Design of Hotspot Labeling System in Interactive Video

Dajie Cong, Ping Shi, Kang Wu
Communication University of China
China
a1175871136@163.com, shiping@cuc.edu.cn, wukangustc@163.com

ABSTRACT: *In this paper, we design a comprehensive* interactive video edit and playback system based on hotspot *labeling. To meet the requirement of smoothness in editing and playback, the system framework is designed in DirectShow. And the improved Camshift tracking algorithm and XML technology are combined with DirectShow to achieve the goal of hotspot labeling in varieties of video.*

Keywords: Interactive Video, Hotspot, Feature Description File, Visual Tracking, XML

Received: 13 June 2014, Revised 20 July 2014, Accepted 23 July 2014

1. Introduction

As an important source of information, video enriches our lives and work. Interactive video [1] is a new type of video which gives us more audio-visual experience. It becomes popular in 2005 with the increase of broadband access speed and the advancement of multimedia playback technology (mainly flash). Interactive video is initially designed for advertisements because it can increase the CTR (Click-Through-Rate). Then it is widely used in education, commerce, etc and has been further promoted as a new form of media.

Interactive video is a research focus owing to its broad market prospects. The solution of the technical problems related to interactive video can also promote the progress of computer vision technology such as video scene segmentation, target tracking and object segmentation. In this paper, we design an interactive video editing and playback system based on video hotspot. Hotspot [2] (hot spot / hot zone), also known as anchor, refers to the region which can respond to mouse operation independently on the platform of computer. In web pages, hyperlink is one form of hotspot. And picture hotspot refers to image hyperlink. The mouse click to different regions of hotspot image can link to different hyperlink. Mouse events control the links between additional data and the objects in video.

The system we designed is mainly applied to the field of video hotspot editing. Network transmission environment, editing efficiency and user experience have been considered for the design and implement of our system. The system can achieve the functions such as video hotspot editing, hotspot labeling interactive video playback, saving of label information, etc.

2. Background And Related Work

Common types of interactive video are clickable interactive video, customizable interactive video, exploratory interactive

video and augmented reality interactive video [3]. Among them, clickable video can present related information or guide viewers to choose interactive new contents when mouse cursor is at a specific location (specific characters, items, etc.) of the picture. It is a rapidly developing type of interactive video. It can be used to push advertisements and analyze users' behaviors. The system designed in this paper provides a hotspot-labeling-based implementation method for clickable video. In this system, video objects can be selected as the hotspots and the additional information can be added to every hotspot to achieve dynamic information display.

Extracting the spatio-temporal information of hotspot objects from video sequences is the first step of hotspot labeling edit. The method combines visual tracking with manually adding is currently an effective solution due to its efficiency in editing and accuracy in extracting. In addition, the format design of hotspot spatio-temporal information and additional information determine the playback performance. The feature description file which contains video editing information and metadata of additional information, can be designed in the formats of HTML, SMIL and XML [4]. HTML is designed to display data, with its focus on the appearance of the data. SMIL has the advantages of time controllability, presentation layouts, multi-language support and multi-bandwidth support. If the labels in SMIL are already specified, it is not good for the expansion of system. XML is designed to transmit information. Since the tags in XML are not predefined, it is self-descriptive. We choose XML as the format of feature description files in our system because of its flexibility. Online streaming media expansion is taken into consideration when designing the hierarchy and nodes of XML. Low latency and process synchronization in multitasking environment are priority factors affecting the function of the system. And multi-tasking in interactive step includes playback control, data reading, event response and information display.

3. System Design

Based on the previous research achievements, requirement of our system is as follows:

- Read, decode and render video files.
- Dynamic display (in response to mouse operation) of the Additional information such as text.
- Video playback controls (play, pause, fast forward, rewind, timeline control).
- Read and store video editing files (is called feature description files in this paper).
- Preview of editing and achievement of hotspot labeling video player.

The system is designed from the aspects of software structure and use case to achieve the requirements above.

The system is divided into two parts: Edit-End and Play-End. Play-End is for users to watch hotspot labeling video. And its framework is similar to Edit-End besides more concise interface. Also editing is deleted from Play-End. Thus the following system architecture is mainly for Edit-End. As shown in Figure 1, the system architecture is divided into three layers: UI Layer, Business Logic Layer and Data Access Layer [5]. Data Access Layer determines the object structure in the memory for reading or storing of the video and feature description files. The object structure in the memory provides data support for Business Logic Layer. Business Logic Layer is the core of the system, and its main functions are hotspots rectangular marquee and additional information editing. Meanwhile, preview of editing effects is also necessary. UI Layer is for the layout and display of user interface. By detecting user events, Business Logic Layer respond to and process user requests and send the results to UI Layer for display.

