J Vis (2022) 25:635–651
https://doi.org/10.1007/s12650-021-00803-w

REGULAR PAPER

Guodao Sun · Tong Li · Ronghua Liang

SurVizor: visualizing and understanding the key content of surveillance videos

Received: 8 July 2021 / Revised: 8 July 2021 / Accepted: 23 September 2021 / Published online: 9 November 2021

Abstract With the rapid development of society, video surveillance has progressively expanded into different areas of life, such as transportation, security inspection, banks. There are a large number of replaced and newly deployed cameras in fields such as safe cities, smart campuses and smart buildings, which leads to a huge amount of video data, slow retrieval speed in video examining, and low efficiency in understanding complete picture of videos. In this paper, we propose SurVizor, a visual analysis system to understand the key content of surveillance videos. We integrate multiple image features and employ time series analysis methods to explore key temporal patterns in the feature. We integrate multiple visualization views from three levels of video, feature, and frame to promote exploration, analysis and understanding of video content. We evaluate the proposed system through a case study based on real-world surveillance videos from multi-camera and a user study. The results demonstrate the usability and effectiveness of our system in analyzing and understanding the key content of surveillance videos.

Keywords Surveillance video · Multi-feature · Time series · Visual analysis

1 Introduction

With the rapid growth of national economy and the rapid progress of society, there is an increasing demand for security protection and on-site inspection in various fields. Video surveillance has been widely used in all aspects of production and life. Meanwhile, with the continuous upgrading and transformation of smart city projects, new smart cities and smart towns are constantly emerging, there are a large number of replaced and newly deployed cameras in fields such as safe cities, smart transportation, smart campuses and smart buildings (Alabdulatif et al. 2018; Alshammari and Rawat 2019). The ultimate goal of intelligent video surveillance technology is to turn cameras into human eyes. The sequence of images obtained from the camera undergoes intelligent analysis, which primarily includes object detection (Liu et al. 2020), object tracking, object re-identification, and object behavior analysis to understand the content of surveillance videos.

Given today's massive video surveillance data, the challenges we face are as follows: the great amount of video data, the inability to quickly identify and analyze abnormal events in video, and the inability to quickly and effectively extract video themes. Machine learning and deep learning are currently widely used

G. Sun · T. Li · R. Liang (✉)
College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
E-mail: rhliang@zjut.edu.cn

G. Sun
E-mail: godoor.sun@gmail.com

T. Li
E-mail: 1332272434@qq.com

in image processing and computer vision (Yuan et al. 2021; Liu et al. 2017), and have achieved fruitful results. However, the challenge is that there is no mature visual analysis system to support hierarchical and effective presentation of the results, as well as the exploratory research.

In response to the above challenges, we proposed SurVizor, a visual analysis system that integrates multi-feature of video frames and analyzes temporal information of features. From three levels of video, feature and frame, multiple visual analysis views are integrated to understand video content and quickly locate the key content. In terms of video context analysis, we map video frame information by a variety of visual analysis methods to help users rapidly understand the key content of videos. In terms of image feature analysis, we investigate related work in video summary domain including aesthetics, image quality, memory assessment, and anomaly assessment to evaluate the importance of frames. In terms of temporal information analysis, we build two-dimensional time series data into a network structure by the Markov transition field and characterize transition of temporal information by the community division. Moreover, for the problem of high redundancy between video frames, we simplify feature series and evaluate the importance of points by identifying perceptually important points, and then build a binary search tree to deliver video summarization.

In short, the contributions of this paper could be summarized as follows:

- We study the key content of surveillance videos from the perspective of multi-feature integration and time series analysis.
- We propose SurVizor, an interactive visual analysis system that retrieves, compares and analyzes the surveillance video content from three levels.
- We introduce a multi-camera-based case study to evaluate the usability and effectiveness of the visual analysis system with respect to event consistency and location difference.

2 Related work

Video Summary. We divide static video summary work into two branches: visual feature-based and semantic information-based. The first branch generally selected visual-related features, such as color, image quality, motion cues, attention to evaluate the importance of frames. In recent years, some work has combined various features for video summary. Gygli et al. (2014) combined attention, aesthetics/quality, face detection, etc. to calculate interest scores. Hu et al. (2017) took image quality as the basis for evaluating the importance of video frames and added other low-level features. The second branch is based on semantic information to reduce the semantic gap between generated videos and raw videos. Wei et al. (2018) extracted the appropriate number of video shots by minimizing the difference between descriptive sentences and artificially annotated text.

In recent years, the analysis of images and videos in the field of computer vision has achieved fruitful results. However, there is no more prominent work that employs visual analysis methods to assist analysis tasks. The interpretability of related models and the understanding of visualized video content are lacking.

Video Visualization. In recent years, visual analytics has been gradually mature and widely applied in many applications (Sun et al. 2013; Wang et al. 2021; Weng et al. 2021; Ye et al. 2021). The difficulty of video analysis lies in the great amount of video data and low efficiency in understanding the complete picture of videos. With the help of visual analysis, the comprehension of video content can be better improved. Further research has been carried out into video content and semantic information with the help of various visualization forms and rich interactive functions. In terms of practical applications, (Chan et al. 2019) combined muscle signal data and video content to analyze the movement of patients with injured brachial plexus to help physicians' diagnosis and patients' rehabilitation training. There is also some of the work which focuses on semantic information in videos. Wu and Qu (2018) employed action recognition technology to focus on analyzing the speaker's body expression and language expression, and discussed the speaker's speech technology. Zeng et al. (2019) combined facial emotions and speech content, and emphasized the emotional coherence of facial and language expression.

The above work ignores the correlation between image features and video content. We employ multi-feature integration and time series analysis methods to assist in understanding surveillance video content.

Time Series Analysis. In view of the high-dimensional characteristics of time series, some work focuses on sub-series and series simplification. Douglas and Peucker (1973) proposed a technique to reduce the number of points. Chung et al. (2001) formally proposed the perceptually important point (PIP) method.

Other work focuses on time series similarity and clustering. Liao (2005) summarized methods for measuring similarity: Pearson Coefficient, Dynamic Time Warping, Short Time-series Clustering and other methods. They also summarized clustering methods into three types: raw data-based, feature-based, and model-based. In addition, some research work focuses on the time attribute and builds time series models to realize time series classification. Cui et al. (2016) proposed the Multi-Scale Convolutional Neural Network model for the problem that the different features existing in different time scale series cannot be extracted. Liu and Wang (2016) and Cheng et al. (2020) built two-dimensional time series data into a network structure.

The above work involves the analysis of time series on single-dimensional data. Based on video data, we model the multi-feature as time series and analyze temporal information of frames.

3 Task analysis and system pipeline

3.1 Task analysis

We summarize our analysis tasks by researching work (Table 1) in related fields. The existing work about video research can be summarized into two levels: content-based analysis and feature-based analysis. In terms of content-based analysis, given the long time-consuming browsing and the difficulty in extracting themes, some work has devoted to video indexing, retrieval and summary. In terms of feature-based analysis, some work has conducted in-depth research based on basic features such as video image features, temporal information, and audio information. To assist users in analyzing the key content of surveillance videos, we introduce the task analysis from two aspects: data processing and visual analysis. The Hierarchical Task Analysis is shown in Fig. 1.

- **Data processing tasks (T1)** can be summarized to two aspects: data collection and feature acquisition. **T1.1 To collect the raw data.** To prepare for analysis, it is necessary to learn from related work in the field of computer vision and collect surveillance videos. **T1.2 To acquire the feature.** Image features can be measured from various aspects. It is necessary to determine the selection of image features based on the collected surveillance video data, and obtain these features through effective methods.
- **Visual analysis tasks (T2)** can be summarized to three level: video-level **(T2.1)**, feature-level **(T2.2)** and frame-level **(T2.3)**. **T2.1.1 To summarize the video information.** Our work evaluates frames from many aspects. It is necessary to integrate multi-feature to characterize video information, provide an overview of video content and guide users for further exploration. **T2.1.2 To provide video context for the analysis.** In addition to the video's overview information, it is necessary to provide video context in order to retrieve raw content and provide factual verification. **T2.2.1 To present the feature information.** As a basic unit of exploration and analysis, the feature is necessary for detailed presentation and analysis. **T2.2.2 To compare and analyze multi-feature associations.** The different evaluation standards will lead to differences between feature values. However, these features represent the same video frame, and there will be some degree of correlation. Therefore, it is necessary to conduct a comparative analysis between the features. **T2.3.1 To analyze temporal information of the feature.** With the development of video events, there are some changes in the feature value. Therefore, it is necessary to pay attention to temporal information and to study the impact of events on features.

