# GENERATING FLUENT TUBES IN VIDEO SYNOPSIS

*Minlong Lu*[†] *Yueming Wang*[⋆] *Gang Pan*[†]

[†]College of Computer Science
[⋆]Qiushi Academy for Advanced Studies
Zhejiang University, Hangzhou, China
{ymlml, ymingwang, gpan}@zju.edu.cn

## ABSTRACT

Video synopsis is one of the effective techniques to build a short video representation preserving the essential activities for a long video. Existing methods usually have the problem that a continuous activity (tube) from a single moving object is separated to a few small pieces. In this paper, two schemes are proposed to generate fluent tubes for video synopsis. The Gaussian mixture model and a texture method are combined to detect more compact foreground with shadow removed. The foreground constitutes a set of initial trajectories. A particle filter tracker is used to concatenate two trajectories if they belong to the same foreground activity, which generates more fluent tubes for video synopsis. Experimental results on 4 videos show that our method produces better accuracies and visual effects in video synopsis.

***Index Terms***— video synopsis, particle filter, shadow removal, fluent tube

## 1. INTRODUCTION

The amount of videos increases explosively with the growth of surveillance cameras. Most cameras work 24 hours per day. It is time-consuming and highly inefficient to find interesting activities in long videos by watching them from the beginning to the end manually. Video abstract aims at producing a brief representation of the original video while preserving key activities, which helps to take a full view of the video content quickly.

Existing work in video abstract can be categorized to two classes, still- and moving-image abstracts [1, 2]. The still-image abstract is a collection of salient key frames extracted from the video source [3, 4, 5, 6]. It can be built very fast, but loses the dynamic property of the original video. The moving-image abstract is itself a video clip with shorter length [7, 8, 9]. It possesses higher level of semantic meanings of an original video while needs higher computational cost. The video synopsis method is one of the effective moving-image abstract methods [10, 11]. It detects moving objects in an original videos, connects them to space-time tubes (object trajectories), and stitches tubes together in a synopsis video. Video synopsis provides effective abstract of videos and makes some applications (e.g., crowd counting [12] and object detection [13]) faster. However, by background cut for background subtraction, shadows are usually extracted as foreground which causes that irrelevant activities suddenly appear in some tubes. Besides, the tube (trajectory) of an object may be disconnected, leading to the sudden interruption of moving objects in the synopsis video.

This work is presented to address the above issues in video synopsis. We combine the Gaussian mixture model and a texture method to extract compact moving objects and remove shadows as much as possible. Then, based on the initial set of trajectories, a particle filter tracker is used to check if two trajectories belong to one activity. If necessary, these trajectories are concatenated to generate a more fluent tube for video synopsis. The experimental results on several videos demonstrate the effectiveness of our method.

## 2. BRIEF REVIEW OF VIDEO SYNOPSIS

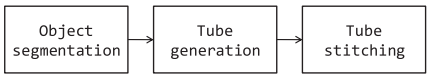


**Fig. 1**. The framework of video synopsis.

In this section, we give a brief review of video synopsis. Fig. 1 shows the main steps of video synopsis algorithm, including object segmentation, tube generation, and tube stitching.

**Object segmentation**: In [10, 11], the background cut method [14] is used to detect and segment moving objects.

**Tube generation**: A tube is generated by connecting the segmentation results of the same object in the frame sequence.

**Tube stitching**: An energy minimization step is used to determine the appearing time of each tube in the synopsis. Tubes from different time periods may appear simultaneously in the result video (See Fig. 2). The selected tubes are stitched to the background to generate the final video by Poisson Editing in [15].

