

# Video abstract system based on spatial-temporal neighborhood trajectory analysis algorithm

Han Huang<sup>1</sup> · Shen Fu<sup>1</sup> · Zhao-Quan Cai<sup>2</sup> · Bin Li<sup>1</sup>

Received: 25 April 2017 / Revised: 13 December 2017 / Accepted: 15 December 2017 /  
Published online: 12 January 2018  
© The Author(s) 2018. This article is an open access publication

**Abstract** In this paper, a video abstract system based on spatial-temporal neighborhood trajectory analysis algorithm which is mainly used to process surveillance videos is proposed. The algorithm uses the spatial adjacency of foreground targets and tracks the spatial-temporal neighboring moving targets to get their whole trajectories in order to meet the requirement of processing speed and accuracy. The indicators consist of trajectory detection rate, trajectory tracking average continuity and video abstract processing speed are used to evaluate the effectiveness of the system. We compare the algorithm with the other three algorithms, and the results show that spatial-temporal neighborhood trajectory analysis algorithm has sufficient trajectory detection rate and processing speed for surveillance video abstraction.

**Keywords** Spatial-temporal neighborhood · Trajectory analysis · Video abstract

## 1 Introduction

With the popularity of the camera and development of video surveillance system [24], video has gradually become one of the main carrier of daily life records [9, 19]. Nowadays, security cameras are deployed in various places. They have attracted both industries and researchers due to its importance for security [2, 5, 10]. However, the video data they produced is huge. Only checking manually can not keep up with the speed of video

---

✉ Zhao-Quan Cai  
cai@hzu.edu.cn  
Han Huang  
hhan@scut.edu.cn

<sup>1</sup> South China University of Technology, Guangzhou, Guangdong 510006, China

<sup>2</sup> College of Huizhou, Huizhou, Guangdong 516007, China

production [8, 20]. Massive data of surveillance video makes the demand of surveillance video automatic processing increasingly urgent [11, 14].

Video abstraction [6] is a mechanism to produce brief summary of the video content which focuses on showing all the events in the video (ie, the movement of the moving target) in the shortest possible time. The video abstraction process usually consists of two parts: foreground target detection and trajectory estimation. There are many algorithms developed for analyzing the trajectories of the foreground targets [22], but they may not be suitable for handling surveillance video data. The surveillance videos have special characteristics such as static background and simple motion trajectory, meanwhile, the processing speed need to meet the requirements in order to achieve real-time. But the analysis and processing speed of algorithms which can accurately analyze the motion trajectory, such as multi-target tracking by rank-1 tensor approximation [18], is far from real-time requirements, because the improvement on accuracy might come with the cost of increasing the execution time [1]. Therefore, developing a trajectory analysis algorithm which can meet requirements of the speed and the accuracy at the same time is significant.

The method of constructing the trajectory tracking initial solution in discrete-continuous optimization for multi-target tracking is based on the following hypothesis. The closer time and space two points in, the higher probability that they belong to the same trajectory. Based on above hypothesis, and inspired by the method of constructing the trajectory tracking initial solution in discrete-continuous optimization for multi-target tracking, this paper proposes a spatial-temporal neighborhood trajectory analysis algorithm. The algorithm selects the existing foreground detection algorithm which is most suitable for processing the surveillance video data as the foreground detection step, and extracts the trajectories of the targets on the basis of it. If we find two moving targets are adjacent both in time and space, i.e. spatial-temporal neighborhood, they will be assigned to the same track. By tracking the spatial-temporal neighboring moving targets, we can get their whole trajectories. After the complete analysis of the entire video, the complete motion trajectory of each moving target is calculated. The spatial-temporal neighborhood trajectory analysis algorithm is effective for surveillance videos, with sufficient processing speed and accuracy.

Fast multi-target tracking is the key technology of the video abstract system due to the accuracy and computational complexity of the multi-target tracking affect the accuracy and computational complexity of the video abstract system directly. Although the existing multi-target tracking methods have achieved a high level of accuracy and precision, the video abstract applications have put forward new requirements for multi-target tracking technology. The multi-target tracking method focuses on the indicators MOTA (Multiple Object Tracking Accuracy, which measures how many distinct errors including missed targets, ghost trajectories, or identity switches are made) and MOTP (Multiple Object Tracking Precision, which measures how well the bounding boxes of objects are matched) [15]. However, video abstract is not sensitive to factors such as tracking precision. The existing multi-target tracking method can not meet the requirement of speed and accuracy in video abstract system at the same time. Therefore, we use three indicators introduced in Section 5 to measure the surveillance video abstract system.

The rest of the paper is structured as follows. Preliminaries are given in Section 2 and related works are given in Section 3. The overall processes of algorithm are described in Section 4. The systematic indicators are described in Section 5. Experiment and results are presented in Sections 6 and 7 concludes the paper.

