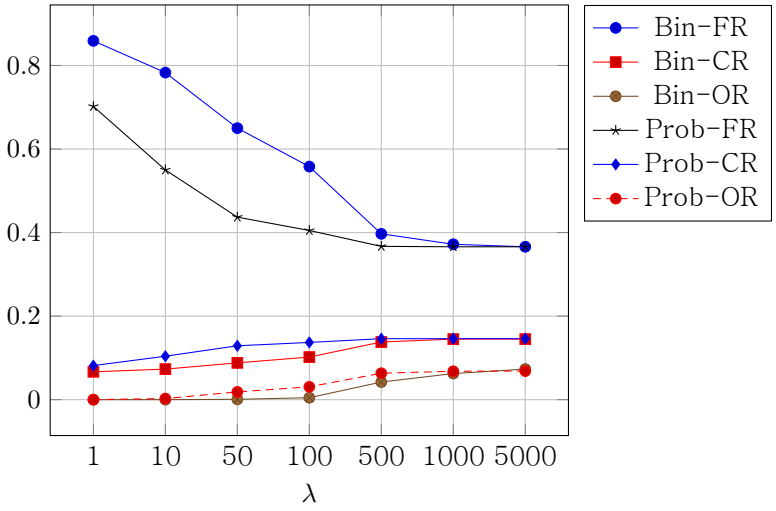


(b) Library lobby II

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

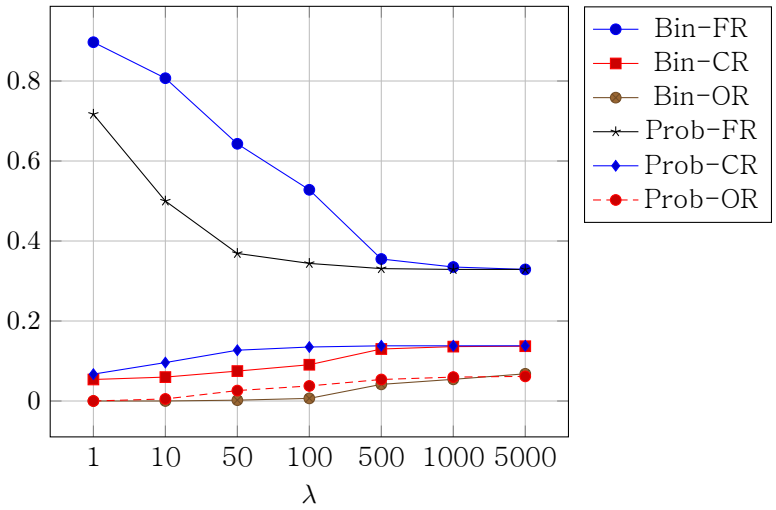


(c) Subway station plaza II



(d) Subway station plaza III

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

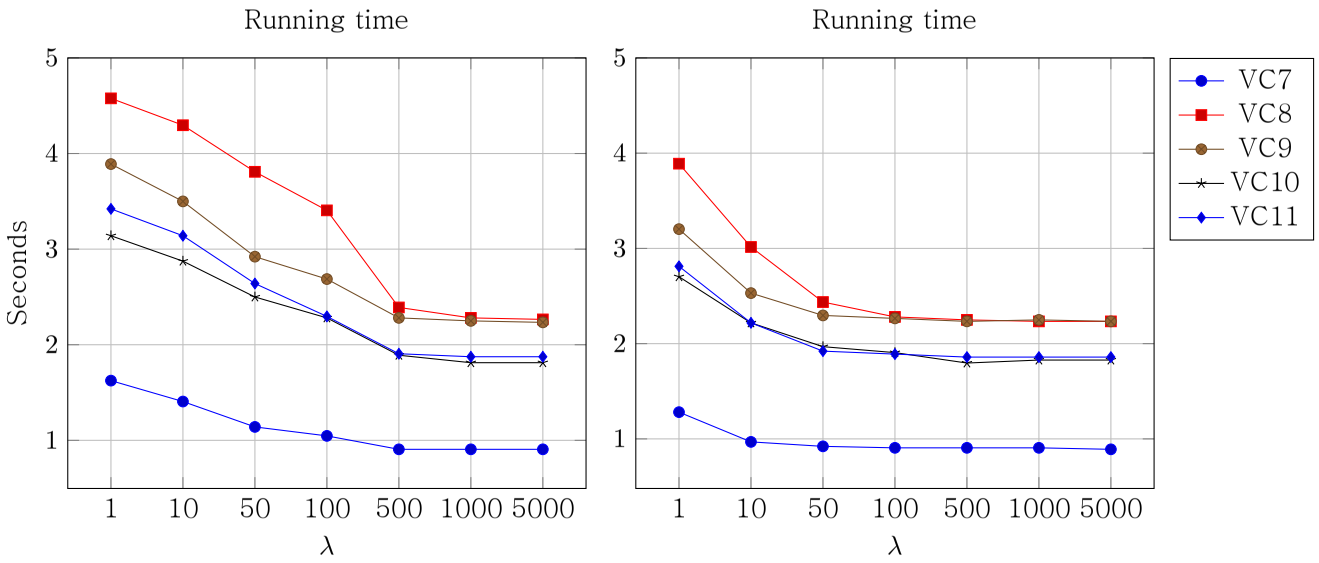


(e) Subway station plaza IV

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

collisions. Then, the resulting video has lower FR and higher CR than the one using the smaller occupation 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 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, the probabilistic occupation matrix allows us to compute rearranged starting labels faster than the binary counterpart; however, the tendency of the computation time for different $\mathcal{M} \times \mathcal{N}$ is almost identical. Especially, the



(a) Binary occupation matrix

(b) Probabilistic occupation matrix

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

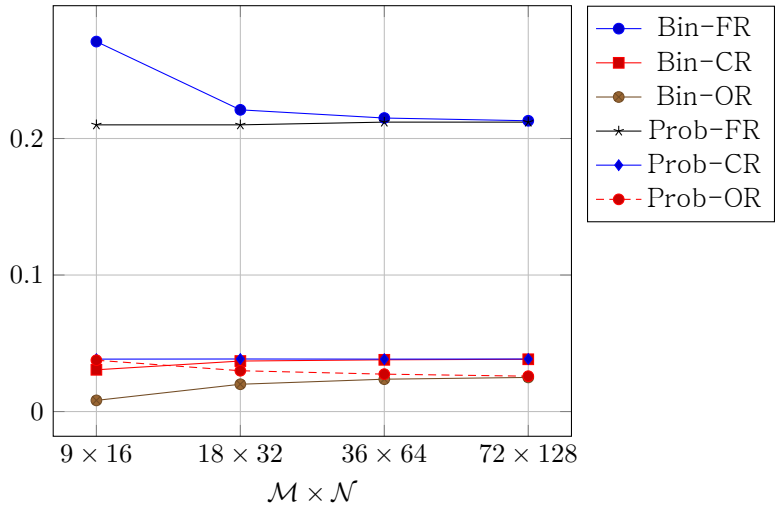
computation time for both types of the occupation matrix is reduced by at least 1/40, when the width and height of the matrix are scaled by 1/8 (from 72×128 to 9×16).