Figure 2 is Use-Case diagram. The participants of the system are editors and users, and user's case is a part of editor's. Preview and playback control are available for both. And editor can also do hotspot labeling related operations. The way of manual rectangular marquee and visual tracking can be chosen by editor depending on specific circumstances.

4. Implementation of Key Technologies

4.1 Hotspots Rectangular Marquee

There are two ways of selecting hotspots in our system: visual tracking using Camshift algorithm and manual rectangular marquee.

4.1.1 Visual tracking using Camshift algorithm

Camshift algorithm has been chosen in our system to achieve visual tracking. Bradski [6] proposed Camshift (Continuously Adaptive Mean Shift) based on the color histogram of the target pattern. The algorithm has high efficiency and performs well

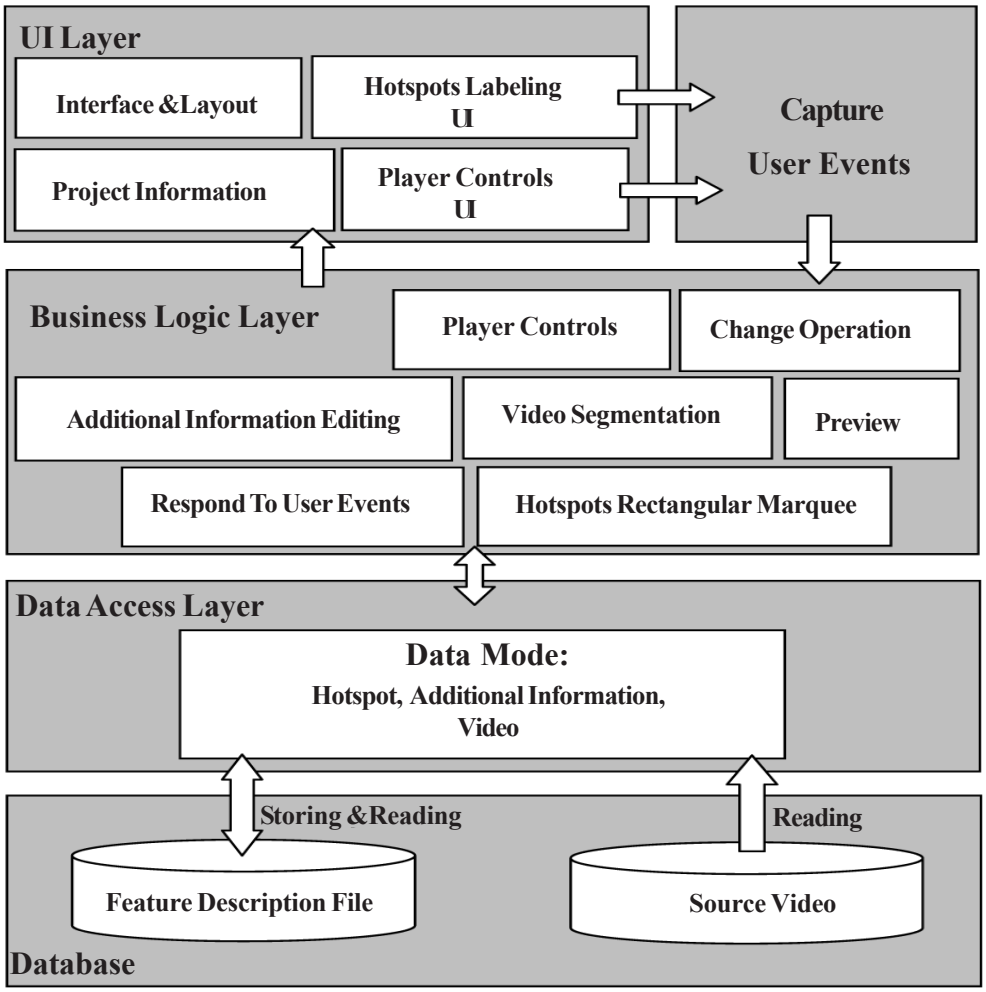


Figure 1. Software framework of Edit-End

in the case of target deformation and occlusion. However, Camshift cannot be applied to the conditions which have various brightness and complex backgrounds. We proposed an improved Camshift algorithm based on RGB histogram equalization by combining the classical Camshift with image equalization. The proposed algorithm performs well in various conditions compared with the classical one. Camshift is based on H channel. Since the effect of image equalization in H channel is not obvious, we convert a image from HSV to RGB, equalize it in RGB channels, and then convert it back to HSV space. Histogram equalization in RGB can perform better in extending the hue differentiation between pixels according to the nonlinear between H and RGB. In addition, there are many extreme situations (the number of the pixels whose values are beyond the equalization boundary is greater than a preset value) in color distribution in RGB channels. In this case, images after histogram equalization will be changed to a particular hue. It is bad of this for processing of Camshift. The correction of equalization boundary is indispensable. Finally we use the boundary of the correct channel to calculate the boundary of the extreme distribution channel according to a certain proportion.

Figure 3 is the flow chart of the improved Camshift. Initialize a search window after RGB histogram equalization, match and track by similarity iteration of kernel function, and obtain the tracking window in the next frame, then loop from RGB histogram equalization. The underlying core in search and match is Mean Shift [7].

Hotspots Rectangular Marquee using visual tracking sharply increases the size of feature description files while improving the efficiency of editing. For hotspots with a long duration, the position information of every hotspot is too much. It will increase the difficulty in storing and transmission. Furthermore, it also increases memory usage during preview and playback. We extract key frame under the premise of not affecting the tracking accuracy due to spatio-temporal redundancy in hotspots. Figure 4 is an example of key frame extraction. Use dot, triangle and star to represent the center of hotspot in each frame.