## 2 Preliminaries

Intelligent surveillance requires a brief overview of the events that occur in the video, that is, the ability to generate video abstracts. In order to detect suspicious events in time and take the response in advance, video abstracts require real-time display of surveillance video events. Therefore, the algorithm of analysis video processing and synthesis abstract must have small time complexity in order to meet the real-time requirements.

The basic function of video abstract system is to analyze the input video stream and extract the abstract of moving targets. The video abstract system obtains inputting video stream, detecting contours of moving objects foreground detection, and then performs the trajectory analysis to get the trajectory of each moving object in the video finally.

The video abstract usually provides two functions: event abstract and full video condensed abstract. The event abstract means that cutting out the time period of moving targets from original video and marking the movement of the target trajectories in the video to form a short video. The full video condensed abstract means that various moving targets trajectories which have been processed to be translucent are merge together in the background image that does not contain moving targets, to form a condensed video.

The output of event abstract is a video clip that clearly reflects the actual movement of the foreground targets while the output of full video condensed abstract is a condensed video clip in which multiple foreground targets go through the background of video. These two different functions meet different requirements. The event abstract is used for the user to see the actual movement of the foreground targets which has the advantage of clear and reflecting the actual situation. Full video condensed abstract allows users to once and quickly view all the events that occurred in the video.

To get the video abstracts, we need to extract the trajectories of the objects in the video first, and then visualize the movement trajectories to form a video abstract. In order to calculate the motion trajectory of an object, it is necessary to perform the foreground target detection on the image sequence of the video first and then preprocess the foreground image to extract the contour of the moving object.

Foreground target detection, that is, the detection of moving objects in the video, has been one of the hot spots in the field of video surveillance. The pixels in the image can be front points or background points. The goal of foreground target detection is to distinguish the foreground and background points of each frame. Motion trajectory analysis is based on foreground target detection. The input of the motion trajectory analysis is the foreground targets' information (the position and size of the foreground target in the image coordinates), and the outputs are the trajectories of the foreground targets, i.e., the classification of all images of foreground targets according to whether it belongs to the same trajectory. Based on the classification the changing video image sequence could be extracted.

## 3 Related works

The video abstraction needs to detect the moving targets first, i.e., foreground target detection. Then analysis the trajectories of the detection results. The goal of foreground detection is to extract the changing video image sequence from the stationary background. The pixels in the image can be front points or background points. The goal of foreground target detection is to distinguish the foreground and background points of each frame.

At present, foreground detection algorithms include frame difference method, SACON (sample consistency detection algorithm), Gaussian background modeling algorithm, Vibe algorithm, statistical average method [25] and so on. These algorithms have their own advantages and disadvantages. The frame difference method, Vibe algorithm and Gaussian background modeling algorithm are the most widely used and practical.

Motion trajectory analysis is based on foreground target detection [12]. The input of the motion trajectory analysis is the foreground targets' information (the position and size of the foreground target in the image coordinates), and the outputs are the trajectories of the foreground targets, i.e., the classification of all images of foreground targets according to whether it belongs to the same trajectory. Based on the classification the changing video image sequence could be extracted. There are usually two ideas used for motion trajectory analysis. One is based on the trend of movement of the current trajectory to predict the next position of the moving object. Another is analyzing the trajectory as a multi-dimensional distribution problem, and assigning detected foreground targets to each track.

In the literature [17], a discrete-continuous optimization for multi-target tracking is proposed. The algorithm regards trajectory analysis as a multidimensional distribution problem whose goal is to minimize the sum of the discrete-continuous energies of the trajectory. The distribution scheme of the foreground target detection point is the prototype of the movement trajectory. The trajectory of each foreground target is obtained by performing the trajectory fitting on the basis of the distribution scheme of the detection points. The algorithm continues to perform iterations of the reassignment and trajectory fitting until a smaller discrete-continuous energy value can not be found.

In the literature [13], a motion trajectory analysis algorithm based on position prediction is proposed which predicts the motion trajectory by estimating, finding and verifying the foreground target position iteratively. The algorithm takes the velocity and direction of motion of the existing trajectory into account, and predicts the next position of the moving object [26]. The goal is to find out whether the moving object, which will be matched with the known trajectory, can be found in the neighborhood of the prediction point.

Recently, tracking-by-detection methods have shown impressive performance improvement in multi-object tracking which can be roughly categorized into batch and online methods. The tracking-by-detection methods generally build long tracks of objects by associating detections provided by detectors and they can recover tracking failures by finding the object hypothesis from the detections. Visesh Chari et al. proposed [4] in 2015 to add pairwise costs to the min-cost network flow framework and design a convex relaxation solution with an efficient rounding heuristic which empirically gives certificates of small suboptimality. In 2017, Seung-Hwan Bae et al. [3] proposed a robust online MOT method which uses confidence-based data association to handle track fragments due to occlusion or unreliable detections.