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.

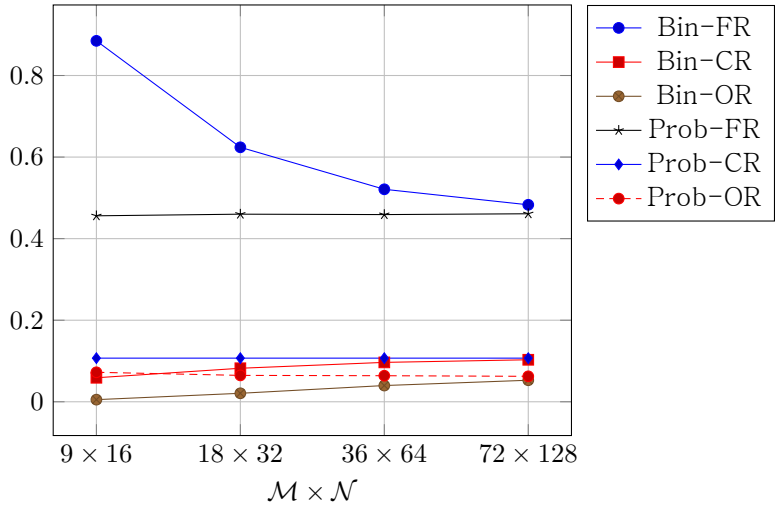
Size of queue

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 Section 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 along with increasing K, FR is decreased by at least 0.2 and CR is increased slightly, while OR remains almost unchanged. On the other hand, the algorithm using the probabilistic matrix produces shorter synopsis videos with higher CR and OR. The key difference between the

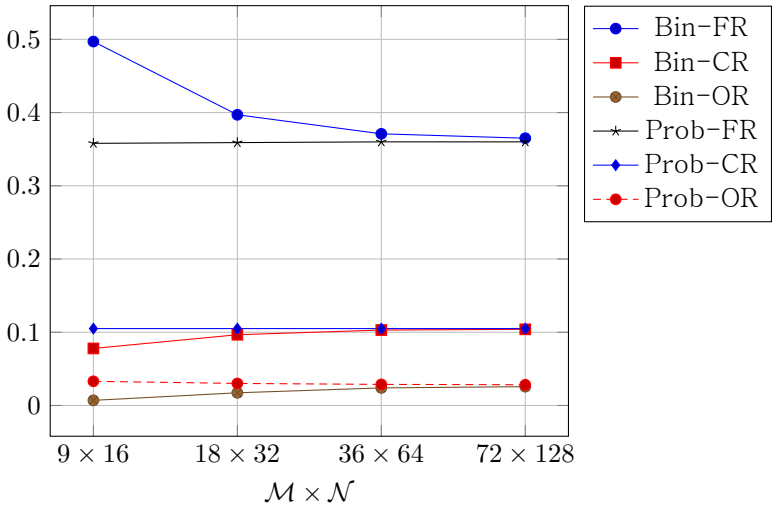


(a) Crossroad III

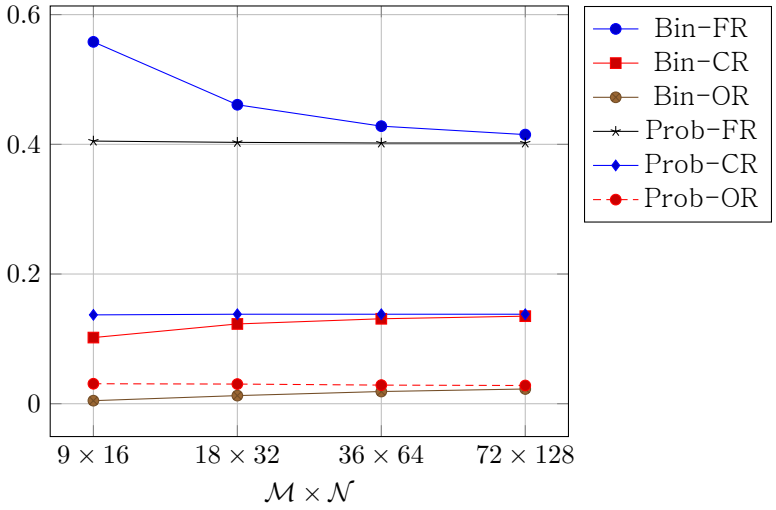


(b) Library lobby II

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



(c) Subway station plaza II

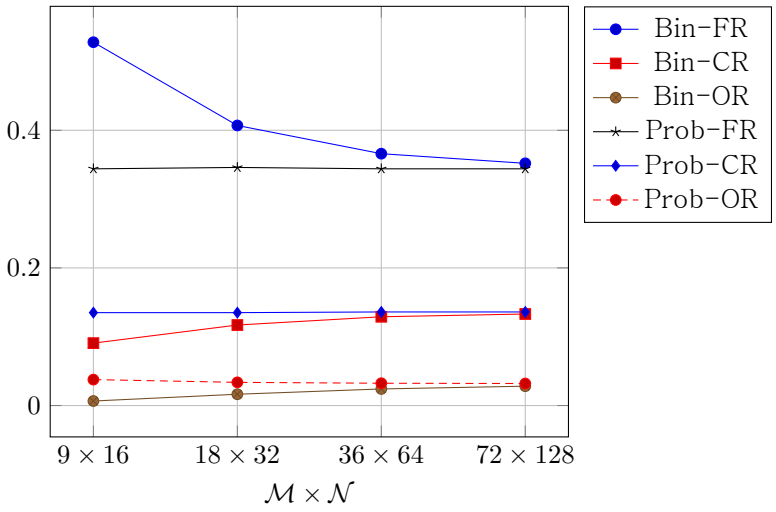


(d) Subway station plaza III

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.