Table 1 Typical related references for hierarchical task analysis

Reasearch	T1.1	T1.2	T2.1.1	T2.1.2	T2.2.1	T2.2.2	T2.3
Chan et al. (2019)	✓	–	–	✓	✓	–	✓
Gygli et al. (2014)	✓	✓	✓	–	–	–	–
Hu et al. (2017)	✓	✓	✓	–	–	–	–
Sun et al. (2021)	✓	✓	–	✓	✓	–	–
Wu and Qu (2018)	✓	–	–	✓	–	–	✓
Zeng et al. (2020)	✓	–	–	✓	—	–	✓
Zeng et al. (2019)	✓	–	✓	✓	–	—	✓

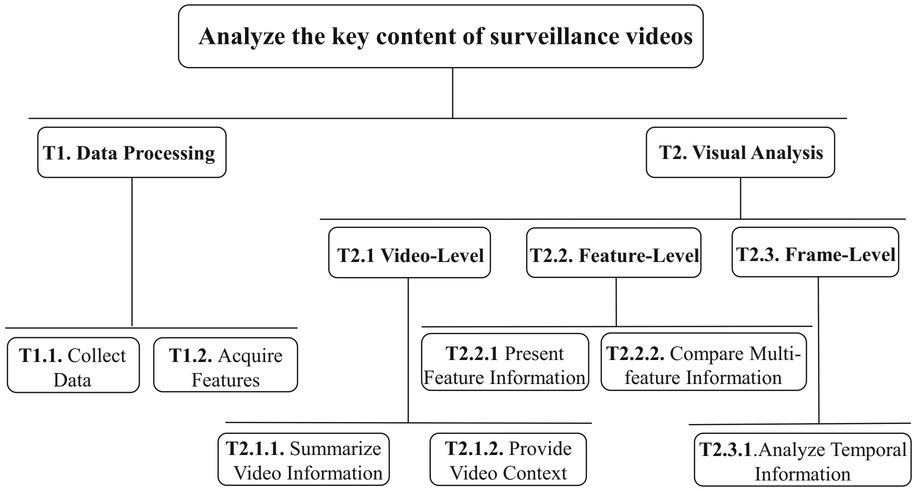


Fig. 1 Hierarchical task abstraction of SurVizor. Each box represents a task or subtask

3.2 System pipeline

The pipeline of SurVizor is shown in Fig. 2. At the data processing phrase, we first collect the raw video data and split them into a set of video frames. Then, we obtain four features of frames through models: Aesthetics, Quality, Memory, and Anomaly. Detailed information can be found in Sect. 4. Subsequently, at the visual analysis phrase, we design and implement visual analysis based on tasks at the video-level, feature-level and frame-level. Detailed information can be found in Sect. 5.

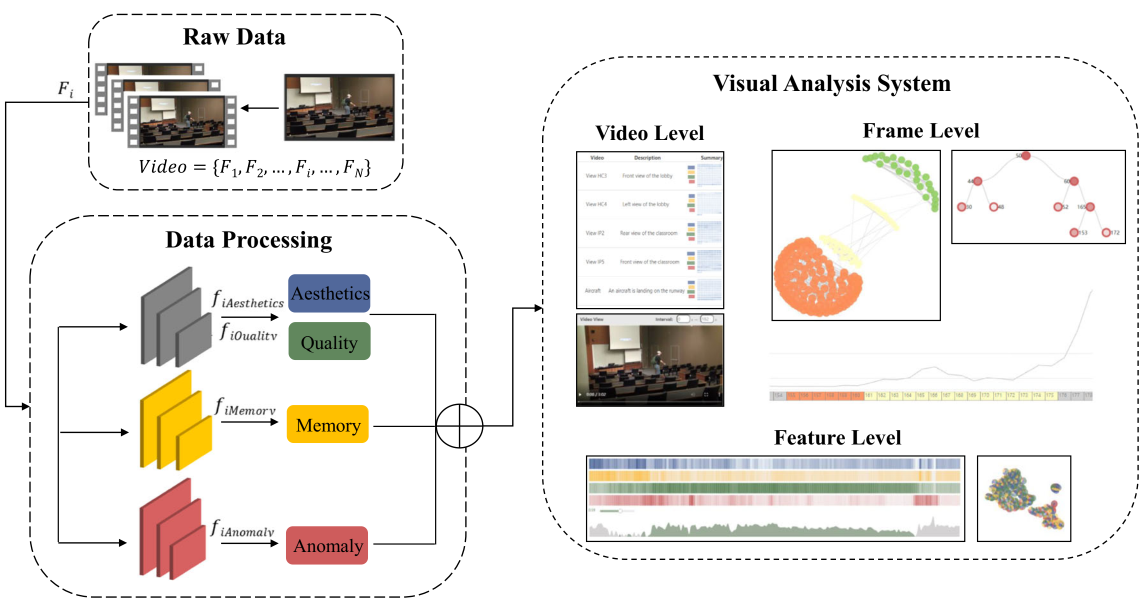


Fig. 2 Our system pipeline for key content analysis of surveillance videos. At the data processing phrase, we collect raw video data and obtain multiple features. At the visual analysis phrase, three-level views (video, feature and frame level) are provided to support exploration

4 Data processing and description

4.1 Data processing

We conduct a series of data processing steps. Firstly, we extract video frames from raw video data (one frame per second), and then apply models to extract feature information (**T1.2**) of frames from various aspects. Finally, we integrate these features and align them in time. In this paper, we choose four features: Aesthetics, Image Quality, Memory and Anomaly. Aesthetics quantifies the semantic level features related to emotion and beauty, which is highly correlated with human perception (Talebi and Milanfar 2018) and helps to retrieve segments people are interested in. Quality quantifies the pixel-level degradation problems such as noise, blur, compression, distortion which is directly related to image content and movement (Talebi and Milanfar 2018) and helps to retrieve the scene change or the object movement. Memory is associated with visual factors, the intrinsic properties of images and saliency (Bylinskii et al. 2015), which helps to retrieve impressive scenes. Anomaly is highly correlated with object behaviors that do not meet expectations (Chandola et al. 2009), which helps to retrieve segments of abnormal events. The four features can help us analyze the key content of videos from different aspects.

Aesthetics Feature and Image Quality Feature Extraction: We employ the model proposed by Talebi and Milanfar (2018) to assess the aesthetic quality and technical quality of video frames. Most of the evaluation methods are only to predict the average score of the dataset, but Talebi et al. predicted the distribution of scores and employed Earth Mover's Distance as a loss function to get a more accurate average score.

Memory Feature Extraction: We employ the model AMNet proposed by Fajtl et al. (2018) to realize the memory assessment of video frames. Some work based on global image features GIST, SIFT, HOG, SSIM studied the factors that produce the image memory effect, analyzed the relationship between memory and various visual factors and saliency, or employed deep learning to predict memory. Fajtl et al. first studied the application of a deep learning method with visual attention mechanism and recurrent network in learning and predicting image memory, which significantly improved the performance of image memory learning and reasoning.

Anomaly Feature Extraction: We employ the model proposed by Liu et al. (2018) to realize the anomaly assessment of video frames. Most solutions based on deep learning employ the AutoEncoder structure model: reconstruct the current video frame and detect anomalies based on the reconstruction error. However, the AutoEncoder has a strong reconstruction capability, and may still be able to reconstruct an abnormal image well and output a smaller reconstruction error. Therefore, Liu et al. believe that anomaly detection should be considered from the perspective of prediction. They employed Conditional Generative Adversarial Nets to build a video frames prediction model, and combined with optical flow to constrain the generator. Compared with Autoencoder, the effect is greatly improved.

4.2 Data description

We evaluated our methods on three public data sets: SumMe (Gygli et al. 2014), CAMPUS (Xu et al. 2016) and SALSA (Alameda-Pineda et al. 2015) (**T1.1**). The SumMe consists of 25 videos that cover egocentric, static, and moving videos. The CAMPUS consists of 16 videos that cover four scenes, namely Garden 1, Garden 2, Auditorium, and Parking Lot. Each scene is shot by 3-4 high-quality DV cameras and each camera covers both overlapping regions and non-overlapping regions with other cameras. The SALSA was recorded in a regular indoor space. It consists of 8 videos and the captured social event involved 18 subjects over 60 minutes. After data processing, each video can be described as:

$$\text{Video} = \{F_1, F_2, F_3, \ldots, F_i, \ldots, F_N\} \tag{1}$$

where $Fi = \left\{f_{\text{iAesthetics}}, f_{\text{iQuality}}, f_{\text{iMemory}}, f_{\text{iAnomaly}}\right\}$.