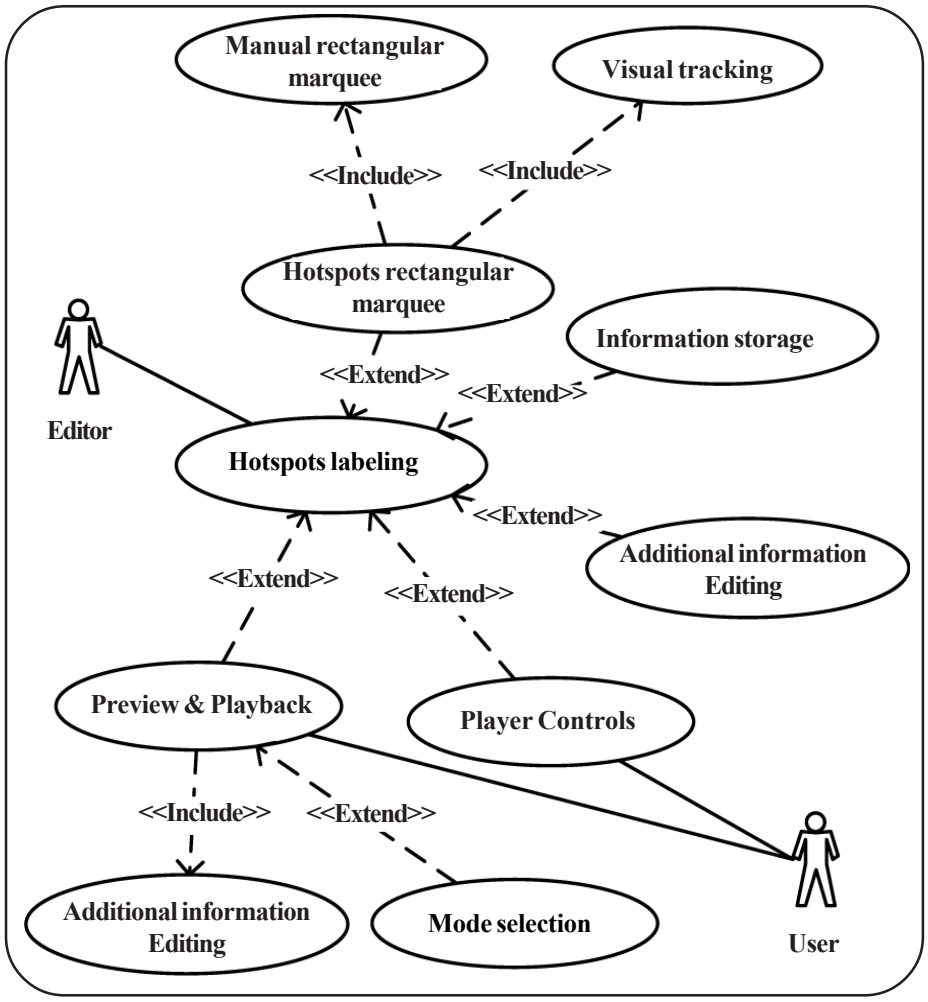


Figure 2. Use-case diagram

Connecting points between lines represent time sequence. The hotspot in original key frames is shown in Figure 4.a. In some adjacent frames, position of the hotspot becomes a cluster such as the areas in the dashed boxes in Figure 4. The key frame whose moving distance and change of size is less than the threshold will be discarded. All frames of one cluster will be replaced with the initial frame approximately. The results are shown in Figure 4. b. Red triangles represent reserved key frames. Then keep key frames in inflection points and nonlinear motion key frames. And discarded frames can be reproduced in liner interpolation. The key frames which store the information of duration shown in the dashed boxes in Figure 4. b cannot be dropped in this step. The stars in Figure 