## 4 The overall process of algorithm

For each frame of the video to be processed, the spatial-temporal neighborhood trajectory analysis algorithm first detects the foreground target using region-based mixture of gaussians modelling [21] and obtains the foreground image followed by three preprocess operations. They are noise reduction, removal of voids and extracting the circumscribed rectangle of the outline using median filter detection algorithm [16], down sampling plus up sampling and line scan clustering algorithm [23], respectively. When the bounding rectangle

of the foreground target is obtained, determining the beginning of the trajectory, updating trajectory, and determining the end of the trajectory are successively performed. Then each frame of input image and the motion trajectories of each foreground target in the video will be processed in chronological order.

#### 4.1 The basic idea of spatial-temporal neighborhood trajectory analysis

The method of constructing the trajectory tracking initial solution in discrete-continuous optimization for multi-target tracking is based on the following hypothesis. The closer time and space two points in, the higher probability that they belong to the same trajectory. Using the Roulette-wheel selection method [13] to classify each test point according to the probability level, each initial motion trajectory is generated.

Based on the above hypothesis, and inspired by the method of constructing the trajectory tracking initial solution in discrete-continuous optimization for multi-target tracking, this paper proposes a spatial-temporal neighborhood trajectory analysis algorithm. If we find two moving objects are adjacent both in time and space, i.e. spatial-temporal neighborhood, they will be assigned to the same track. After the complete analysis of the entire video, the complete motion trajectory of each moving target is calculated.

As other trajectory analysis algorithms, the spatial-temporal neighborhood trajectory analysis algorithm must obtain the position and size of the foreground target. Before we can determine the spatial position of the foreground object, we must reduce the interference of the ambient noise as much as possible, and we need to preprocess the foreground image. In addition, in order to make the outline of the moving objects clear and complete, we try to fill the void of the foreground image to avoid the object “broken” into two or more objects because the void is too large. After getting the complete foreground object outline, we get the bounding rectangle of the foreground object. Then we match the resulting bounding rectangle to the existing list of tracks to update the track list information. We get all the trajectories in the video by performing above operations for foreground images to be detected of each frame in the chronological order. The general steps of the spatial-temporal neighborhood trajectory analysis algorithm are as follows:

- (1) foreground target detection.
- (2) foreground image preprocessing, access to the bounding rectangle of foreground image.
- (3) the trajectory of the target is analyzed based on the bounding rectangle of the moving object.

Foreground target detection, that is, separating the foreground target from video sequence and get its outline. Spatial temporal neighborhood trajectory analysis algorithm needs to obtain the outlines of the foreground target to calculate its position information. Then foreground target is preprocessed and the bounding rectangle is obtained by the coordinates of the extreme points on four directions of its outlines, because the foreground image of a target is a connected domain. Finally, the trajectory of the target is analyzed based on the bounding rectangle of the moving object.

#### 4.2 Foreground target spatial adjacency determination

To determine whether the two foreground targets are close to each other in space, the most direct way is to see their overlap of outlines. The computational complexity of judging the

overlap degree of irregular images directly is high because the outlines of foreground targets are often irregular. Therefore, this paper uses the bounding rectangles of the foreground target outlines to represent the foreground targets in order to simplify the model. After pre-processing the foreground images, we have obtained the targets rectangle of the foreground outlines, which can be used to determine the spatial adjacency of the foreground targets.

The spatial adjacency of the foreground objects must satisfy two constraints at the same time, which is the size constraint and the overlapping area constraint. If the bounding rectangles of the two foreground targets are  $A$  and  $B$ , respectively. The length and breadth of the rectangle  $A$  are  $W_A$  and  $H_A$ , respectively. The length and breadth of the rectangle  $B$  are  $W_B$  and  $H_B$ , respectively. If both of the following two formulas are satisfied, the rectangle  $A$  and the rectangle  $B$  are similar in size and satisfy the size constraint.

$$0.5 \leq \frac{W_A}{W_B} \leq 2 \quad (1)$$

$$0.5 \leq \frac{H_A}{H_B} \leq 2 \quad (2)$$

The meaning of the above two formulas is: In two neighbor frames, the bounding rectangle's length of one of foreground targets must be 0.5 to 2 times of another. In addition to size similarity, the neighbor target's outlines have position overlap constraints. If the area of rectangle  $A$  and  $B$  are  $S_A$  and  $S_B$ , respectively, the area of overlap is  $S_{AB}$ . The area constraint formula is as follows:

$$0.65 < \frac{S_{AB}}{S_A} + \frac{S_{AB}}{S_B} < 0.9 \quad (3)$$

If the rectangles  $A$  and  $B$  satisfy formulas (1), (2) and (3) at the same time, the two rectangles represent the foreground objects which have spatial adjacency. If one of the formulas is not satisfied, the two foreground targets do not have spatial adjacency.