That is, given a video *Video*, we represent it as a series of frames, where F_i represents the i-th frame in *Video*, and N represents the total number of frames in *Video*. F_i is characterized by four features: $f_{\text{iAesthetics}}$, f_{iQuality}, f_{iMemory}, and f_{iAnomaly}.

5 Visual design and implement

We research related work (Table 2) about time series, multi-feature analysis, and video analysis in the field of visual analysis. In term of time series data, existing work mostly uses line chart, bar chart, stream graph, etc., and a few work uses network graph to represent time series data. In term of multi-feature data, existing work usually designs new glyphs to represent data. In term of video analysis, existing work designs video players to provide context.

5.1 Video overview

In this part, we first design four labels to represent four features: represents *Aesthetics* feature, represents image *Quality* feature, represents *Memory* feature, represents *Anomaly* feature.

A video usually contains a large number of frames and frame information. Therefore, it is critical to provide visualization techniques that can summarize feature information to help users identify a video of interest and narrow down the search space (**T2.1.1**). We designed the video list which contains three aspects of the information: the name of the video, a brief description of the video context, and the summary of features mapped visually. As shown in (Fig. 3A), a *Summary* represents a video. We map the height of

Table 2 Typical related references for visual design

Reasearch	LineChart	BarChart	StreamGraph	Network	Glyph	VideoPlayer
Chan et al. (2019)	–	✓	–	–	✓	✓
Cheng et al. (2020)	–	–	–	✓	–	–
Lee et al. (2019)	–	–	–	–	–	✓
Sun et al. (2017)	–	–	✓	–	–	–
Sun et al. (2021)	–	✓	–	-	✓	✓
Wu and Qu (2018)	–	–	–	–	✓	✓
Zeng et al. (2020)	✓	✓	–	–	–	✓
Zeng et al. (2019)	✓	✓	–	–	✓	✓

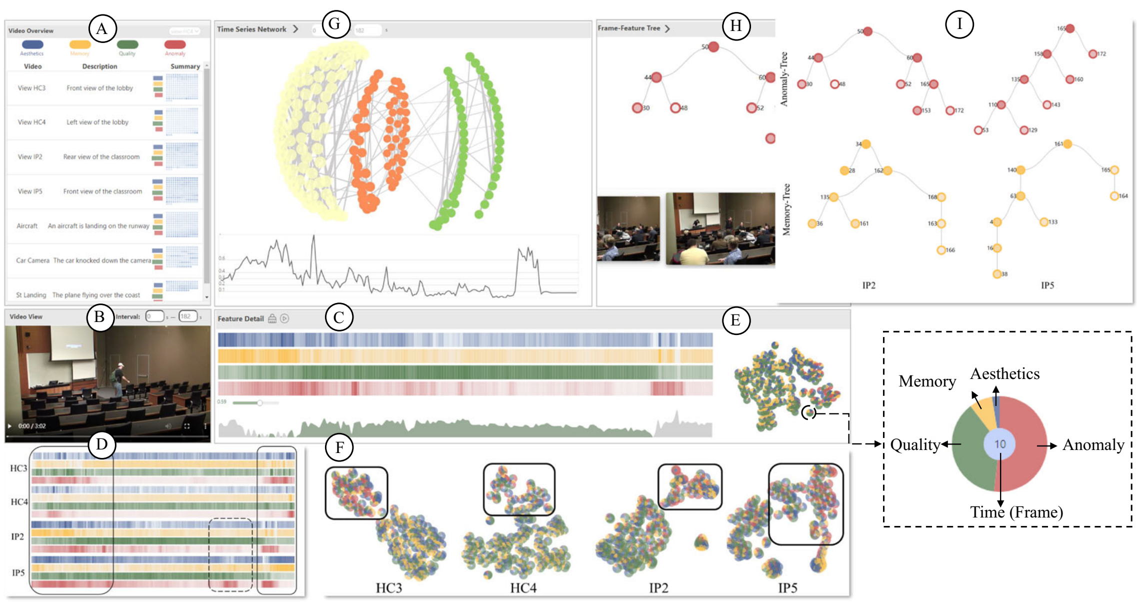


Fig. 3 The interface of visual analysis system SurVizor showing how to analyze video content from three levels. The Video Overview **a** summarizes the overview information for each video and provides guidance for users' interest exploration. The Video View **b** provides raw video context and serves as an auxiliary analysis tool. The Feature Detail **c** focuses on feature-level analysis and supports multi-feature comparison. The t-SNE View **e** displays dimensionality reduction results of the local period. The Time Series Network **g** focuses on frame (time)-level analysis and displays result of the community division and transition of feature values. The Frame-Feature Tree **h** presents the binary search tree to construct video summarization

column in bar chart by calculating the average of four features. Each cell in grid graph represents a frame in the video. We integrate four features to calculate a new value:

$$f = \alpha_{aes} f_{\text{Aesthetics}} + \alpha_{q} f_{\text{Quality}} + \alpha_{m} f_{\text{Memory}} + \alpha_{ab} f_{\text{Anomaly}} \tag{2}$$

After understanding overview of the video, users can select videos to analyze based on their interests. The corresponding video content will be displayed in Video View (Fig. 3Ⓑ) and provided evidence for subsequent research (**T2.1.2**).

5.2 Feature detail

In order to clearly show overall trend of the feature value and the comparison between different features over time (**T2.2.2**), we decided to tile and align the feature data in horizontal way. Compared to other complex visual designs, this style is a more traditional and familiar visual type to facilitate users to comprehend data.

We design the pixel bar (Fig. 3Ⓒ) to map the four features (each pixel represents a frame). In addition, we have employed two other design schemes (Fig. 4a): line chart and bar chart. However, the feature values of adjacent frames have high similarity. In the case where we focus on the critical period of feature change, using the first two schemes is likely to result in visual clutter, and the third scheme is more suitable for our data. We map the feature value to pixel color. The light color means that the feature of this frame is low, and dark color is the opposite.

Considering that pixel bar above cannot inspect the specific values, we have further designed area chart based on the work of Heer et al. (2009) for better clarity (**T2.2.1**). Users can pull the button to set the threshold (Fig. 3Ⓒ). The area chart will color the feature parts above this threshold according to our color scheme, and the feature parts below this threshold will be flipped and colored in gray (Fig. 4b).

5.3 t-SNE view

In order to help users explore the relationship between feature value and video events in detail (**T2.2.2**), we decided to employ the t-Distributed Stochastic Neighbor Embedding (t-SNE) to model the similarity of four features dimensions of frames. The local feature information that users are interested in is displayed in clusters on two-dimensional space, and frames with similar features will be clustered together. Initially, each

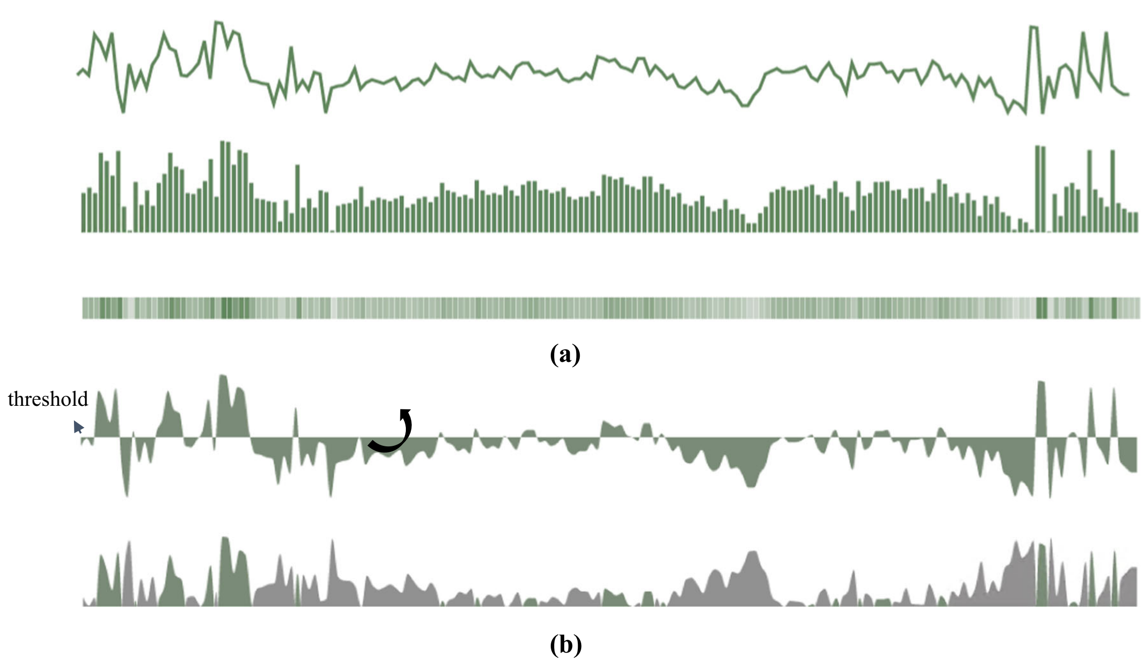


Fig. 4 The design scheme we considered for the feature detail view. **a** The design scheme for multi-feature analysis. **b** The design scheme for single feature value analysis

frame was denoted by a point. However, such a visual design loses feature information and is insufficient in providing visual guidance to identify patterns. Therefore, inspired by Zeng et al. (2019), we design a ring to represent a frame (Fig. 3Ⓔ).