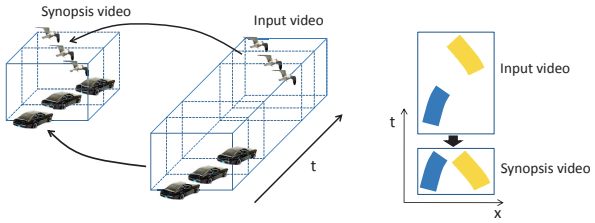


**Fig. 2**. Video synopsis: tubes from different time periods appear simultaneously.

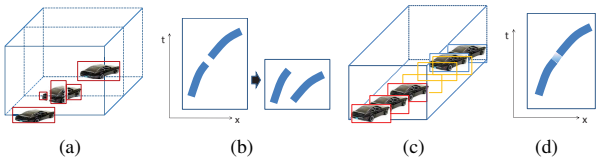


**Fig. 3**. (a) An object is segmented into several parts. (b) The tube fragments in the original video (left) and the tube interruption in the synopsis video (right). (c) Tracking from a tube (red) to another tube (blue) by estimating the rectangles (yellow). (d) The concatenating result.

There are two main drawbacks in this synopsis method:

1) In the object segmentation step, shadows are often misclassified as foreground. This misclassification can cause object merging (see Fig. 5(b)), object shape distortion, and even object missing.

2) In the tube generation step, the tube (trajectory) of an object may be disconnected, leading to the sudden interruption of moving objects in the synopsis video.

## 3. OUR METHOD

To address the above problems, we combine the Gaussian mixture model and a texture method to extract compact moving objects and remove shadows as much as possible. Then, based on the initial set of trajectories, a particle filter tracker is used to check if two trajectories belong to one activity. For those from the same activity, the trajectories are concatenated to generate a more fluent tube for video synopsis.

### 3.1. Texture-GMM method

Commonly, the texture feature of shadow is similar to that of the background. Thus, we integrate texture feature with Gaussian mixture model (GMM) to detect foreground and remove shadows.

Local binary pattern (LBP) has shown strong ability in characterizing texture and is used as the texture feature in this work. The LBP operator labels each pixel of an image with a binary number by comparing the pixel with its neighbors [16]:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P} s(g_p - g_c + a)2^p, s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}, \tag{1}$$

where $g_c$ corresponds to the gray value of the center pixel $(x_c, y_c)$ and $g_p$ denotes the gray values of equally spaced $P$ neighbor pixels. The feature vector of a particular pixel is a histogram computed over a region of radius $R$ centered at the pixel.

We measure the similarity between two LBP features as follows,

$$S(\mathbf{a}, \mathbf{b}) = \sum_{n=1}^{N} \min(a_n, b_n), \tag{2}$$

where $\mathbf{a}$ and $\mathbf{b}$ are the LBP features and $N$ is the dimension of the features.

Our texture-GMM foreground detection method has three steps:

1. background construction;
2. foreground extraction by the LBP feature and GMM;
3. foreground combination.

**Background construction**: In this step, we construct two background models. One is the popular Gaussian mixture model [17] and the other is a temporal median over a clip before and after each frame. The purpose of the temporal median is to construct a background image so that the LBP feature can be computed on background. The length of the clip for the temporal median computation is 450 frames before and after the current frame, thus is 900 frames in total.

**Foreground extraction by the LBP feature and GMM**: We extract two foregrounds for each frame at this step. Firstly, the GMM-based foreground is extracted. Then, for each frame, the LBP features of every pixel on the median image and the current frame are computed. These two features are compared by (2). If the score of a pixel is less than a threshold, it is marked as foreground. In our experiment, the threshold is set to 0.6 which is able to reach the satisfied accuracy.

**Foreground combination**: A pixel is labeled as the foreground only if it is labeled as foreground in both the GMM-based foreground and the temporal-median foreground. By this combination scheme, a final foreground is obtained.

### 3.2. Tracking for Tube Concatenating

By our texture-GMM method, more compact foreground can be extracted which helps to build fluent tubes for synopsis. However, if one object is segmented into several parts, the tube disconnection can happen, leading to sudden interruption of objects in the synopsis video, as shown in Fig. 3 (a) and (b). We use particle filter tracking algorithm [18] to concatenate tube fragments and generate fluent tubes.