### 4.3 Trajectory tracking

The trajectory tracking is the core step of the spatial-temporal neighborhood trajectory analysis algorithm. Foreground target detection and foreground image preprocessing are the preparation steps for trajectory tracking.

#### 4.3.1 Judgment of the trajectory starts

During the execution of the algorithm, we keep a list of existing trajectories all the times. When the algorithm obtains a new frame of foreground image, the bounding rectangle of the foreground target is detected after noise reduction and the void removal. Each of these bounding rectangles is judged by the the bounding rectangles that added to existing trajectory recently. The method of judgment is described in Section 4.2 of this paper. If the bounding rectangle does not belong to the same moving target with the the bounding rectangles that added to existing trajectory recently, the bounding rectangle belongs to the new trajectory. A new track will be added in the existing track list. And the bounding rectangle's information will be stored into the structure.

### 4.3.2 Update of motion trajectory

In the process of analyzing the video, each time a new foreground image is acquired, it is need to judge whether the existing trajectory can be updated. The procedures for updating the trajectory is:

- (1) Get every bounding rectangles of the foreground target's outline that foreground image contained.
- (2) Each bounding rectangle obtained in step (1) is judged with the newly added rectangle in the trajectory that is still updated to determine whether they are spatial adjacent.
- (3) If a bounding rectangle has spatial adjacency with the newest rectangle of one of the tracks, the relevant information of the bounding rectangle is saved and corresponding trajectory is updated.

### 4.3.3 Judgment of trajectory ends

The objective reasons for the end of a trajectory is that the foreground target stops moving, or leaves the region that taken by camera. This paper judges whether a trajectory is end by the principle that if a trajectory is not updated within consecutive  $N$  frames, the trajectory is over. The setting of the threshold  $N$  needs attention. If you set  $N$  too big, it is easy to merge the different tracks together. This is because  $N$  represents the time interval that allows the foreground detection algorithm to miss the moving object. If the object on another track appears near the original path during this time interval, the two tracks are merged.

### 4.3.4 Remove the trajectory of too short time

Background targets often jitter due to the lens shaking slightly and other reasons in the long-term surveillance videos. This is not the trajectory we need to care about so we try to remove such jitter trajectory. The track caused by background jitter, which is characterized by a relatively short duration. Therefore, we need to remove the motion trajectories which have a too short duration.

When judging the end of the trajectory, the spatial-temporal neighborhood trajectory analysis algorithm will indicates that a complete trajectory has been obtained. When the algorithm is executed to this step, it is necessary to make a judgment on the duration of the trajectory. If the duration is too short, it means that the trajectory is invalid and the trajectory should be removed. In this paper, the duration threshold is set to 2 seconds, that is, the tracks whose duration are less than 2 seconds, are not effective trajectories we concerned about.

## 5 Systematic indicators

Video abstract system need to meet the requirement of accuracy and real-time. Multi-target tracking is the key technology of video abstract system. The accuracy and computational complexity of multi-target tracking affect the accuracy and computational complexity of video abstract system directly. However, video abstract system is not sensitive to indicators such as tracking precision. Therefore, in this paper, the accuracy of video abstract system is verified by trajectory detection rate and trajectory tracking average continuity, and the real-time is verified by the speed of the video abstract processing.

## 5.1 Trajectory detection rate

The trajectory detection rate refers to the number of tracks detected by the algorithm correctly divided by the number of actual motion events in the video. The actual trajectory in the video is the trajectory that starts from the stationary target begins move to it stops move, or target outside the video moves into the view, then gets away from the view. The trajectory detection rate is calculated as follows: If the number of actual motion events in the video is  $N$ . The the number of tracks the system detects correctly (removal of false positive, missed) is  $M$ . Then the detection rate  $R$  is calculated as follows:

$$R = \frac{M}{N} \quad (4)$$

## 5.2 Trajectory tracking average continuity

The trajectory tracking continuity refers to the ratio of the duration of the track detected by the system to the actual movement duration of the object. If the actual start frame of a track is  $F_{s,i}$ , the end frame is  $F_{e,i}$ , the corresponding event start frame system detected is  $F'_{s,i}$  and the end frame is  $F'_{e,i}$ . Then the duration of the trajectory  $P_i$  is calculated as follows:

$$P_i = \frac{F'_{s,i} - F'_{e,i}}{F_{e,i} - F_{s,i}} \quad (5)$$

If the total number of tracks in the video detected by the system is  $M$  and the duration of each track is  $P_i$  The trajectory tracking average continuity  $P_{avg}$  is

$$P_{avg} = \frac{\sum_1^M P_i}{M} \quad (6)$$

## 5.3 Video abstract processing speed

Video abstract processing speed refers to the number of frames processed per second when the video abstract system processes a video. Assume that the number of total frames is  $F$  and the total processing time is  $T$ , the video abstract processing speed  $S$  is calculated as:

$$S = \frac{F}{T} \quad (7)$$

# 6 Experiment

## 6.1 Introduction to experimental data

The video abstract system in this paper are for scenes taken by fixed cameras. The experimental data used in this paper are: a section of pedestrian video in CDTS 2014(An Expanded Change Detection Benchmark Dataset), a section of road video in CDTS 2014, a section of the high-definition(HD) traffic surveillance video, school sidewalk surveillance video, and school laboratory surveillance video. The main moving objects in these videos are pedestrians and vehicles. CDnet 2014 is an international foreground target detection data set; high-definition traffic surveillance video is downloaded from the Internet; school sidewalk and laboratory video is taken in South China University of Technology. Figure 1 is a frame in a pedestrian data set in CDnet 2014, Fig. 2 is a frame of the road data set in CDnet 2014,





**Fig. 1** Pedestrian data set in CDnet 2014

Fig. 3 is a frame of a traffic surveillance video, Fig. 4 is a frame of the sidewalk surveillance video taken at school, and Fig. 5 is a frame of the school lab surveillance video.

To compare with the spatial-temporal neighborhood trajectory analysis algorithm in this paper, the classical particle filter tracking algorithm [7] that used commonly in engineering is used. As well as the discrete-continuous multi-target tracking algorithm which is authoritative in the tracking field, and kernelized correlation filters(KFC) tracking algorithm. We take experiment with the data sets described above.

## 6.2 Result and discussion

### 6.2.1 Trajectory detection rate

This paper uses the artificial way to count the movement event in the video first, then uses the systems to analyze the video and get a list of motion events. Since the particle filter



**Fig. 2** Road data set in CDnet 2014



**Fig. 3** Traffic surveillance video

algorithm needs to mark the starting frame of the trajectory and the target to be tracked manually, the trajectory detection rate experiment is not performed for it. The results of the detection rates are shown in Table 1.

From the data in Table 1, the spatial-temporal neighborhood trajectory analysis algorithm has 80% or higher trajectory detection rate for each video data, as well as the KCF algorithm. The discrete-continuous optimization of the multi-target tracking algorithm's trajectory detection rates are relatively low, especially on highway high-definition video surveillance, the detection rate is only 40%. This is related to the characteristics of the data set in this paper. This article studies the surveillance video. The moving targets in these videos are mostly move from far to near or from near to far (targets of highway HD video move from far to near particularly obvious). The targets' sizes change slowly. Therefore, the center point position trajectories of the objects are not smooth. Discrete-continuous multi-target tracking algorithm is based on the center point of the moving object for trajectory



**Fig. 4** Sidewalk surveillance video taken at school



**Fig. 5** School lab surveillance video

analysis, which results in low trajectory rate, while the spatial-temporal trajectory analysis algorithm is based on the overlap ratio of adjacent rectangles to analyze whether they belong to the same trajectory that not affected by the change of targets' sizes. From the perspective of track detection rate, the spatial-temporal neighborhood trajectory analysis algorithm is more suitable for dealing with surveillance video.

Spatial-temporal neighborhood trajectory analysis algorithm also has some shortcomings in the trajectory detection. The algorithm can detect targets passing through the video with uniform speed correctly, such as pedestrians and vehicles. But for the targets which move particularly fast, the system can not detect its trajectory.

As shown in Fig. 6, because the surveillance camera is far from the road, the speed of vehicles move in the video is not very fast. In this case, the system can detect the vehicle's trajectory correctly. But the bike in the left of Fig. 7 is closer to the camera. Its own

**Table 1** Result of trajectory detection rate

Video name	Number of Trajectories	Spatial-temporal Neighborhood Algorithm Trajectory Detection Rate	Discrete-continuous Multi-target Tracking Trajectory Detection Rate	Kernelized Correlation Filters Trajectory Detection Rate
CD net2014-Pedestrian video	5	80.0%	60.0%	100%
CD net2014-Rode video	30	93.3%	86.6%	90%
High-definition Traffic surveillance Video	15	93.3%	40.0%	93.3%
School sidewalk video	80	91.3%	77.5%	87.5%
School laboratory Video	89	89.8%	56.2%	87.6%



**Fig. 6** Trajectory detection situation: detect trajectory correctly

movement speed is particularly fast. From the figure you can see the shooting blurred phenomenon caused by the high speed of the bike. The overlap degree of two bicycles' outlines which spatial-temporal neighborhood analysis algorithm gets in the two adjacent detection time can not reach the set threshold. Thus, the system will judge that the outlines does not belong to the same moving targets and resulting in lost of traces. To solve this situation, we need to reduce the overlap degree of the threshold or decrease the interval of algorithm detection.