Among the four features, the larger feature value will be assigned to more circular part. At the same time, in order to retain time information, we draw a circle inside the ring. The color of the circle is mapped to the frame (second), represents the beginning of the video, and represents the end of the video.

5.4 Time series network

In this part, we focus on studying temporal information of the feature, converting the time series into a network structure for analysis. We realize this idea through the Markov Transition Field (MTF). The whole process is divided into four stages (Fig. 6): quantify the time series, calculate the Markov matrix, calculate the MTF, and draw the time series network.

Quantification of the Time Series. Given a video frame series:

$$F = \{f_1, f_2, \ldots, f_t, \ldots, f_N\} \tag{3}$$

f represents a certain feature, t represents the t-th frame (t-th second), we employ the quantile method to discretize the continuous time series F into Q bins.

Breakpoints are a sorted list of numbers Bins $= q_1, q_2, \ldots, q_Q$. So far, each value f_t in time series F has been mapped to q_i.

Calculation of the Markov Matrix. After each f_t is allocated to the corresponding bin q_i, we construct a $Q * Q$ matrix W (Eq. 4) by calculate the transitions between bins in the manner of a first-order Markov chain along each time step. W_{ij} in W represents the total number Num of points in bin q_j, followed by points in bin q_i. We further normalize by $\sum_j W_{ij} = 1$ and calculate the transition probability to obtain the Markov Matrix W' (Eq. 5), where the main diagonal W_{ii} represents the self-transition probability. W'_{ij} in W' represents the frequency P of a point in bin q_j, followed by a point in bin q_i.

$$W = \begin{pmatrix} W_{11}|_{Num(f_t \in q_1 | f_{t-1} \in q_1)} & \cdots & W_{1Q}|_{Num(f_t \in q_1 | f_{t-1} \in q_Q)} \\ W_{21}|_{Num(f_t \in q_2 | f_{t-1} \in q_1)} & \cdots & W_{2Q}|_{Num(f_t \in q_2 | f_{t-1} \in q_Q)} \\ \vdots & \ddots & \vdots \\ W_{Q1}|_{Num(f_t \in q_Q | f_{t-1} \in q_1)} & \cdots & W_{QQ}|_{Num(f_t \in q_Q | f_{t-1} \in q_Q)} \end{pmatrix} \tag{4}$$

$$W' = \begin{pmatrix} W'_{11}|_{P(f_t \in q_1 | f_{t-1} \in q_1)} & \cdots & W'_{1Q}|_{P(f_t \in q_1 | f_{t-1} \in q_Q)} \\ W'_{21}|_{P(f_t \in q_2 | f_{t-1} \in q_1)} & \cdots & W'_{2Q}|_{P(f_t \in q_2 | f_{t-1} \in q_Q)} \\ \vdots & \ddots & \vdots \\ W'_{Q1}|_{P(f_t \in q_Q | f_{t-1} \in q_1)} & \cdots & W'_{QQ}|_{P(f_t \in q_Q | f_{t-1} \in q_Q)} \end{pmatrix} \tag{5}$$

Calculation of the Markov Transition Field. The Markov Matrix does not consider the dependence between the distribution of time series F and time step t_i, while the Markov Transition Field aligns each transition probability along time and retains information in time series. That is, time step i and j in W' refer to q_i and q_j, respectively, so W_{ij} only represents the transition probability between bins. Therefore, we consider the time dependence to extend $W'$to the $N * N$ MTF M (Eq. 7). M_{ij} represents the transition probability between time step f_i and time step f_j, that is, $M_{ij} = W'_{ij}|_{P(f_i \in q_i | f_j \in q_j)}$.

When $j - i = 0$, that is, the main diagonal M_{ii} represents the probability of self-transition.

$$M = \begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1N} \\ M_{21} & M_{22} & \cdots & M_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ M_{N1} & M_{N2} & \cdots & M_{NN} \end{pmatrix} \tag{6}$$

$$= \begin{pmatrix} W'_{ij}|_{P(f_1 \in q_i | f_1 \in q_j)} & \cdots & W'_{ij}|_{P(f_1 \in q_i | f_N \in q_j)} \\ W'_{ij}|_{P(f_2 \in q_i | f_1 \in q_j)} & \cdots & W'_{ij}|_{P(f_2 \in q_i | f_N \in q_j)} \\ \vdots & \ddots & \vdots \\ W'_{ij}|_{P(f_N \in q_i | f_1 \in q_j)} & \cdots & W'_{ij}|_{P(f_N \in q_i | f_N \in q_j)} \end{pmatrix} \tag{7}$$

Construction of the Time Series Network. After getting the Markov Transition Field M, we model it as a graph structure $G = (V, E)$. The index of row/column represents the vertex index, and M_{ij} represents the edge weights. We focus on analyzing the continuous time step in the video, so that we extract only the adjacent time steps to construct a time series network graph (Cheng et al. 2020). Initially, we employed force-directed algorithms for layout. However, such a design was insufficient in providing visual guidance to help users explore special patterns in the network graph. Therefore, we divide the time series network into communities based on community detection ideas to assist users in exploring the relationship between temporal information of the feature and video events (**T2.3.1**).

The Louvain (Blondel et al. 2008) has high computational efficiency and does not need to manually specify the number of communities, but automatically obtains community with the highest modularity, so we employ the Louvain algorithm to get community detection. We extract the node index, edge index and weight of M to construct the network, and the weight is mapped to width of the edge. The community category and modularity of the node are mapped to node color and size, respectively.

The time series network for *Anomaly* feature of a certain video is shown in Fig. 5. Between the 155th and 170th seconds, these frames belong to three communities () . The most special one is the 159th frame, which connects frames of the other two communities. We only keep the connection one time step before and after it to highlight this key mode. In addition, we design a line-block chart to show transition of the *Anomaly* value between bins to assist Time Series Network analysis. Between the 155th and 159th seconds, the *Anomaly* value is in low-bin domain, and the fluctuation is not large. At the 159th second, the *Anomaly* value rises abruptly and transfers to high-bin domain. Between 159 seconds and 170 seconds, the *Anomaly* value is in high-bin domain and fluctuates greatly. Retrieving the original video content found that between the 155th and 159th seconds, the students were in class. At the 159th second, the students quickly fled the classroom after hearing the alarm sound (Fig. 6).

5.5 Frame-feature tree

In this part, we realize the feature-based video summary by analyzing the key mode of the temporal change in feature series (**T2.1.1**). The whole process is divided into three stages: simplifying feature series, acquiring the importance of points, and constructing a feature tree.

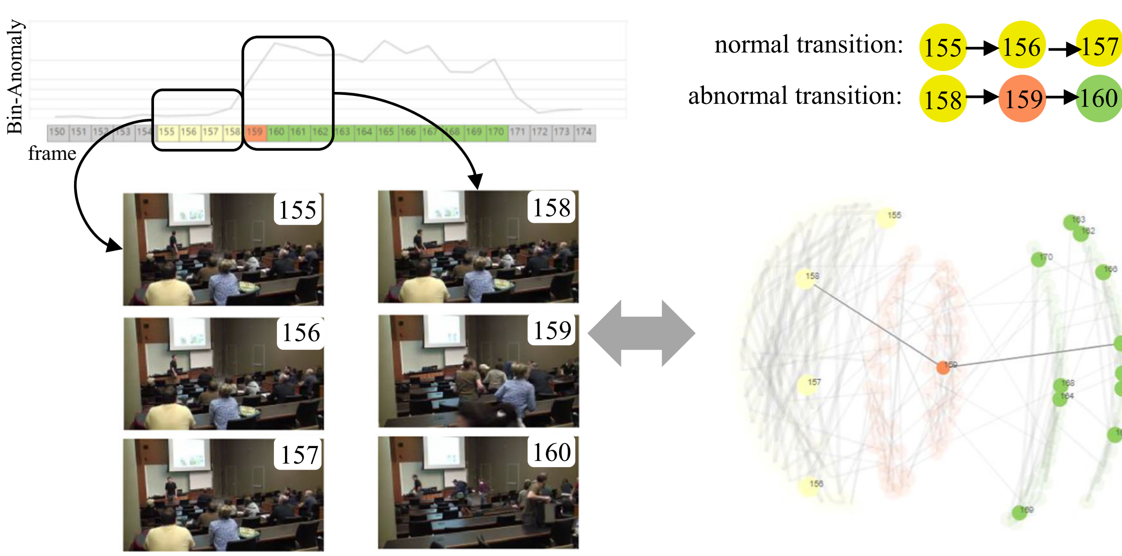


Fig. 5 Normal transition and abnormal transition in Time Series Network of the *Anomaly* feature