Denoting $x_t$ and $y_t$ the hidden state and observation data at time $t$ and given all available observation $y_{0:t} = \{y_0, \ldots, y_t\}$, the filtered estimation of $x_t$ is $p(x_t|y_{0:t})$, obeying the recursion [18]:

$$p(x_t|y_{0:t}) \propto p(y_t|x_t) \int_{x_{t-1}} p(x_t|x_{t-1}) p(x_{t-1}|y_{0:t-1}) dx_{t-1}, \tag{3}$$

where $p(x_t|x_{t-1})$ is the transition model specifying how objects may move between frames and $p(y_t|x_t)$ is the observation model representing the likelihood of objects being in specific states.

A set of $M$ weighted particles $\{x_t^i\}_{i=1\ldots M}$ is used to approximate the filtering distribution $p(x_t|y_{0:t})$,

$$p(x_t|y_{0:t}) \approx \sum_{i=1}^{M} w_t^i \delta(x_t - x_t^i). \tag{4}$$

The particles $\{x_t^i\}_{i=1\ldots M}$ are drawn from $p(x_t|x_{t-1})$, and the weight $w_t^i$ for the $i$th particle is chosen to be the data likelihood $p(y_t|x_t^i)$.

The prior distribution $p(x_0)$ for the tracking process is based on the HSV histogram in an image patch, and this histogram is regarded as reference histogram. Similar to [18], we use the second-order auto-regression dynamics model as the transition model. The observation model is:

$$p(y_t|x_t) \propto e^{-\lambda D^2[\boldsymbol{h_0}, \boldsymbol{h_t}(\boldsymbol{x_t})]}, \tag{5}$$

where $D[\boldsymbol{h_0}, \boldsymbol{h_t}(\boldsymbol{x_t})] = [1 - \sum_{n=1}^{N} \sqrt{h_0(n) h_t(n; \boldsymbol{x_t})}]^{1/2}$ defines the distance between reference HSV histogram $\boldsymbol{h_0}$ and current histogram $\boldsymbol{h_t}(\boldsymbol{x_t})$.

For a set of initial trajectory fragments, the tracker is used to concatenate the disconnected fragments. Suppose each trajectory (tube) $t_i$ is represented by a sequence of bounding boxes of the object in each frame, $t_i = \{r_{i,1}, \ldots, r_{i,n}\}$, where $r_{i,j} = (x_1, x_2, y_1, y_2), (j = 1, \ldots, n)$ denotes the bounding box around the object. For a tube $t_i$ in the set, we start from its last bounding box $r_{i,n}$ and continuously track the image patch in the box on $k$ subsequent frames. When there exist a tube $t_j$ starting from a frame such that its first bounding box has significant overlap with the tracked rectangle in this frame, we compute the distance between the histograms in the two rectangles. If the distance is smaller than a threshold, we regard $t_i$ and $t_j$ as the same object activity and concatenate them (see Fig. 3(c)).

## 4. EXPERIMENTS

Four videos are captured to evaluate our method including both indoor and outdoor scenes as follows:

1. Playground: 11410 frames, recorded on a playground (outdoor).
2. Corridor: 1078 frames, recorded in a building (indoor).

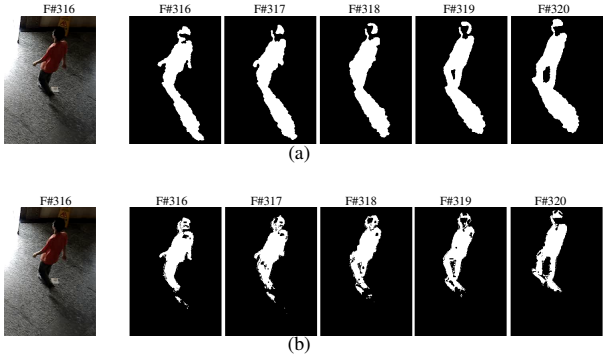


**Fig. 4**. Foreground extraction results on the "Corridor" video. Most shadows are removed from foreground by our method. F#316 denotes the 316th frame in the video. (a) The background cut method. (b) Our texture-GMM method.


**Fig. 5**. The background cut method merges two objects while our texture-GMM method segments two people correctly. (a) Input frames. (b) The results of the background cut method. (c) The results of the texture-GMM method.

