Surveillance video motion segmentation based on the progressive spatio-temporal tunnel flow model

Yunzuo Zhang, Wenxuan Li, and Panliang Yang
School of Information science and Technology, Shijiazhuang Tiedao University, 17 North Second Ring East Road, Shijiazhuang, China
Email: zyy2016@stdu.edu.cn

Introduction: With the rapid construction and development of smart cities, video data captured by surveillance cameras are increasing at an explosive rate. Browsing videos in an effective manner has become an urgent problem. Video summarisation, as an active research topic, enables users to browse massive surveillance videos effectively [1]. Motion segmentation is the first and important step of surveillance video summarisation. Therefore, how to effectively extract motion segments becomes particularly important.

Existing surveillance video motion segmentation methods are performed by determining whether there is motion in the frame. For example, Wang et al. [2] used the inter-frame feature difference method to determine whether the current frame contains moving targets. Murtaza et al. [3] computed the energy of motion history images, which provides spatio-temporal information of motion. The lower energy segments are static segments. Sheng et al. [4] proposed a new trajectory clustering method using submodular optimisation for motion segmentation. Stoffregen et al. [5] segmented the scene into multiple targets to achieve motion segmentation. Zhou et al. [6] proposed a novel multi-mutual consistency inducted transfer subspace learning framework for human motion segmentation. Guo et al. [7] improved the optical flow method to extract moving targets more accurately. These methods can achieve high segmentation accuracy, but they need to process the video frame by frame. And some deep learning-based segmentation methods require a large amount of sample data to train the network, which require extremely high-quality sample data and are relatively time consuming.

The traditional video temporal domain segmentation is to divide video into temporal structure units. Surveillance video motion segments segmentation aims to divide the video into motion segments and static segments, which is similar to traditional video temporal domain segmentation. Based on this observation, this letter extracts motion segments by detecting the boundaries between motion segments and static segments. It only needs to process the pixels on the circular sampling line of the video to construct the video progressive spatio-temporal tunnel (STT) flow model. The proposed method can greatly reduce the data processing capacity without reducing the accuracy of motion segments segmentation.

Proposed method: The present letter proposes a novel method for surveillance video motion segmentation. It is based on the analysis of the pixels on the circular sampling line, and the amount of calculation is very light. By expanding and processing the progressive STT, a progressive STT flow model is constructed to obtain motion segments. Figure 1 shows the flow chart of the proposed method.

1. Progressive STT: In order to detect motion inside the surveillance area, progressive circular sampling lines of all the frames are sampled and arranged on the time axis to form a progressive STT. Simultaneously, the surveillance area is divided into multiple sub-surveillance areas, as shown in Figure 3. It is almost impossible for the target to move in a standard circle, thus the target will definitely cross a sampling line, either moving in the surveillance areas or crossing the surveillance area.

2. Progressive STT expansion diagram: In order to get the trajectory and direction of the target, the STT needs to be expanded. Suppose the length of a video sequence is \( L \). The pixel in the \( r \)th row and \( j \)th column of the \( k \)th frame can be expressed as \( P_{i,j}^k \). \( R \) and \( S \) are the radius and perimeter of circular sampling lines. And the centre is (\( centreX, centreY \)), the coordinates of pixel \((i, j, \theta)\) on the circular sampling line are as follows:

\[
\begin{align*}
\{ i_0 & = centreX - R \times \sin(\theta) \\
\{ j_0 & = centreY + R \times \cos(\theta) \}
\end{align*}
\] (1)

The circular sampling line with radius \( R \) and the coordinates of each point are arranged from top to bottom to form a vector \( CR \) whose length is the perimeter of the circular sampling line, as shown below:

\[
CR^k = \{ P_{i_0,j_0}^k, P_{i_1,j_1}^k, P_{i_2,j_2}^k, \ldots, P_{i_S,j_S}^k, P_{i_1,j_1}^k, P_{i_2,j_2}^k, \ldots, P_{i_S,j_S}^k \}^T.
\] (2)

These vectors from successive frames are connected along the timeline to form the STT expansion diagram matrix Tunnel\((R)\), as follows:

\[
\text{Tunnel}(R) = \begin{bmatrix}
P_{i_1,j_1}^1 & P_{i_1,j_1}^2 & \cdots & P_{i_1,j_1}^k & \cdots & P_{i_1,j_1}^S \\
P_{i_2,j_2}^1 & P_{i_2,j_2}^2 & \cdots & P_{i_2,j_2}^k & \cdots & P_{i_2,j_2}^S \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
P_{i_S,j_S}^1 & P_{i_S,j_S}^2 & \cdots & P_{i_S,j_S}^k & \cdots & P_{i_S,j_S}^S \\
\end{bmatrix}
\] (3)

3. Progressive STT flow model: Motion segments are obtained by constructing the progressive STT flow model. As shown in Figure 3, a surveillance area can be divided into multiple sub-surveillance areas. The letter describes that, when there is no moving target in the first frame of the video, the target entering the sub-surveillance area is assigned positive spatio-temporal flow, and the target exiting the sub-surveillance area is assigned negative spatio-temporal flow, as shown in Figure 4.