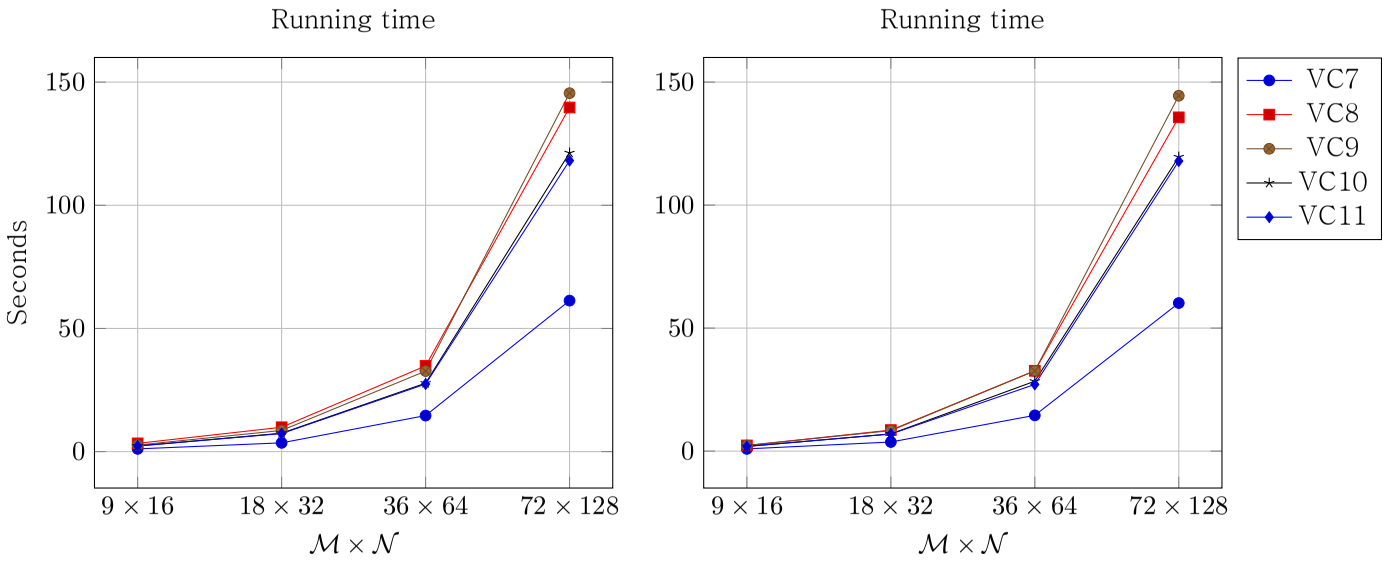
(e) Subway station plaza IV

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

binary and probabilistic occupation matrices is that the binary matrix allows us to produce synopsis videos having constant OR regardless of K. In other words, the resulting synopsis video never becomes more crowded even if K is large.

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



(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.

selecting K. Empirically, for both binary and probabilistic matrices, a median of the candidate values, 50, is used in the subsequent experiments.

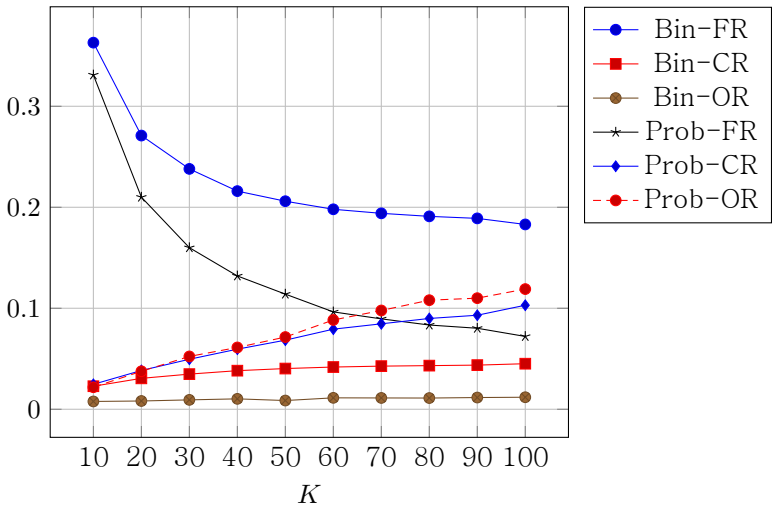
Type of occupation matrix

Based on the observations so far, the one and only advantage of the binary occupation matrix over the probabilistic occupation matrix is OR. Therefore, we can say that using the probabilistic occupation matrix is beneficial for most of the 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.

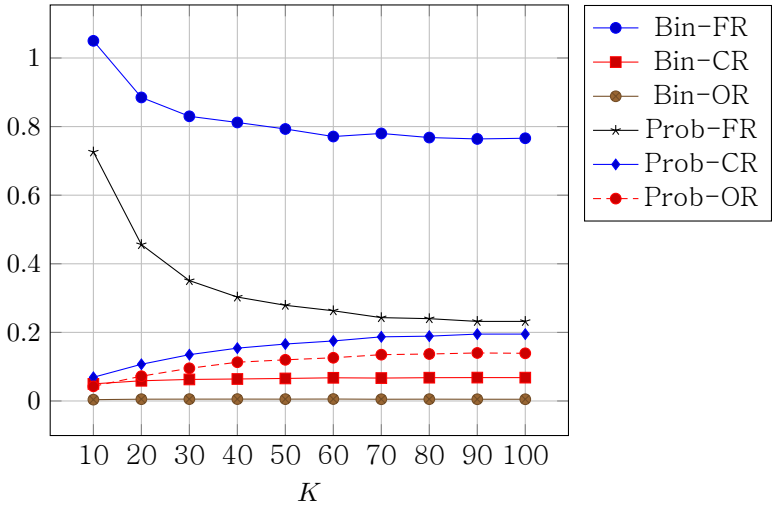
5.3.2 Ablation study for speed up techniques

In this section, ablation study for the two speed up techniques used in the proposed algorithm is conducted: FFT [22] 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 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