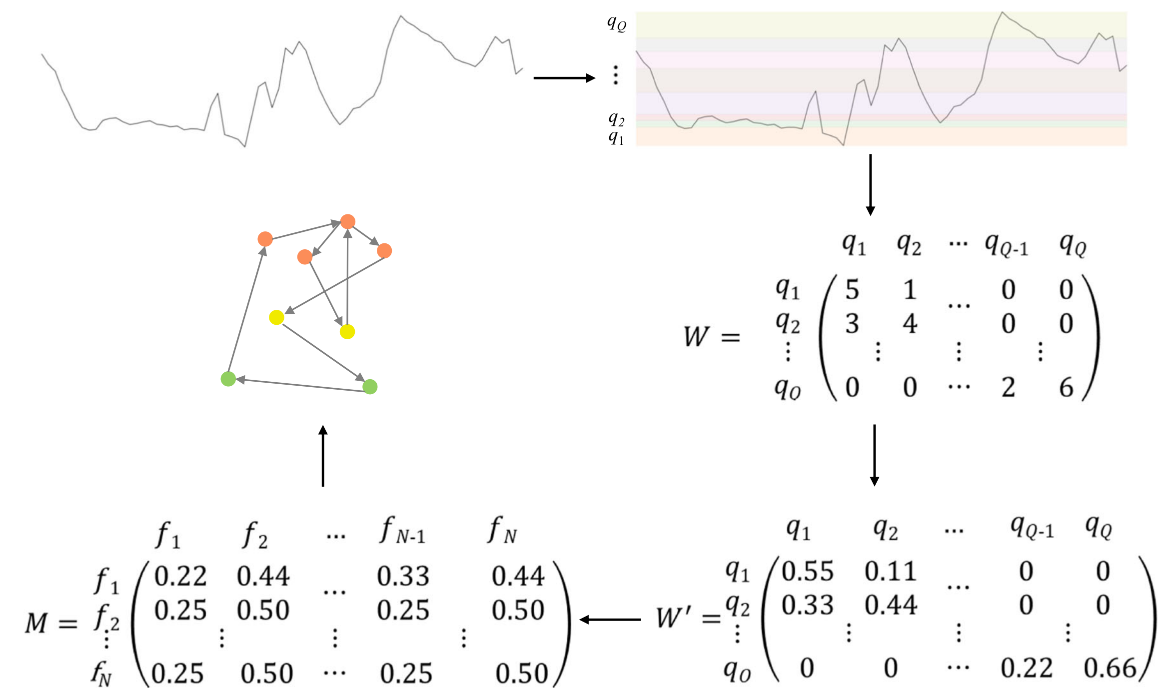


Fig. 6 The flow of using the Markov Transition Field to construct time series network. (1) Firstly, we discretize the series F into Q quantile bins, (2) then calculate the Markov Matrix W' and Markov Transition Field M, (3) and finally construct the time series network based on M

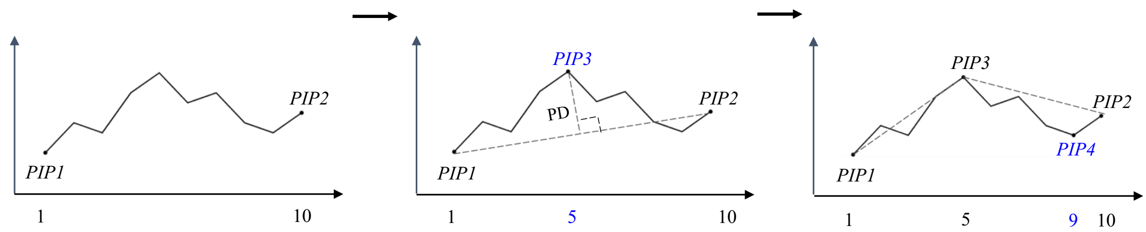
Simplification of Feature Series. In the research process, we found that the feature similarity of adjacent frames is relatively high. In order to reduce redundancy and highlight key patterns, we decided to simplify the series of features. Identifying the perceptually important points (PIP) was first introduced in 2001 and its core ideas can be traced back to 1973. While simplifying, the importance of points can be sorted to facilitate video summary. Therefore, we decided to employ the PIP to simplify the series. PIP algorithm flow: $Step_1$: First point and last point of the series as the initial PIPs; $Step_2$: Among the current PIPs, the farthest point in series becomes next PIP; here, we measure the distance between PIPs by calculating the perpendicular distance (PD). $Step_3$: Repeat $Step_2$ to find next PIP until the number of PIPs meets the threshold. Finally, connect all the PIPs to get a simplified series.

Importance of PIPs. In the process of finding PIPs, we evaluate the importance of each PIP. The first point and the second point are defined as the initial PIPs, and the importance is defined as 1 and 2, respectively. The importance of the third PIP is defined as 3. Repeat the previous step to get the importance list of all points. The workflow in Fig. 7 uses 10 PIPs as an example to describe the series simplification and importance ranking.

Construction of the Frame-Feature Tree. After the above steps, we obtain PIPs-Importance information, and further build a binary search tree. Ignoring PIP_1 and PIP_2, the tree is iteratively constructed from PIP_3, where PIP_i represents the PIP whose importance is i. As shown in Fig. 3(H), we construct a Anomaly-Tree. The number label represents index of the frame, and color is mapped to the importance of the frame. The video summary is placed under the tree in a carousel manner.

6 Case study

In this section, we evaluate the proposed system through a case study based on real-world surveillance videos from multi-camera, which contains a total of four cameras. View HC3 and View HC4 are two surveillance cameras located at different angles in the lobby, and View IP2 and View IP5 are two surveillance cameras located at the front and rear of the classroom. Hereinafter referred to as *HC*3, *HC*4, *IP*2 and *IP*5. The four surveillance cameras monitor the occurrence of the same event from different angles. In this case, through experiments we found that the high-level semantic information features: *Memory* and



Importance	*x*	*y*	PD
1	1	0.1	n/a
2	10	0.5	n/a
3	5	0.9	0.57
4	9	0.3	0.22
5	3	0.3	0.09
6	2	0.4	0.14
7	6	0.6	0.08
8	7	0.7	0.14
9	8	0.4	0.04
10	4	0.7	0.03

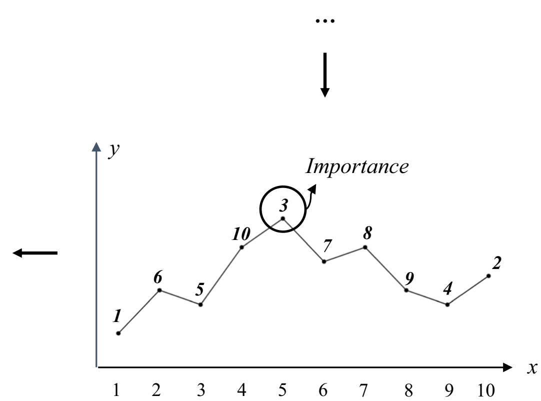


Fig. 7 Workflow of the example for 10 PIPs

Anomaly can better convey information. We determined that the weights α_{aes}, α_q, α_m, α_{ab} are, respectively: 0.1, 0.1, 0.4, 0.4.

Event Consistency. As shown in Video Overview (Fig. 3A), the *Summary* of *HC*3, *HC*4, *IP*2, and *IP*5 are all more obvious in early and late stages of the video. As shown in in Feature Detail (Fig. 3D), *Memory*: *HC*3 and *HC*4 are higher in the later stage, while *IP*2 and *IP*5 are higher at the beginning and end. *Quality*: *HC*3 and *HC*4 fluctuate greatly at the beginning and end, while *IP*2 and *IP*5 are lower at the beginning and end. *Anomaly*: The four cameras are higher at the beginning and the end. Brush pixel bar of the two periods for analysis. As shown in t-SNE View (Fig. 3F), frames of the two periods are clustered together, which are clearly distinguished from other periods. As shown in Fig. 8, the network is constructed according to the *Anomaly* feature, and the analysis is performed in three time periods.