3. Road1: 2716 frames, recorded in a road beside a building (outdoor).
4. Road2: 2431 frames, recorded in a road beside a gymnasium (outdoor).

On these videos, we test the texture-GMM method, compared it with the background cut method used in [10], and evaluate the effect of tube concatenating.

### 4.1. Evaluating Texture-GMM Method

Fig. 4 shows the foreground extraction results of our texture-GMM method and background cut method [10] on "Corridor". It can be found that most shadows are removed by our method.

The compact foreground helps to avoid object merging. Fig. 5 shows two example frames in the "Playground" and "Road2" videos. Our texture-GMM method extracts two people correctly in each frame while the background method outputs only one object.

We consider two video clips from "Playground" and "Road2" in which two objects are continuously close to each other and easily causes object merging problem. The object number in the clips are manually counted as the ground truth first. Then, we run our texture-GMM method and the background cut method on these clips to compare the foreground

**Table 1**. Comparison of total object number in foreground extraction between our method and the background cut method.

| | Length (frame#) | Ground truth | Background cut [10] | Our method | Improvement |
|---|---|---|---|---|---|
| Playground clip | 90 | 180 | 117 | 154 | 20.6% |
| Road2 clip | 30 | 60 | 41 | 55 | 23.3% |

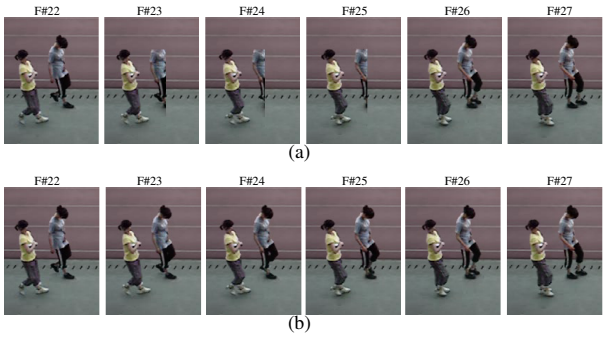


**Fig. 6**. Synopsis video comparison by different foreground extraction methods. (a) Irrelevant object suddenly appear or disappear by the background cut method. (b) More fluent tube and synopsis results by our texture-GMM method.

extraction results. Table 1 shows the comparison. More than 20% improvement in the accuracy of the object number can be obtained by our method.

In video synopsis, the object merging problem leads to undesired result. As shown in Fig. 6(a), irrelevant object suddenly appears/disappears in some tubes. With our result of more compact foreground, more fluent tubes and better visual results are obtained (see Fig. 6(b)).

### 4.2. Evaluating Tube Concatenating

As stated in the previous section, by our texture-GMM method, most object merging cases in foreground extraction can be avoided. However, it can not handle the case if one object is segmented into several parts which leads to tube disconnection and further causes sudden interruption of objects in the synopsis video. As shown in Fig. 7(a), some objects suddenly interrupted due to the tube disconnection. By tracking, the disconnected tube fragments can be concatenated, obtaining more fluent tubes in synopsis video. Fig. 7(b) shows the improved result by our method.

In Table 2, we compare the total tube numbers before/after tracking. The ground truth is manually counted. It can be found that the tube numbers by our method is much closer to the ground truth. This means many tube fragments belonging to one object activity are successfully concatenated.

Fig. 8 shows an example frame sequence of the final synopsis video by our texture-GMM and tracking method.

**Table 2**. Comparison of total tube numbers before/after tracking.

| | Length (frame#) | Ground truth | Without Tracking [10] | With Tracking (Ours) |
|---|---|---|---|---|
| Playground | 11410 | 38 | 103 | 42 |
| Corridor | 1078 | 4 | 7 | 5 |
| Road1 | 2716 | 8 | 17 | 8 |
| Road2 | 2431 | 7 | 8 | 7 |

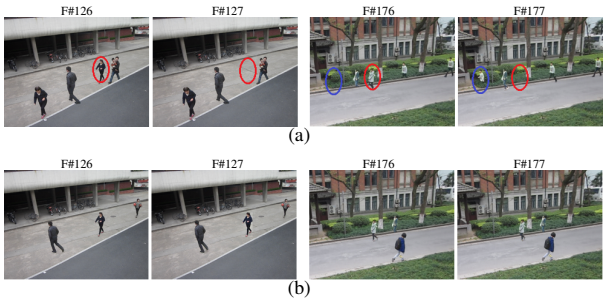


**Fig. 7**. Synopsis video comparison before/after tracking. (a) Without tracking, objects are suddenly interrupted. (b) More fluent tubes are obtained by tracking.