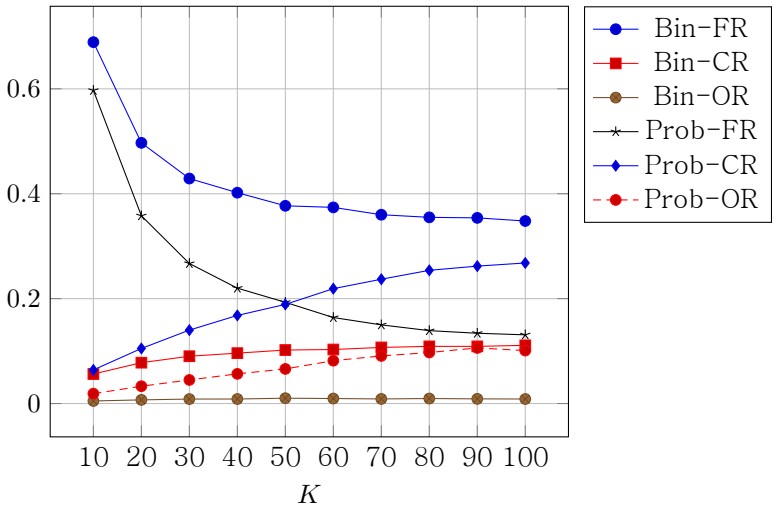


(a) Crossroad III

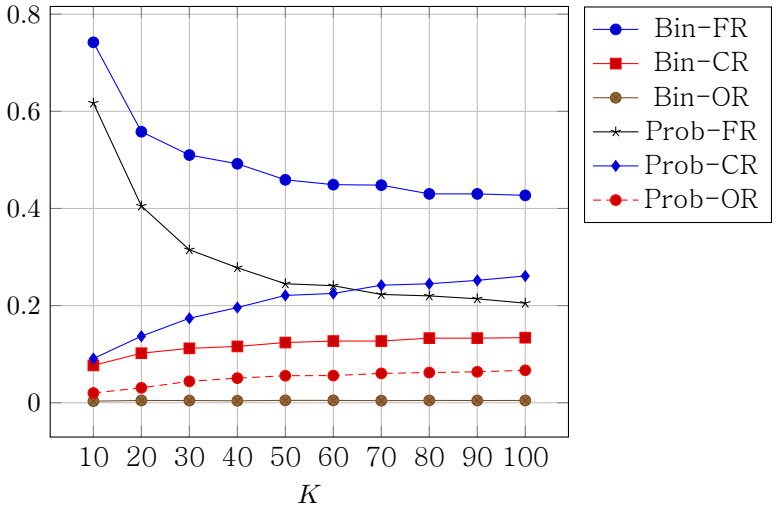


(b) Library lobby II

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

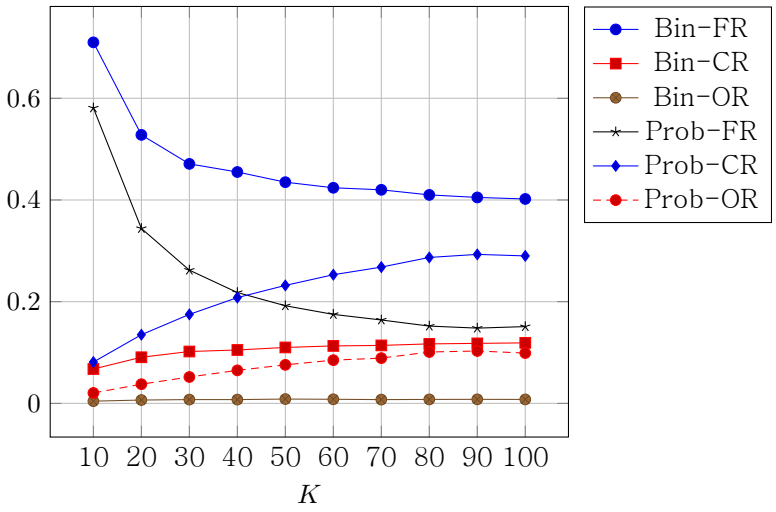


(c) Subway station plaza II



(d) Subway station plaza III

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

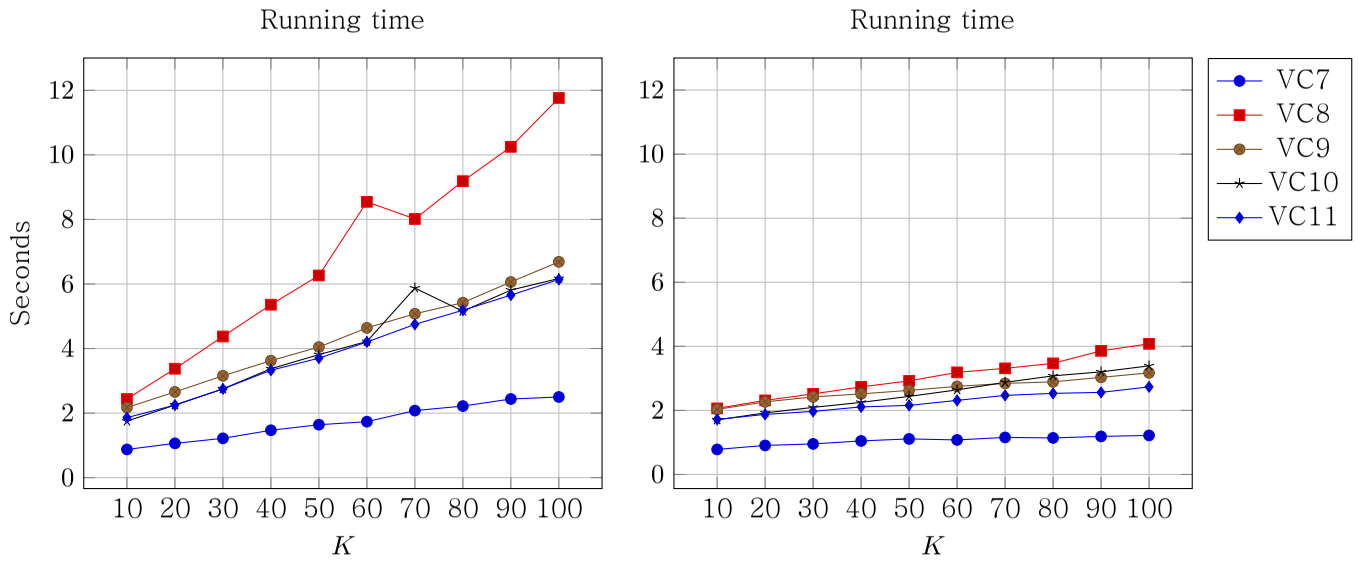


(e) Subway station plaza IV

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

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

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



(a) Binary occupation matrix

(b) Probabilistic occupation matrix

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

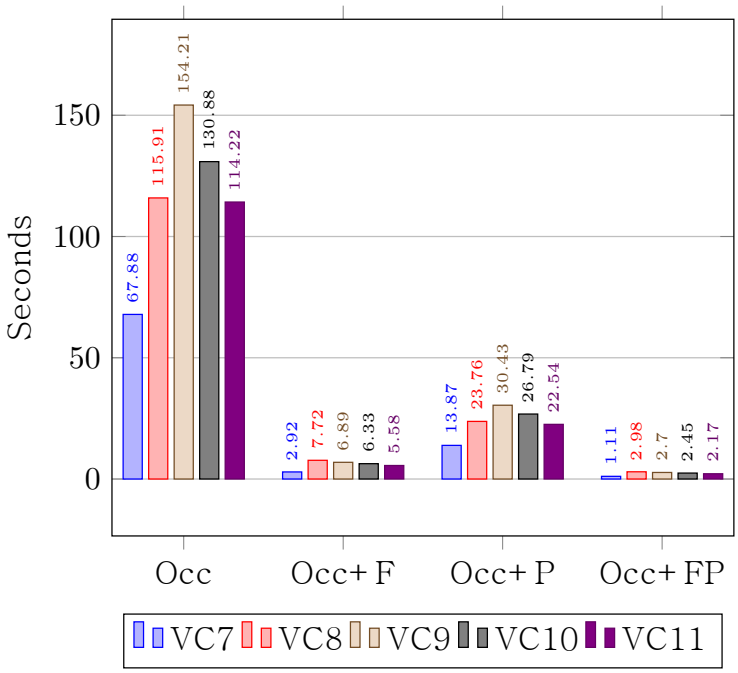


Figure 5-8. Result of the ablation study to compare four different versions of the proposed algorithm.

of the 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 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 [5, 6, 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 [20] 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 maintaining moving objects. Note that existing algorithms have not employed any spatial subsampling.