Sub-sampling lines which are adjacent to the sampling line inside sub-surveillance areas are used to determine the motion direction. Each column of the STT expansion diagram corresponds to the sampling line at the same position of the video, and each column of the STT expansion diagram is regarded as an input of Gaussian mixture background model [10], as a result, the trajectory of motion in the STT expansion diagram is extracted. The target trajectory in the STT expansion diagram formed by the sub-sampling line can be obtained similarly. Comparing the sequence of target appearances in the two STT expansion diagrams, the moving direction of the targets can be obtained. After that, the proposed method detects the target centre, and sets the value of mpxels around the centre to 255. This standardises the moving target size. Figure 5 shows the expansion diagram and its processing.
The spatio-temporal flow can be calculated according to the number of white pixels in a single video frame, and the spatio-temporal flow of each STT expansion diagram is calculated as follows:

\[ F_n(f_k) = \sum_{i=1}^{S} p_{i,k} \]  \hspace{1cm} (4)

\[ p_{i,k} = \begin{cases} 1, & \text{entry} \\ -1, & \text{exit} \end{cases} \]  \hspace{1cm} (5)

Where, \( n \) represents the \( n \)th STT expansion diagram, \( k \) represents the \( k \)th frame of the video sequence, and the \( i \)th column of the STT expansion diagram. \( S \) represents the perimeter of the circular sampling line, and the height of the STT expansion diagram. Suppose \( N \) is the number of sub-surveillance areas. The spatio-temporal flow \( F(f_k) \) and the accumulative spatio-temporal flow \( AF(f_k) \) of the \( n \)th sub-surveillance region can be obtained by:

\[ F^n(f_k) = F^n(f_{k-1}) + F_n(f_k), \quad (n = 1 \cdots N, \quad F_0(f_k) = 0) \]  \hspace{1cm} (6)

\[ AF^n(f_k) = \sum_{i=1}^{k-1} F^n(f_i) \]  \hspace{1cm} (7)

Connecting the spatio-temporal flow and accumulative spatio-temporal flow of each frame, the spatio-temporal flow curve and accumulative spatio-temporal flow curve of the sub-surveillance area can be obtained, as shown in Figure 6.

4. Motion segments: When the accumulative spatio-temporal flow is greater than 0, it means that there are moving targets in the sub-surveillance area. Through the above, the entire surveillance area is divided into \( N \) sub-surveillance areas, and the motion segment \( S(i) \) of the sub-surveillance area is obtained by the accumulative spatio-temporal flow curve. The motion segment \( S \) is as follows:

\[ S = S(1) \cup S(2) \cup \cdots \cup S(i) \cup \cdots \cup S(N). \]  \hspace{1cm} (8)
Table 1. Experimental performance

| Methods            | Precision (%) | Recall (%) | F1 (%) | Time consuming (s) |
|--------------------|---------------|------------|--------|-------------------|
| Method in [2]      | 85.1          | 78.2       | 81.3   | 504.9             |
| Method in [3]      | 67.6          | 94.6       | 76.6   | 721               |
| Method in [7]      | 83.8          | 84.7       | 83.8   | 519               |
| Our method (1C)    | 98.8          | 70.1       | 81.0   | 28.05             |
| Our method (2C)    | 97.5          | 75.1       | 83.9   | 62.82             |
| Our method (3C)    | 97.4          | 75.7       | 84.1   | 92.05             |

Fig. 8 The experimental results on each video. (a) The comparison of precision. (b) The comparison of time consuming.

Experiments: The experiments were performed on a general-purpose computer with an Intel Core 3.4 GHz CPU and 16 GB memory. The video involved in this experiment were from 8 real surveillance scenes, including 10 videos. Figure 7 shows part of the experimental process of video1. It can be seen that when the car crosses the sampling line, the spatio-temporal flow varies with the car entering or exiting sub-surveillance areas. Table 1 shows the average precision, recall, F1 and time consuming of test methods: Method in [2], Method in [3], Method in [7], the method presented here with one sampling circle (1C), two sampling circles (2C), and three sampling circles (3C). Figure 8 shows the comparison of precision and time consuming on each video.