4. c are the results of this step. The number of reserved key frames in each step is shown in Figure 4. Frame extraction rate is more obvious due to the regularity of the movements in practice.

4.1.2 Manual rectangular marquee

Manual rectangular marquee method is shown in Figure 5. Liner interpolation is used to calculate the hotspots in non-key frames. If frame *a* and frame *b* ($a < b$) are chosen as key frames, and location information of hotspot in frame *a* and frame *b* is:

Frame *a*: Centerÿ (X_a, X_b), **Width:** W_a, Height: H_a;

Frame *b*: Centerÿ (X_b, Y_b) , **Width:** W_b, Height: H_b.

Then calculate the position information of hotspot in frame *m* ($a < m < b$) with the following formula:

$$X_m = \frac{(b-m)X_a + (m-a)X_b}{(b-a)}, \tag{1}$$

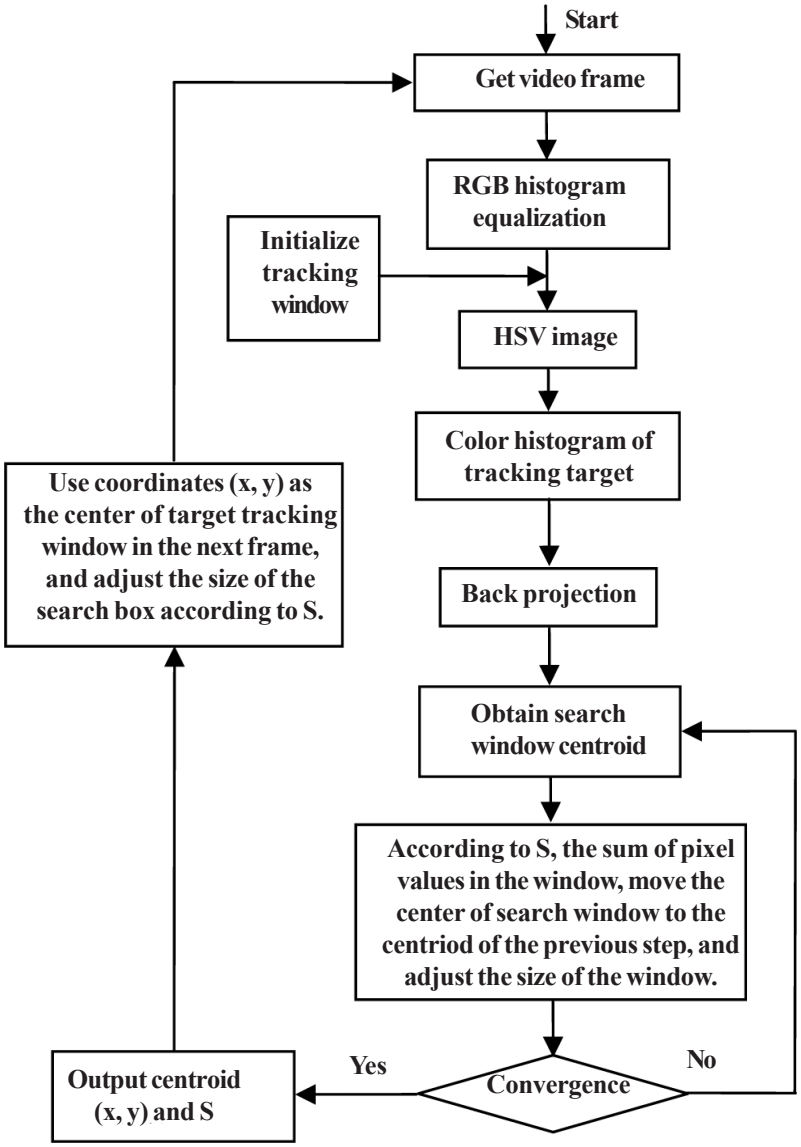
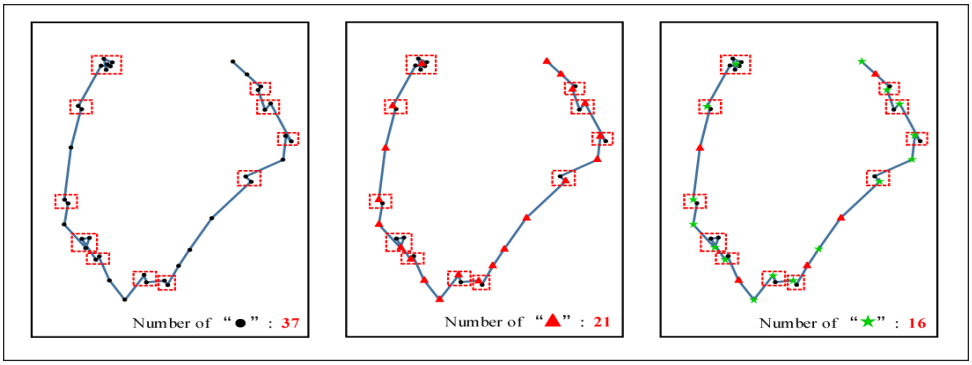