In summary, if you want to achieve a better track detection rate, you need to understand the speed of moving objects in the video first, then use this as the basis for adjusting threshold of overlap degree for the spatial-temporal neighborhood trajectory analysis algorithm.



**Fig. 7** Trajectory detection situation: not detect trajectory correctly

### 6.2.2 Trajectory tracking average continuity

This paper counts the start and end frames of the motion trajectory in the video first, and then analyze the video once to get the track list. The results of the spatial-temporal neighborhood algorithm, particle filter algorithm, discrete-continuous optimization multi-target tracking algorithm and DCF are shown in Table 2. They are calculated by the trajectory continuity formula as described in Section 5.

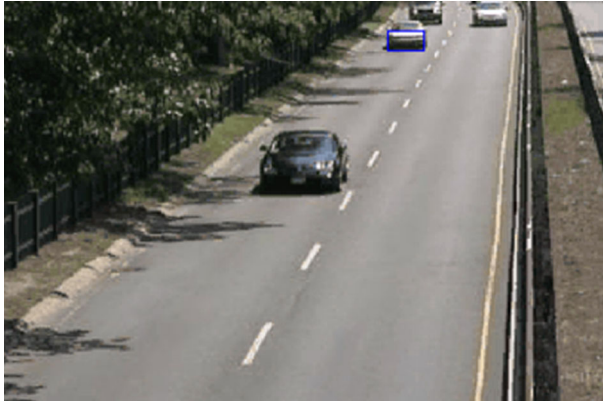
From the data in Table 2, the trajectory tracking average continuity of the spatial-temporal neighborhood trajectory analysis algorithm is 83.8%, which proves that for most of the surveillance videos, our system can track good trajectories.

In addition, spatial-temporal neighborhood trajectory analysis algorithm, particle filter algorithm and KCF algorithm have the relatively close degree of tracking continuities. Discrete - continuous optimization of multi - target tracking algorithm has low tracking continuities. The particle filter algorithm is particularly low for the school sidewalk surveillance video while the tracking continuities of other four videos are all more than 80%. This is because the particle filter algorithm is tracking based on the target color but the color differences of moving targets and the background colors are relatively small in that video. Discrete-continuous optimization of multi-target tracking algorithm's tracking continuities are not high, which is still caused by the size changes of the targets from the far to near. This algorithm has a higher degree of tracking continuities of moving objects in the distant, while the tracking continuity is low for the moving objects near the camera (the change of size is more obvious). The spatial-temporal neighborhood trajectory analysis algorithm not only is not affected by the target color, but also can adapt to the size changes of the targets which cause the trajectory tracking continuities of it for the surveillance videos are relatively high.

However, the tracking continuities of the spatial-temporal neighborhood trajectory analysis algorithm can not reach 90%, which still has space for improvement. Typical examples are shown below:

**Table 2** Results of trajectory tracking continuity

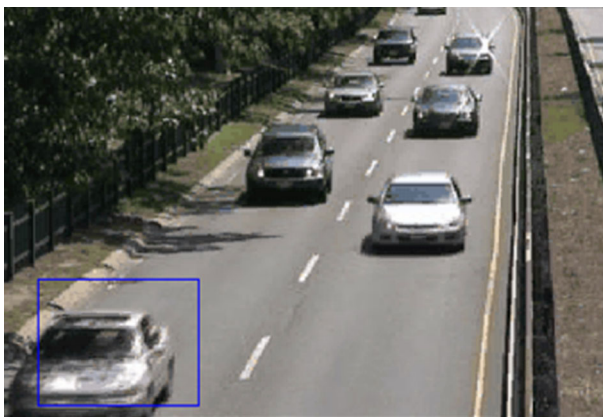
Video name	Spatial-temporal Neighborhood Algorithm Trajectory Tracking Continuity	Particle Filter Algorithm Trajectory Tracking Continuity	Discrete-Continuous Multi-target Tracking Algorithm Trajectory Tracking Continuity	Kernelized Correlation Filters Trajectory Tracking Continuity
CD net2014-Pedestrian video	79.3%	82.1%	58.5%	82.3%
CD net2014-Rode video	86.7%	79.8%	78.5%	84.2%
High-definition Traffic surveillance Video	83.2%	86.1%	52.3%	80.7%
School sidewalk Vvideo	82.4%	52.7%	71.2%	78.4%
School laboratory Video	87.5%	93.4%	61.0%	82.6%