From Table 1 and Figure 8, it can be seen that the proposed method outperforms contrastive algorithms in terms of segmentation precision and F1. And the time consuming of the proposed method is much smaller than contrastive methods. Suppose the size of a video sequence is \( S \times L \), where \( S \) is the size of video frame, and \( L \) is the length of the video sequence. When the perimeters of circular sampling lines are, respectively, \( s_1, s_2, \ldots, s_n \), the computational complexity of the proposed method is \( O(L \times (s_1 + s_2 + \ldots + s_n)) \); whereas \( O(L \times S) \) is the computational complexity for the contrastive methods. Commonly, \( (s_1 + s_2 + \ldots + s_n) < cS \) holds, thus the computational efficiency of the proposed method is higher than that of contrastive methods.

Different from the existing method of motion detection frame by frame, the paper proposes a new method of spatio-temporal slice, which greatly reduces the amount of data processing. Compared with the existing horizontal and vertical slices, the new circular sampling method proposed can better take into account the target movement in all directions. From the last three rows in Table 1, it can be seen that \( F_1 \) increases with the number of sampling circles. The main reason is that the progressive STT can better detect targets that only move inside the surveillance area.

Conclusion: The present letter proposes a novel method of surveillance video motion segmentation based on progressive STT flow model. The proposed method analyses the video through a small number of pixels on the circular sampling line, and constructs a progressive STT flow model to segment surveillance video motion segments. In future work, we will plan to perform motion classify on the motion segments.

Acknowledgement: This work was supported by the National Nature Science Foundation of China under Grant 61702347, 61972267.

Conflict of interest: The authors declare that they have no conflicts of interest. The data that support the findings of this study are available on request from the corresponding author.

© 2021 The Authors. Electronics Letters published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Received: 7 December 2020 Accepted: 1 April 2021
doi: 10.1049/ell-12186

References
1. Zhang, Y., Tao, R., Wang, Y.: Motion-state-adaptive video summarization via spatiotemporal analysis. IEEE Trans Circuits Syst. Video Technol. 27(6), 1340-1352 (2017). https://doi.org/10.1109/TCSVT.2016.2539638
2. Wang, Y.P., Li, R.W., Liu, X.: Extraction method of surveillance video synopsis combines objects and key frames. Industrial Control Comput. 28(3), 11–13 (2015). https://doi.org/10.3969/j.issn.1001-182X.2015.03.005
3. Murtaza, F., Yousaf, M.H., Velastin, S.: PMHI: Proposals from motion history images for temporal segmentation of long uncut videos. IEEE Signal Process Lett. 25(2), 179–183 (2018).https://doi.org/10.1109/LSP.2017.2778190
4. Shen, J., Peng, J., Shao, L.: Submodular trajectories for better motion segmentation in videos. IEEE Trans Image Process. 27(6), 2688–2700 (2018). https://doi.org/10.1109/TIP.2018.2795740
5. Stoffregen, T., et al.: Event-based motion segmentation by motion compensation. In: IEEE/CVF International Conference on Computer Vision (ICCV), Seoul , pp. 7243–7252, (2019). https://doi.org/10.1109/ICCV.2019.00734
6. Zhou, T., et al.: Multi–mutual consistency inducted transfer subspace learning for human motion segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), Seattle, pp. 10274–10283, https://doi.org/10.1109/CVPR42600.2020.01029
7. Guo, F., et al.: Motion-aware rapid video saliency detection. IEEE Trans. Circuits Syst. Video Technol. 30(12), 4887–4898 (2020). https://doi.org/10.1109/TCSVT.2019.2906226
8. Carneiro, S.A., et al.: Fight detection in video sequences based on multi-stream convolutional neural networks. In: SBIGRAPI Conference on Graphics, Patterns and Images (SIGRAPI), Rio de Janeiro (2019). https://doi.org/10.1109/SBIGRAPI.2019.00010
9. Souza, M.R.E., Pedrini, H.: Visual rhythms for qualitative evaluation of video stabilization. EURASIP J Image Video Process 2020(1), 19 (2020). https://doi.org/10.1186/s13640-020-00508-4
10. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, pp. 246–252 (1999). https://doi.org/10.1109/CVPR.1999.784637