(a) Between the 5th and 25th seconds, the *Anomaly* values of the four videos have a tendency to transition from low-bin domain to high-bin domain, but the overall span is not large. In the Time Series Network, the frames of *HC*3 and *IP*5 are divided into the same community, while the frames of *HC*4 and *IP*2 are divided into two adjacent communities. Retrieving the video content found that during this period, all four cameras captured people entering the classroom from the lobby one after another.

(b) Between the 70th and 90th seconds, the *Anomaly* values of the four cameras are not as high as in period (a). The *Anomaly* values of *HC*3 and *HC*4 are in low-bin domain, with almost no fluctuation. These frames are divided into the same community. The *Anomaly* values of *IP*2 and *IP*5 are transferred in multiple bin domains, with slight fluctuations. Most frames of *IP*2 are divided into adjacent communities, while the frames of *IP*5 are all divided into two adjacent communities. Retrieving the video content found that during this period, all the students had entered the classroom and were active in the classroom, and no one was walking around in the lobby. The reason for the anomaly in the 89th frame of *IP*2 is that the teacher abruptly shut down the course player at that second, and the screen flickered obviously.

(c) Between the 155th and 175th seconds, the *Anomaly* values of the four cameras have a clear upward trend compared with the previous two periods, and most of them are in high-bin domain. The *Anomaly* values of *HC*3, *IP*2 and *IP*5 are all transferred in multiple bin domains, and these frames are divided into different communities. In contrast, the *Anomaly* value of *HC*4 did not change much before the 175th second, and these frames were divided into two adjacent communities. Retrieving the video content found that

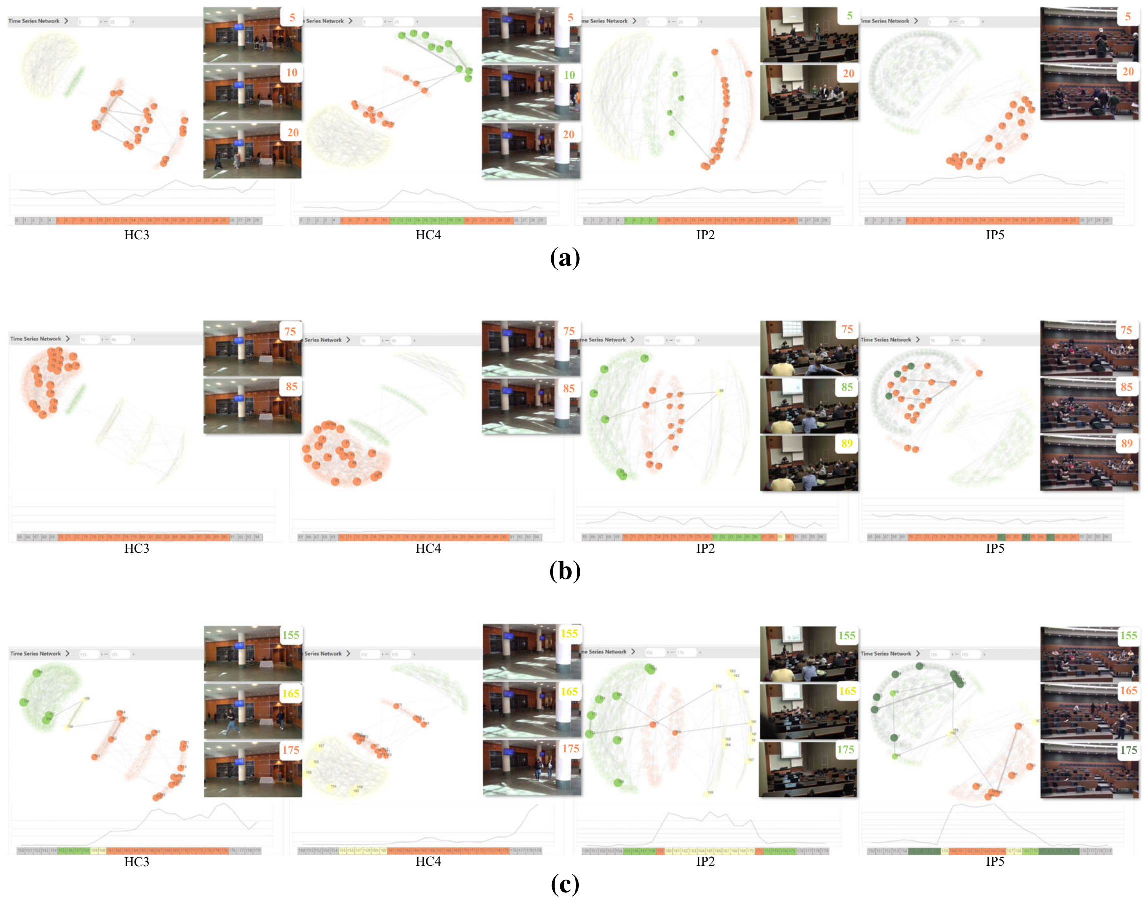


Fig. 8 Case of cross-camera monitoring: event consistency. **a** Between the 5th and 25th seconds, the students entered classroom through lobby and the *Anomaly* values have an upward trend. **b** Between the 70th and 90th seconds, the students were in class and the *Anomaly* values fluctuate little. **c** Between the 155th and 170th seconds, the students escaped from classroom and the *Anomaly* values rise sharply

during this period, the students quickly escaped from the classroom and passed through the lobby after hearing the alarm.

These four cameras record the incident that students enter the classroom, the teacher teaches and everyone escapes quickly after hearing the alarm. Therefore, the *Anomaly* values of the four cameras have an upward trend and obvious fluctuations in the early and late stages, and little change in the other periods. In other words, although they are cameras from different angles, there will be certain similarities in features due to the same event.

Location Difference. As shown in Fig. 8a, c, the *Anomaly* feature of *HC*3 and *HC*4 have opposite trends at the 10th second and the 175th second. At the 10th second, the crowd at the classroom door gradually reduced, and a lady wearing a brightly colored scarf appeared in the surveillance range of *HC*4, but at this time *HC*3 did not caught her. At the 175th second, almost everyone left the *HC*3's surveillance range. Among them, two men ran straight to the position of *HC*4, and it clearly captured their behavior. Prior to this, no similar incident occurred within the monitoring range of *HC*4. This is why between the 155th and 175th seconds, the *Anomaly* value of *HC*4 is different from the other three (Fig. 9).

In addition, as shown in the dotted box in Feature Detail (Fig. 3Ⓓ), at about the 130th second, the *Memory* and *Anomaly* of *IP*5 change significantly compared to *IP*2. As shown in Fig. 10, between the 130th and 140th seconds, the *Memory* value and *Anomaly* value of *IP*5 have a significant upward trend compared to *IP*2, and they are in high-bin domain. Among them, these frames of *IP*5 are divided into 3 communities in the *Anomaly* feature network. As shown in Fig. 11, at the 135th second, the teacher entered the *IP*5 monitoring range, and at the same time a student changed his seat. Both *IP*2 and *IP*5 can monitor this

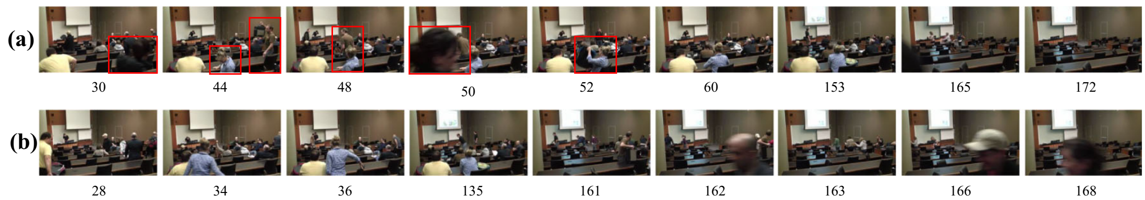


Fig. 9 **a** Video summary of the IP2 Anomaly-Tree (illustrated in Fig. 3①). The tree pays more attention to the incidents of students entering the classroom. **b** Video summary of the IP2 Memory-Tree