**Fig. 8** A typical example of a track that can not be tracked continuously: start frame of trajectory that can not be tracked continuously

The foreground detection method of the spatial-temporal neighborhood trajectory analysis algorithm adopts the hybrid Gaussian background modeling algorithm, which needs to model the background of each pixel of the image. Therefore, when the foreground target enters the view, it needs a delay to detect it. As shown in Fig. 8, the system detects the trajectory start frame, in fact the target has been moved for some distant in the view. When the target leaves the view, as shown in Fig. 9 the two adjacent outlines' degree of overlap is reduced slowly, and the spatial-temporal neighborhood trajectory analysis algorithm is mistaken because they not belong to the same target trajectory.

In order to improve the continuity of trajectory tracking, we need to adopt a more sensitive foreground detection algorithm. At the same time, we need to improve the spatial-temporal neighborhood trajectory analysis algorithm to solve the problem.



**Fig. 9** A typical example of a track that can not be tracked continuously: end frame of trajectory that can not be tracked continuously



**Table 3** Results of video abstract processing speed

Video name	Video Resolution	Spatial-temporal Neighborhood Algorithm (frames per second)	Particle Filter Algorithm (frames per second)	Discrete-Continuous Multi-target Tracking Algorithm (frames per second)	Kernelized Correlation Filters (Frames per second)
CD net2014-Pedestrian video	320*240	223.5	186.5	25.3	180.73
CD net2014-Rode video	352*288	187.4	153.4	21.1	150.85
High-definition Traffic surveillance Video	1280*640	36.5	26.4	7.1	32.11
School sidewalk video	640*480	102.6	82.5	12.3	88.97
School laboratory Video	1280*640	42.5	25.1	6.5	37.23

### 6.2.3 Video abstract processing speed

As shown in Table 3, the particle filter algorithm and KCF are slower than the spatial-temporal neighborhood trajectory analysis algorithm. Although particle filter algorithm can meet the real-time requirements, it can not handle the surveillance video totally automatically. The processing speed of discrete - continuous optimization of multi-target tracking algorithm is slower than the others, can not meet the real-time requirements completely. As seen from Table 3, the speed of video abstract processing becomes slower with the higher video resolution. The lowest speed that spatial-temporal neighborhood trajectory analysis algorithm has is for the high-definition traffic surveillance video, which is 36.5 frames per second, but it still can meet the requirements of real-time processing (30 frames per second). The test environment is a personal computer. If the system is deployed in the server for video abstract processing, there is no doubt that the standard for dealing with real-time video can be achieved.

## 7 Conclusions

Spatial-temporal neighborhood trajectory analysis algorithm is introduced and used on video abstract system in this paper. The algorithm selects the existing foreground detection algorithm which is most suitable for processing the surveillance video data as the foreground detection step, and extracts the trajectories of the targets on the basis of it. By tracking the spatial-temporal neighboring moving objects, we can get their whole trajectories. Experiments on five different surveillance video data sets demonstrate the algorithm is effective for surveillance videos, with sufficient processing speed and accuracy.

**Acknowledgements** This work is supported by National Natural Science Foundation of China (61370102, 61370185), Guangdong Natural Science Funds for Distinguished Young Scholar (2014A030306050), the

Ministry of Education - China Mobile Research Funds (MCM20160206) and Guangdong High-level personnel of special support program(2014TQ01X664).