Regarding RT, as shown in Figure 5-9, the proposed algorithm with FFT (Occ+ F) runs at least 2.26 times faster than other algorithms. This performance gap can be increased by applying parallel processing to the proposed algorithm. In this case, RT of the proposed algorithm is less than 1/6 of comparing algorithms' RT. For the

algorithm of Fu *et al.* [5], due to the extra computation of the motion proximity and interaction, it takes much more time in computing rearranged labels even though it has benefits from the hierarchical optimization. For other two algorithms, we can see that the key concepts of the optimization (Tetris-like tube filling [20] and PCG [6]) definitely contributes to the performance, but their differences are 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.* [20]. 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.* [20] 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 has a lack of correlation for FR. According to Figure 5-12, the 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 [20] follows it. The algorithm of Fu *et al.* [5] takes the 3rd position and the PCG baesd algorithm [6] performs the worst.

Except for the algorithm considering the structured motion [5], 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 starting labels within 10 seconds at maximum for the test sequences. However, the algorithms have their own distinctive characteristics. For the Tetris like optimization [20], 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.* [6], 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 (FFT 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.

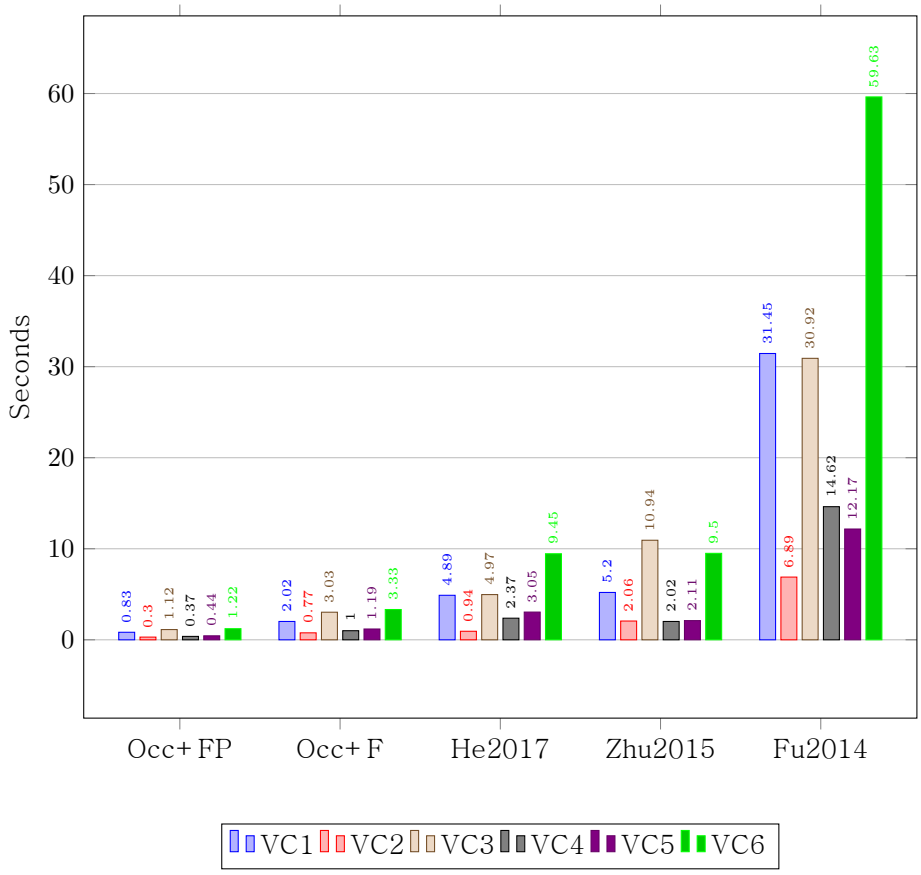


Figure 5-9. Result of the experiment regarding RT measured in seconds.

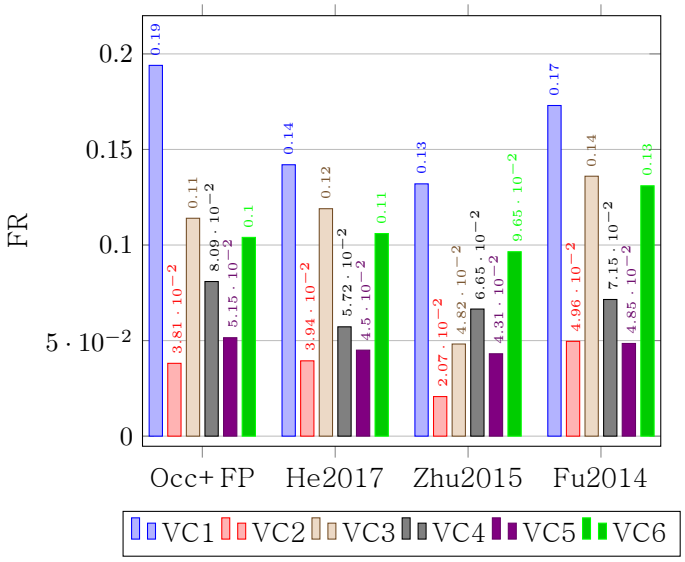


Figure 5-10. Result of the experiment regarding FR.

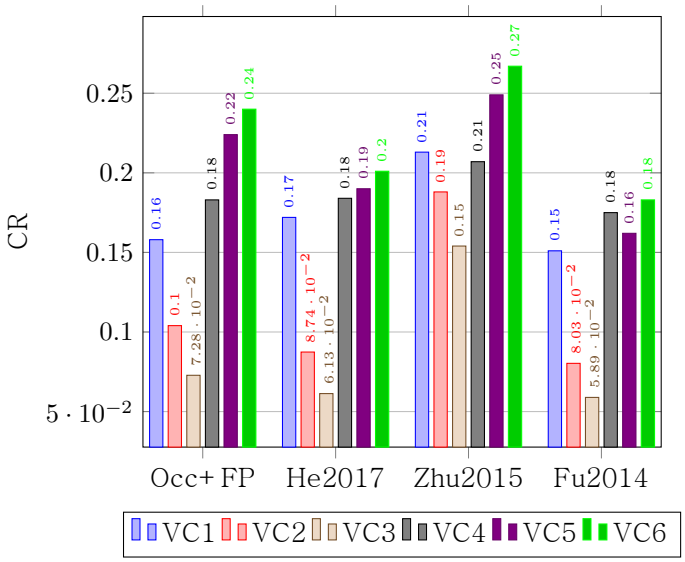


Figure 5-11. Result of the experiment regarding CR.