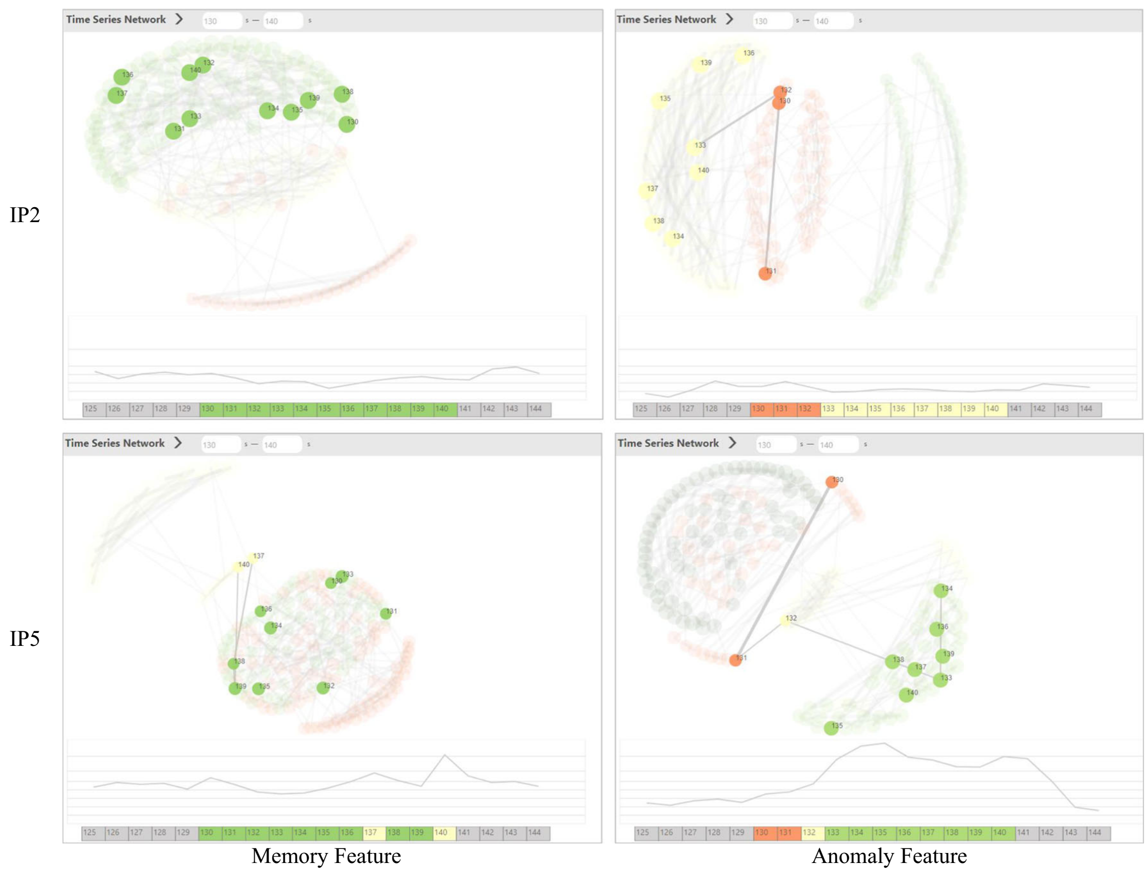


Fig. 10 Case of cross-camera monitoring: location difference. *IP*5 more obvious changes in *Anomaly* feature and *Memory* feature than *IP*2

change, but because the monitored object is closer to *IP*5, *IP*5 has a more obvious upward trend than *IP*2 in *Anomaly* and *Memory*.

At the global level (Fig. 3①), the Anomaly-Tree and Memory-Tree of *IP*2 take the 50th frame and the 34th frame as the root node, respectively, and the root node balance factors are −1 and −3, respectively. The Anomaly-Tree and Memory-Tree of *IP*5 take the 165th frame and the 161th frame as the root node, respectively, and the balance factor of the root node is 3. The two feature trees of *IP*5 pay more attention to the later events, while the Anomaly-Tree of *IP*2 is the opposite. Retrieving the video summary (Fig. 9), we found that *IP*2 captured the activities of students entering the classroom behind the classroom, which was more obvious than the later events.

In other words, although it is the same event, there will be some differences in feature changes due to the different cameras positions.

Fig. 11 Case of cross-camera monitoring: location difference. *IP*5 captures the event more clearly than *IP*2

UT1 Please select a video of interest

UT2 Please describe the key period based on the overview information.

UT3 Please analysis and describe the changing trend of the features.

UT4 Please select a feature of interest and check the threshold.

UT5 Please brush the period of interest and analyze the dimensionality reduction results.

UT6 Please enter the period of interest, analyze the transition of the feature value and the result of the community division.

UT7 Please retrieve the video and describe the cause of the feature change.

UT8 Please summarize your personal understanding of the video content.

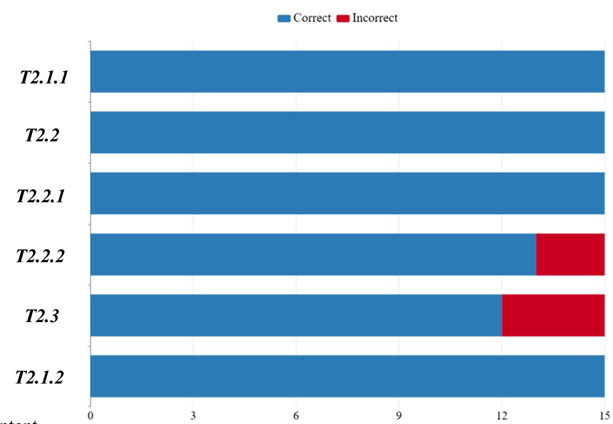


Fig. 12 Task for usage experience: 8 tasks and the evaluation results of UT2–UT7

Q1: Do you think view A is helpful for you to find the key period of the video?

Q2: Do you think view C-Feature Detail is helpful for you to analyze the trend of the feature values?

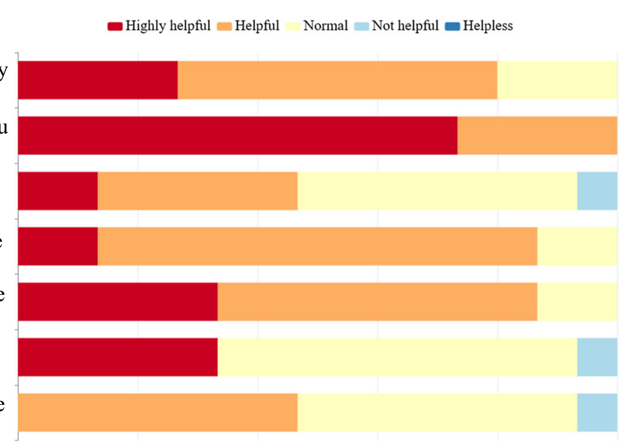
Q3: Do you think view C-Area Chart is helpful for you to analyze the single feature?

Q4: Do you think view E is helpful for you to analyze the frames distribution?

Q5: Do you think view G is helpful for you to analyze the transition of features and the distribution of frames?

Q6: Do you think view B is helpful for you to quickly index the video content?

Q7: Do you think view H is helpful for you to analyze the key content of the video?



SQ1: Is it easy or hard to operate the system?
SQ2: Do you think the system can help you analyze the key content in surveillance video?
SQ3: Do you have any suggestions for our visual design, user interaction or other aspects?

Fig. 13 Questions and evaluation result: 7 questions (Q1–Q7), 3 questions (SQ1–SQ3) about subjective feedback and the evaluation results of Q1–Q7

7 User study

To further evaluate the usability and effectiveness of our system, we conducted a user study. Fifteen participants from different study backgrounds were involved in our system experience task. They are engaged in computer vision, visual analysis, business administration, secondary education, and urban environmental design. Firstly, we spent about $10-15$ min introducing SurVizor and operations of our system. Then, we gave 8 user tasks (UT1–UT8) in Fig. 12 based on the visual tasks proposed in Sect. 3.1. The user tasks are designed to utilize all visual components and evaluate how our system could assist in the visual task. Among them, UT1 and UT8 only involve the subjectivity of participants. For each participant, we will evaluate the results of their task results (UT2–UT7). After they completed the tasks, they needed to answer 7 questions (Q1–Q7) in Fig. 13. Each question has five options, including two positive options, one neutral option, and two negative options. In addition, they needed to complete a simple questionnaire that contains 3 questions (SQ1–SQ3) in Fig. 13 about subjective feedback after completing all the tasks. Finally, We organized the results from all participants and visualized the evaluation results in Fig. 13. The feedback of questionnaire is summarized as follows:

- All participants operated our system smoothly in accordance with user tasks. Among them, two of participants made mistakes in analyzing the frames distribution (UT5), and three of participants made mistakes in analyzing community division (UT6). However, all participants correctly summarized the key content of the selected video. They think that SurVizor is easy to operate and can help them discover the key content of the surveillance video.
- Regarding the visual design of SurVizor, five of participants think that temporal information in Feature Detail is not effective enough, and they hope that SurVizor can add a timeline to assist analysis. Four of participants think that the feature color theme is similar to the community division color theme, which is prone to misunderstandings. Two of participants think that the guidance information of SurVizor is not obvious enough, and they hope to add the icon description and the attributes represented by the axes.
- Regarding functions of SurVizor, one of participants think that a functional module can be added to make up for the lack of model recognition Depth of Field. Two of participants think that the *Aesthetics* feature is not enough to help their analysis, and a function module can be added to filter features with higher effectiveness.