## 5. CONCLUSION

In this paper, we have presented two schemes to generate more fluent tubes for video synopsis. The texture-GMM method was proposed to extract more precise foreground and the tracking algorithm was applied to tube fragment concatenating. Experimental results show that our method outperforms the existing methods.

## 6. ACKNOWLEDGEMENT

This work was partly supported by National 973 Program (2013CB329504), National Natural Science Foundation of China (No. 61070067, No. 61103107), and Research Fund for the Doctoral Program of Higher Education of China (No. 20110101120154).

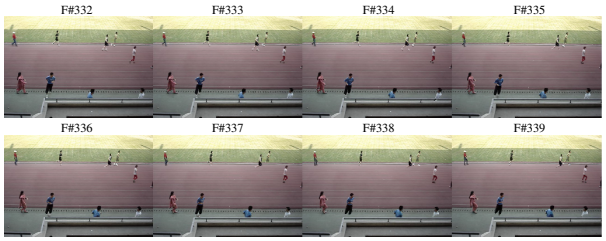


**Fig. 8**. An example frame sequence of the final synopsis video by our method. Fluent tubes can be seen in the sequence.

## 7. REFERENCES


[1] Y. Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," *HP Laboratories Palo Alto*, 2001.

[2] J. Oh, Q. Wen, S. Hwang, and J. Lee, "Video abstraction," *Video data management and information retrieval*, pp. 321–346, 2005.

[3] M. Mills, J. Cohen, and Y.Y. Wong, "A magnifier tool for video data," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1992, pp. 93–98.

[4] Y. Taniguchi, A. Akutsu, Y. Tonomura, and H. Hamada, "An intuitive and efficient access interface to real-time incoming video based on automatic indexing," in *Proceedings of the 3rd ACM international conference on Multimedia*, 1995, pp. 25–33.

[5] J. Nam and A.H. Tewfik, "Video abstract of video," in *IEEE 3rd Workshop on Multimedia Signal Processing*, 1999, pp. 117–122.

[6] C. Kim and J.N. Hwang, "An integrated scheme for object-based video abstraction," in *Proceedings of the 8th ACM international conference on Multimedia*, 2000, pp. 303–311.

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

[8] Y.F. Ma, X.S. Hua, L. Lu, and H.J. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 907–919, 2005.

[9] H.W. Kang, X.Q. Chen, Y. Matsushita, and X. Tang, "Space-time video montage," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 1331–1338.

[10] 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, 2008.

[11] Y. Pritch, S. Ratovitch, A. Hendel, and S. Peleg, "Clustered synopsis of surveillance video," in *6th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2009, pp. 195–200.

[12] M Hashemzadeh, G Pan, and M Yao, "Counting moving people in crowds using motion statistics of feature-points," *Multimedia Tools and Applications*, pp. 1–35, 2013.

[13] M Tan, Y.M Wang, and G Pan, "Feature reduction for efficient object detection via l1-norm latent svm," in *Intelligent Science and Intelligent Data Engineering*, pp. 322–329. 2013.

[14] J. Sun, W. Zhang, X. Tang, and H.Y. Shum, "Background cut," *Computer Vision–ECCV 2006*, pp. 628–641, 2006.

[15] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM Transactions on Graphics (TOG)*, 2003, vol. 22, pp. 313–318.

[16] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.

[17] Z Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition*, 2004, vol. 2, pp. 28–31.

[18] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *Computer Vision–ECCV 2002*, pp. 661–675, 2002.