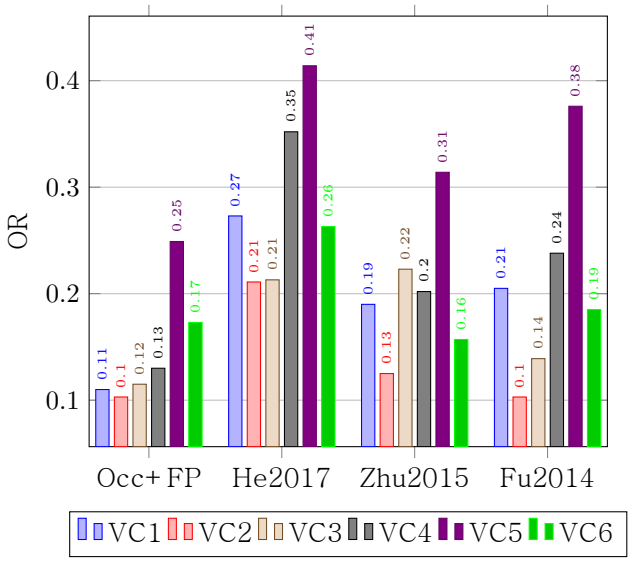


Figure 5-12. Result of the experiment regarding OR.

6 Conclusion

In this dissertation, efficient tube rearrangement algorithms for online video synopsis are proposed. Efficiency of the proposed algorithms comes from the new representation of the object tube, the occupation matrix, spatial approximation of foreground pixels. By using the occupation matrix, the new collision energy can be defined as element-wise multiplications between two occupation matrices. To find the optimum starting label of the object tube, we need to calculate collision energies for different starting labels. This process is same as conducting $\mathcal{M} \times \mathcal{N}$ 1D cross-correlations between two sets of signals. Therefore, the proposed algorithm can effectively compute the collision energy by using FFT with the low computational complexity. According to the ablation study, utilizing FFT during the collision energy computation reduces running time of the proposed algorithm by 1/20, and the further improvement can be done by using FFT with parallel processing.

For the comparative experiments, the proposed algorithm with FFT optimizes starting labels of the objects within 1.89 seconds in average for all test sequences. This is at least 2.26 faster results than state-of-art online tube rearrangement algorithms. Moreover, these differences can be widened by using FFT with parallel

processing. In this case, the proposed algorithm only takes 1/6 less time than competing algorithms. In addition, it produces condensed videos with the lowest OR, which indicates that the resulting videos are visually less complex and easier to understand behaviors of rearranged objects than the others.

For better understanding of the proposed algorithm, effect of adjusting four required parameters is extensively analyzed. When the length parameter λ is either too small or large, the proposed algorithm produces the trivial solution. To avoid the situation, the balanced value of λ should be selected. For the occupation matrix related parameters, increasing resolution of both binary and probabilistic occupation matrices has little benefits for FR and CR, but a huge disadvantage for RT; therefore, the smallest resolution of the matrix (9×16) is preferred. Also, the probabilistic occupation matrix produces better results than the binary matrix in general. As the size of queue K increases, FR and CR become better but other two metrics become worse; therefore, we need to compromise the performance while selecting K.

The future work of this dissertation will involve running the proposed algorithm on GPU and improving other components of the online video synopsis framework, especially for the object tube generation.

BIBLIOGRAPHY

[1] 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/

[2] Y. Nie, C. Xiao, H. Sun, and P. Li, “Compact video synopsis via global spatiotemporal optimization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 10, pp. 1664–1676, Oct. 2013. [Online]. Available: https://ieeexplore.ieee.org/document/6280551

[3] Y. Hoshen and S. Peleg, “Live video synopsis for multiple cameras,” in *2015 IEEE International Conference on Image Processing (ICIP)*, vol. 2015-Decem. IEEE, Sep. 2015, pp. 212–216. [Online]. Available: http://ieeexplore.ieee.org/document/7350790/

[4] J. Zhu, S. Liao, and S. Z. Li, “Multicamera Joint Video Synopsis,” *IEEE Transactions on Circuits and Systems for Video Technology*,

vol. 26, no. 6, pp. 1058–1069, Jun. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7103038/

[5] 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

[6] 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/

[7] A. Rav-Acha, Y. Pritch, and S. Peleg, “Making a Long Video Short: Dynamic Video Synopsis,” in *2006 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, Jun. 2006, pp. 435–441. [Online]. Available: http://ieeexplore.ieee.org/document/1640790/

[8] 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

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

[10] 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

[11] 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

[12] 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

[13] 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/

[14] 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/

[15] 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

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

[17] 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/

[18] S. Feng, Z. Lei, D. 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/

[19] 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/

[20] J. Zhu, S. Feng, D. Yi, S. Liao, Z. 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/

[21] Y. He, C. Gao, N. Sang, Z. Qu, and J. Han, "Graph coloring based surveillance video synopsis," *Neurocomputing*, vol. 225, pp. 64–79, Feb. 2017. [Online]. Available: https://doi.org/10.1016/j.neucom.2016.11.011

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

[23] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT press, 1998.

[24] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *1999 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, June 1999, pp. 246–252. [Online]. Available: https://ieeexplore.ieee.org/document/784637

[25] 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/

[26] 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

[27] 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

[28] ——, "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

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

[30] 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

[31] 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

[32] ——, "The SOBS algorithm: What are the limits?" in *2012 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2012, pp.

21–26. [Online]. Available: https://ieeexplore.ieee.org/document/6238922

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

[34] 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

[35] 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/

[36] 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

[37] 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 Confer-*

ence on Image Processing, Oct. 2018, pp. 4018–4022. [Online]. Available: https://ieeexplore.ieee.org/document/8451603/

[38] 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

[39] D. Sakkos, E. S. L. Ho, and H. P. H. Shum, "Illumination-Aware Multi-Task GANs for Foreground Segmentation," *IEEE Access*, vol. 7, pp. 10976–10986, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8606933/

[40] "The BackgroundSubtractorCNT project." [Online]. Available: https://github.com/sagi-z/BackgroundSubtractorCNT

[41] "OpenCV 4.1 documentation for BackgroundSubtractorGSOC Class." [Online]. Available: https://docs.opencv.org/4.1.0/d4/dd5/classcv_1_1bgsegm_1_1BackgroundSubtractorGSOC.html

[42] P. St-Charles, G. Bilodeau, and R. Bergevin, "A Self-Adjusting Approach to Change Detection Based on Background Word Consensus," in *2015 IEEE Winter Conference on Applications of Computer Vision*, Jan 2015, pp. 990–997. [Online]. Available: https://ieeexplore.ieee.org/document/7045991

[43] L. Guo, D. Xu, and Z. Qiang, "Background Subtraction Using Local SVD Binary Pattern," in *2016 IEEE Conference on Computer Vision and Pattern*

Recognition Workshops, Jun. 2016, pp. 1159–1167. [Online]. Available: http://ieeexplore.ieee.org/document/7789638/

[44] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An Expanded Change Detection Benchmark Dataset," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2014, pp. 393–400. [Online]. Available: https://ieeexplore.ieee.org/document/6910011

[45] 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

[46] ——, "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

[47] 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

[48] 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

[49] 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

Appendices

A Result of proposed framework

This appendix summarizes some frames of the condensed videos generated by the proposed framework when the original videos are the test sequences described in Section 5.2. Time stamps in the condensed video indicate objects' appearing time in the original video.