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Al-Ayyoub M, Alzu'Bi S, Jararweh Y, Shehab MA, Gupta (2016) Accelerating 3d medical volume segmentation using gpus. *Multimedia Tools & Applications*, pp 1–20
2. Alsmirat MA, Jararweh Y, Obaidat I, Gupta (2016) Automated wireless video surveillance: an evaluation framework. *J Real-Time Image Proc* 1–20
3. Bae SH, Yoon K (2017) Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Trans Pattern Anal Mach Intell* PP(99):1–1
4. Chari V, Lacoste-Julien S, Lapedes I, Sivic J (2015) On pairwise costs for network flow multi-object tracking. In: *IEEE conference on computer vision and pattern recognition*, pp 5537–5545
5. Gupta BB, Agrawal DP, Yamaguchi S (2016) Handbook of research on modern cryptographic solutions for computer and cyber security [M]
6. Hanjalic A, Zhang H (1999) An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Trans Circuits Syst Video Technol* 9(8):1280–1289
7. Hue C, Cadre JPL, Perez P (2006) Tracking multiple objects with particle filtering. *IEEE Trans Aerosp Electron Syst* 38(3):791–812
8. Li J, Li J, Chen X, Jia C, Lou W (2015) Identity-based encryption with outsourced revocation in cloud computing. *IEEE Trans Comput* 64(2):425–437
9. Li P, Li J, Huang Z, Gao CZ, Chen WB, Chen K (2017) Privacy-preserving outsourced classification in cloud computing. *Clust Comput* 1–10
10. Li J, Yu C, Gupta BB, Ren X (2017) Color image watermarking scheme based on quaternion hadamard transform and schur decomposition. *Multimedia Tools & Applications*, pp 1–17
11. Li J, Zhang Y, Chen X, Xiang Y, Li J, Zhang Y et al (2017) Secure attribute-based data sharing for resource-limited users in cloud computing. *Comput Secur* 72
12. Lin L, Lu Y, Pan Y, Chen X (2012) Integrating graph partitioning and matching for trajectory analysis in video surveillance. *IEEE Trans Image Process Publ IEEE Signal Process Soc* 21(12):4844–4857
13. Lipowski A, Lipowska D (2012) Roulette-wheel selection via stochastic acceptance. *Physica A: Stat Mech Its Appl* 391(6):2193–2196
14. Liu N, Wu H, Lin L (2014) Hierarchical ensemble of background models for ptz-based video surveillance. *IEEE Trans Cybern* 45(1):89–102
15. Milan A, Leal-Taixe L, Reid I et al (2016) MOT16: A Benchmark for Multi-Object Tracking
16. Ng PE, Ma KK (2006) A switching median filter with boundary discriminative noise detection for extremely corrupted images. *IEEE Trans Image Process Publ Signal Process Soc* 15(6):1506–16
17. Roth S (2012) Discrete-continuous optimization for multi-target tracking. In: *Computer vision and pattern recognition*, vol 157. IEEE, pp 1926–1933
18. Shi X, Ling H, Xing J, Hu W (2013) Multi-target tracking by rank-1 tensor approximation. *Comput Vis Pattern Recognit* 2387–2394
19. Truong BT, Venkatesh S (2007) Video abstraction: a systematic review and classification. *ACM Trans Multimed Comput Commun Appl* 3(1):3
20. Vaidya J, Shafiq B, Basu A, Hong Y (2013) Differentially private naive bayes classification. In: *Ieee/wic/acm international joint conferences on Web intelligence*, vol 01. IEEE, Piscataway, pp 571–576
21. Varadarajan S, Miller P, Zhou H (2015) Region-based mixture of gaussians modelling for foreground detection in dynamic scenes. *Pattern Recogn* 48(11):3488–3503
22. Wang X, Türetken E, Fleuret F et al (2014) Tracking interacting objects optimally using integer Programming[J]. *Comput Vis - Eccv Pt I 2014(8689)*:17–32
23. Yang Y, Zhang D (2003) A novel line scan clustering algorithm for identifying connected components in digital images. *Image Vis Comput* 21(5):459–472
24. Yu C et al (2017) Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. *Multimedia Tools & Applications*, pp 1–24



25. Zhang X, Yan X (2015) A new statistical precipitation downscaling method with Bayesian model averaging: a case study in China. *Clim Dyn* 45(9):1–15
26. Zhang S, Wang J, Wang Z, Gong Y, Liu Y (2015) Multi-target tracking by learning local-to-global trajectory models. *Pattern Recogn* 48(2):580–590



**Han Huang** PhD, Professor at School of Software Engineering, South China University of Technology (SCUT). He is currently Director of Intelligent Algorithm and Intelligent Software Studio, Director of SCUT-CUHK (City University of Hong Kong) Logistics Optimization Researcher Center. He is also the principal investigator of a number of research funding, among which includes Outstanding Young Natural Science Funding of Guangdong Province, Chinese Natural Science Foundation, and Pearl River Research Star Project, etc. He is also the awardee of the Guangdong Natural Science Award (Second Prize). His software studio has successfully undertaken intelligent software development projects for a number of institutions and corporations. Prof. HUANG's research interests include theories and practice of computational intelligence. He has published more than 50 research articles in journals and conference proceedings home and abroad, among which include *Information Science*, *IEEE Transaction on Cybernetics*, *Science in China*, *Journal of Computer*, *Proceeding of GECCO* and *Proceeding of CEC*. He has applied for 21 national patents, among which six have been endorsed by the Chinese Bureau of Patent, and five published. In addition, he has held 26 cases of software copyright, and more than 30 instances of his intelligent algorithm and software have been successfully applied in industry.



**Shen Fu** born in 1995, MRS, Student of School of Software Engineering, South China University of Technology (SCUT).



**Zhao-Quan Cai** born in 1970, MR, Professor at Huizhou University, CCF member, his current research interests include computer networks, intelligent computing and database.



**Bin Li** born in 1991, MR, Student of School of Software Engineering, South China University of Technology (SCUT).