Overall, above results show that SurVizor has shortcomings in visual design and functions. However, the system is simple and easy to operate, and can help users analyze and understand the key content of surveillance videos.

8 Discussion and future work

Model Performance. In our work, three models are employed to quantify features, and visual analysis techniques are further employed for the study. However, the performance of these models will affect the effectiveness and usability of the proposed system, and there are differences in the performance of the models on different datasets. Therefore, in future work, we will unite experts in this field to conduct relevant assessments on different datasets to improve the adaptability of our system.

Scalability. In our work, we employ the fixed sampling frequency of extracting one frame per second for surveillance video data. However, such a sampling frequency may lose key frame information for special scenes. Therefore, in future work, we will try to dynamically adjust the sampling frequency or adaptive selection to extract video frames. In addition, some of our visual designs may not be friendly to lengthy video data. In future work, we will further optimize the visual design and enhance the scalability of our system.

Generalizability. In our work, we employ multi-feature integration methods and time series analysis methods to conduct research and analysis on surveillance video datasets. We think that such a method can be extended to the audio field, air quality field, and network security field, etc.

9 Conclusion

In this paper, we propose SurVizor, an interactive visual analysis system that analyzes the key content of the surveillance videos by integrating multiple features of images and time series analysis methods. SurVizor integrates multiple views, allowing users to explore and analyze from the video level, feature level and frame level. It can efficiently help users discover the key events in the surveillance video and extract the video theme. A cross-camera case and a user study show that SurVizor can help users to explore and analyze the key content of surveillance videos.

In future work, we will further optimize the visual design and enhance the scalability of the system. Moreover, we plan to strengthen the research on image feature selection to further improve the accuracy of the system, and strengthen the learning of video surveillance and computer vision domain to further expand our system.

Acknowledgements This work is partly supported by National Natural Science Foundation of China (62036009), National Natural Science Foundation of China (61972356), Fundamental Research Funds for the Provincial Universities of Zhejiang (RF-A2020001).

References

Alabdulatif A, Khalil I, Forkan ARM, Atiquzzaman M (2018) Real-time secure health surveillance for smarter health communities. IEEE Commun Mag 57(1):122–129

Alameda-Pineda X, Staiano J, Subramanian R, Batrinca L, Ricci E, Lepri B, Lanz O, Sebe N (2015) Salsa: a novel dataset for multimodal group behavior analysis. IEEE Trans Pattern Anal Mach Intell 38(8):1707–1720

Alshammari A, Rawat DB (2019) Intelligent multi-camera video surveillance system for smart city applications. In: Proceedings of the IEEE annual computing and communication workshop and conference, pp 0317–0323

Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):P10008

Bylinskii Z, Isola P, Bainbridge C, Torralba A, Oliva A (2015) Intrinsic and extrinsic effects on image memorability. Vision Res 116:165–178

Chan GYY, Nonato LG, Chu A, Raghavan P, Aluru V, Silva CT (2019) Motion browser: visualizing and understanding complex upper limb movement under obstetrical brachial plexus injuries. IEEE Trans Visual Comput Graph 26(1):981–990

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):1–58

Cheng Z, Yang Y, Wang W, Hu W, Zhuang Y, Song G (2020) Time2graph: revisiting time series modeling with dynamic shapelets. Proc AAAI Conf Artif Intell 34:3617–3624

Chung FL, Fu TC, Luk R, Ng V, et al (2001) Flexible time series pattern matching based on perceptually important points, pp 1–7

Cui Z, Chen W, Chen Y (2016) Multi-scale convolutional neural networks for time series classification. arXiv preprint arXiv:1603.06995

Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartograph Int J Geograph Inform Geovisual 10(2):112–122

Fajtl J, Argyriou V, Monekosso D, Remagnino P (2018) Amnet: memorability estimation with attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6363–6372

Gygli M, Grabner H, Riemenschneider H, Van Gool L (2014) Creating summaries from user videos. In: Proceedings of the European conference on computer vision, pp 505–520

Heer J, Kong N, Agrawala M (2009) Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In: Proceedings of the special interest group on computer-human interaction conference on human factors in computing systems, pp 1303–1312

Hu T, Li Z, Su W, Mu X, Tang J (2017) Unsupervised video summaries using multiple features and image quality. In: Proceedings of the IEEE international conference on multimedia big data, pp 117–120

Lee C, Kim Y, Jin SM, Kim D, Maciejewski R, Ebert D, Ko S (2019) A visual analytics system for exploring, monitoring, and forecasting road traffic congestion. IEEE Trans Visual Comput Graph 26(11):3133–3146

Liao TW (2005) Clustering of time series data: a survey. Pattern Recogn 38(11):1857–1874

Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietikäinen M (2020) Deep learning for generic object detection: a survey. Int J Comput Vision 128(2):261–318

Liu L, Wang Z (2016) Encoding temporal markov dynamics in graph for visualizing and mining time series. arXiv preprint arXiv:1610.07273

Liu M, Shi J, Li Z, Li C, Zhu J, Liu S (2017) Towards better analysis of deep convolutional neural networks. IEEE Trans Visual Comput Graph 23(1):91–100

Liu W, Luo W, Lian D, Gao S (2018) Future frame prediction for anomaly detection: a new baseline. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6536–6545

Sun G, Liang R, Qu H, Wu Y (2017) Embedding spatio-temporal information into maps by route-zooming. IEEE Trans Visual Comput Graph 23(5):1506–1519. https://doi.org/10.1109/TVCG.2016.2535234

Sun G, Wu H, Zhu L, Xu C, Liang H, Xu B, Liang R (2021) VSumVis: interactive visual understanding and diagnosis of video summarization model. ACM Trans Intell Syst Technol 12(4):1–28. https://doi.org/10.1145/3458928

Sun GD, Wu YC, Liang RH, Liu SX (2013) A survey of visual analytics techniques and applications: state-of-the-art research and future challenges. J Comput Sci Technol 28(5):852–867

Talebi H, Milanfar P (2018) Nima: neural image assessment. IEEE Trans Image Process 27(8):3998–4011

Wang J, Wu J, Cao A, Zhou Z, Zhang H, Wu Y (2021) Tac-miner: visual tactic mining for multiple table tennis matches. IEEE Trans Vis Comput Graph 27(6):2770–2782. https://doi.org/10.1109/TVCG.2021.3074576

Wei H, Ni B, Yan Y, Yu H, Yang X, Yao C (2018) Video summarization via semantic attended networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 32

Weng D, Zheng C, Deng Z, Ma M, Bao J, Zheng Y, Xu M, Wu Y (2021) Towards better bus networks: a visual analytics approach. IEEE Trans Vis Comput Graph 27(2):817–827. https://doi.org/10.1109/TVCG.2020.3030458

Wu A, Qu H (2018) Multimodal analysis of video collections: visual exploration of presentation techniques in ted talks. IEEE Trans Visual Comput Graph 26(7):2429–2442

Xu Y, Liu X, Liu Y, Zhu SC (2016) Multi-view people tracking via hierarchical trajectory composition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4256–4265

Ye S, Chen Z, Chu X, Wang Y, Fu S, Shen L, Zhou K, Wu Y (2021) Shuttlespace: exploring and analyzing movement trajectory in immersive visualization. IEEE Trans Vis Comput Graph 27(2):860–869. https://doi.org/10.1109/TVCG.2020.3030392

Yuan J, Chen C, Yang W, Liu M, Xia J, Liu S (2021) A survey of visual analytics techniques for machine learning. Comput Vis Media 7(1):3–36. https://doi.org/10.1007/s41095-020-0191-7

Zeng H, Shu X, Wang Y, Wang Y, Zhang L, Pong TC, Qu H (2020) Emotioncues: emotion-oriented visual summarization of classroom videos. IEEE Trans Visual Comput Graph

Zeng H, Wang X, Wu A, Wang Y, Li Q, Endert A, Qu H (2019) Emoco: visual analysis of emotion coherence in presentation videos. IEEE Trans Visual Comput Graph 26(1):927–937