(a) (b)

(c) (d)

Figure A-1. Some frames of the condensed video of Parking lot square I sequence (VC1).

(a) (b)

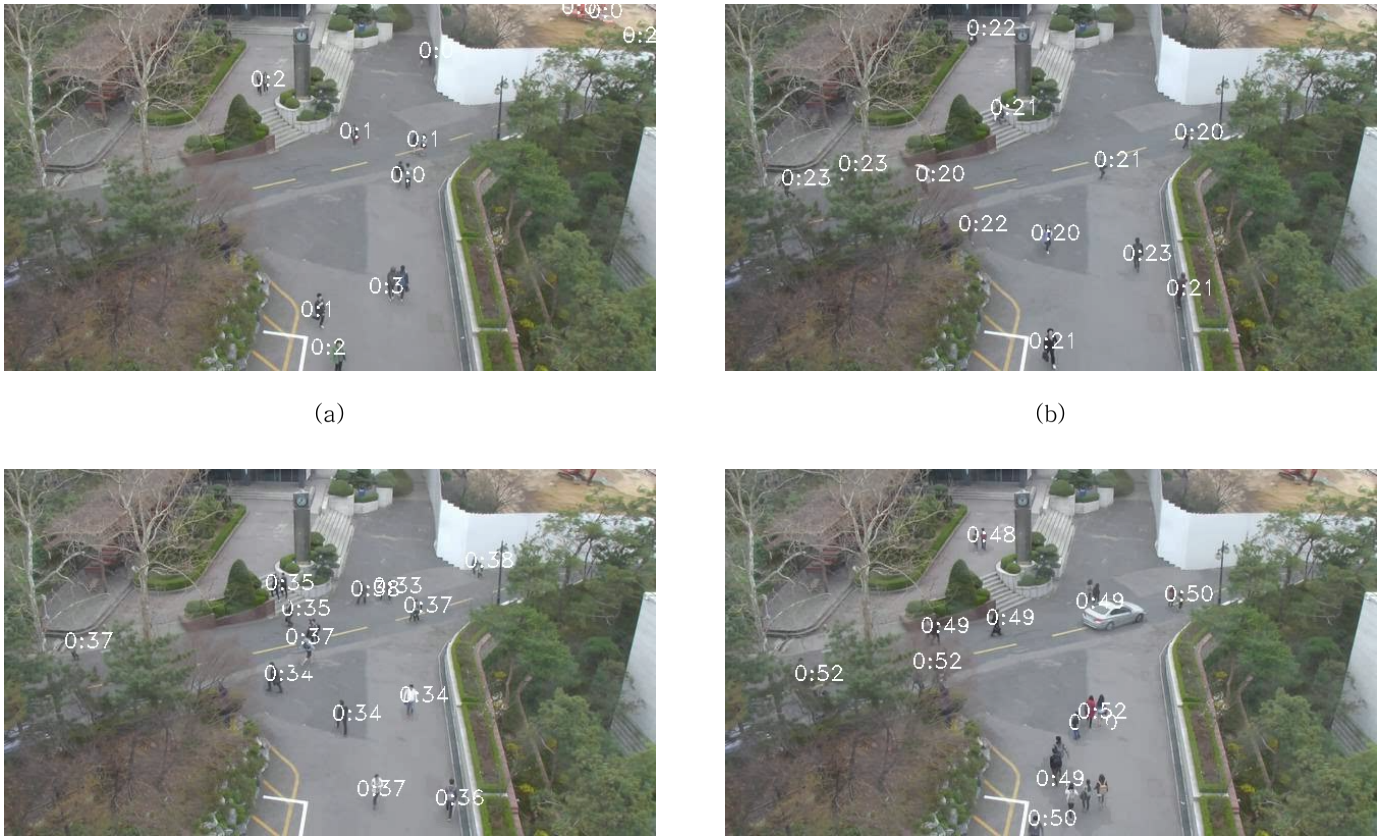
(c) (d)

Figure A-2. Some frames of the condensed video of Parking lot square II sequence (VC2).

(a) (b)

(c) (d)

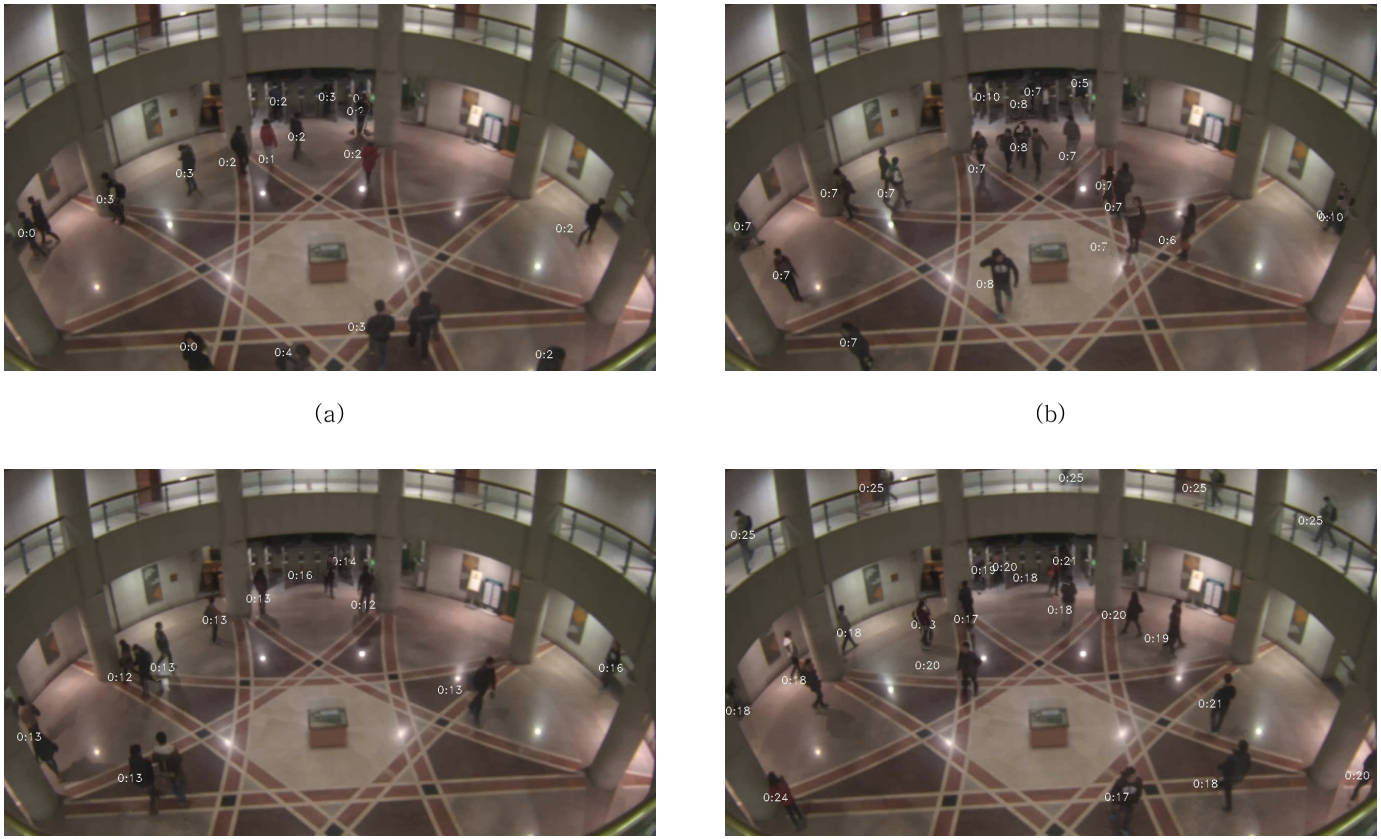
Figure A-3. Some frames of the condensed video of Crossroad I sequence (VC3).



(a) (b)

(c) (d)

Figure A-4. Some frames of the condensed video of Crossroad II sequence (VC4).



(a) (b)

(c) (d)

Figure A-5. Some frames of the condensed video of Library lobby I sequence (VC5).

(a) (b)

(c) (d)

Figure A-6. Some frames of the condensed video of Subway station plaza I sequence (VC6).

국문 요지

비디오 시놉시스는 긴 동영상을 짧게 요약하여 보안 동영상을 효율적으로 분석 할 수 있도록 돕는 기술이다. 요약 된 동영상을 생성하기 위해서는 원본 동영상에서 움직이는 물체를 물체 튜브의 형태로 추출하는 과정이 필요하다. 이후, 이 튜브들은 사전 정의 된 목적 함수를 기반으로 시간 축에서 재배치된다. 이 목적 함수는 보기 편한 요약 동영상을 만들기 위해 중요한 역할을 수행하는 몇 가지 에너지로 구성된다. 그 중 충돌 에너지는 계산 과정에서 두 개의 물체 튜브를 필요로 하기 때문에 목적 함수 계산 과정에서 병목 현상을 발생 시킨다. 이 문제를 해결하기 위해 기존 접근 방법들은 한 번에 처리되는 튜브의 수를 줄임으로써 충돌 에너지 계산에 필요한 시간을 단축한다. 하지만, 상기 접근 방법들의 계산 복잡도는 많은 수의 튜브를 사용하여 요약 동영상을 생성하기에 충분하지 않다.

본 학위논문은 온라인 비디오 시놉시스에서 효율적인 튜브 재배치 알고리즘을 제안한다. 제안하는 알고리즘은 고속 푸리에 변환(FFT)을 사용하여 충돌 에너지 계산 자체의 복잡성을 줄인다. 알고리즘 복잡성 감소의 첫 번째 과정으로 낮은 해상도를 가지는 전경 마스크인 점령 행렬을 도입하여 물체의 대략적인 위치를 표현한다. 이후, 충돌 에너지는 두 점령 행렬 사이의 연속된 요소별 곱으로 계산 할 수 있다. 이 과정은 두 개의 신호 집합에 대한 1차원 교차 상관을 계산하는 것과 동일하다. 그러므로 충돌 에너지 계산의 복잡도를 줄이기 위해 FFT를 사용할 수 있다. 또한 추가적인 연산 시간 개선을 위해 병렬 처리를

사용할 수 있다.

제안하는 알고리즘의 성능을 평가하고 분석하기 위해 한양대학교 서울 캠퍼스의 4개 장소에서 총 10시간의 동영상을 획득하였다. 모든 테스트 동영상에 대해 제안하는 알고리즘은 평균 1.89초 내에 요약 된 동영상을 생성하였고, 이는 기존 알고리즘들 보다 최소 2.26배 빠른 결과이다. 또한 이 결과는 병렬처리를 같이 사용하여 더 개선될 수 있다. 이 경우, 튜브 재배치 과정에 평균적으로 0.75초가 소요되었다. 또한, 제안하는 알고리즘은 기존 알고리즘과 비교하여 물체 사이의 충돌이 적은 요약 동영상을 생성하였다. 제안하는 알고리즘의 특징을 보다 잘 이해하기 위해 두 가지 속도 향상 기법(FFT 및 병렬 처리)의 효율성을 분석하고 제안하는 알고리즘의 파라미터 분석을 수행하였다.

감사의 글

가장 먼저 2011년 21살 어린 나이에 연구실에 석박통합과정으로 입학한 저를 박사 학위논문이 마무리되는 지금까지 지도 해주신 김회율 교수님께 깊은 감사를 드립니다. 선배들이 일찍 졸업하여 연구실에서 최고참 선배로써 지내는 기간이 길어서 무거운 책임감 때문에 항상 힘들고 어려웠지만, 후배들의 헌신적인 도움으로 여기까지 올 수 있었다는 생각을 합니다. 그리고 앞이 보이지 않는 어둠 속에 갇혀 있던 제 삶에 빛이 되어 주신 하나님께 감사드립니다. 또한 항상 제 곁에서 존재 자체로 저에게 격려가 되었던 가족들과 예은이에게 감사합니다, 그리고 사랑합니다.

연구 윤리 서약서

본인은 한양대학교 대학원생으로서 이 학위논문 작성 과정에서 다음과 같이 연구 윤리의 기본 원칙을 준수하였음을 서약합니다.

첫째, 지도교수의 지도를 받아 정직하고 엄정한 연구를 수행하여 학위논문을 작성한다.

둘째, 논문 작성시 위조, 변조, 표절 등 학문적 진실성을 훼손하는 어떤 연구 부정행위도 하지 않는다.

셋째, 논문 작성시 논문유사도 검증시스템 "카피킬러"등을 거쳐야 한다.

2019년06월20일

학위명 : 박사

학과 : 전자컴퓨터통신공학과

지도교수 : 김회율

성명 : 라문수 (서명)

한 양 대 학 교 대 학 원 장 귀 하

Declaration of Ethical Conduct in Research

I, as a graduate student of Hanyang University, hereby declare that I have abided by the following Code of Research Ethics while writing this dissertation thesis, during my degree program.

"First, I have strived to be honest in my conduct, to produce valid and reliable research conforming with the guidance of my thesis supervisor, and I affirm that my thesis contains honest, fair and reasonable conclusions based on my own careful research under the guidance of my thesis supervisor.

Second, I have not committed any acts that may discredit or damage the credibility of my research. These include, but are not limited to : falsification, distortion of research findings or plagiarism.

Third, I need to go through with Copykiller Program(Internet-based Plagiarism-prevention service) before submitting a thesis."

JUNE 20, 2019

Degree : Doctor

Department : DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

Thesis Supervisor : KIM, WHOI-YUL

Name : RA MOONSOO (Signature)