Figure 3. The flow chart of the improved Camshift algorithm



a. Original key frame b. Step 1 of the extraction c. Step 2 of the extraction

Figure 4. Key frame extraction

$$Y_m = \frac{(b-m)\,Y_a + (m-a)\,Y_b}{(b-a)}, \qquad (2)$$

$$W_m = \frac{(b-m)\,W_a + (m-a)\,W_b}{(b-a)}, \tag{3}$$

$$H_m = \frac{(b-m)\,H_a + (m-a)\,H_b}{(b-a)}, \tag{4}$$

where (X_m, Y_m) is the center of hotspot in frame m, W_m and H_m are width and height of the hotspot.

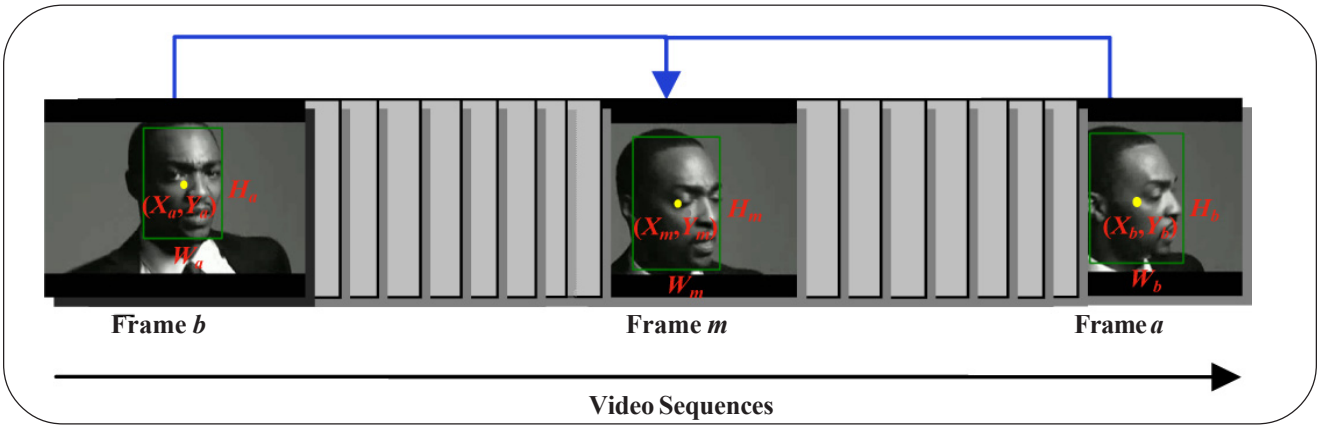


Figure 5. Manual rectangular marquee and linear interpolation

4.2 Feature Description File

Our hotspots labeling system doesn't change video contents. It just saves and transmits the relationship between main video and additional media resources in feature description files. After reaching the client, the client renders the appropriate media contents according to the feature description files.

Points of designing the feature description files are:

• Effectiveness in reading and storing, extension in streaming media format.

• Various additional media format support.

• Non-linear storage of relationship among media elements [9].

We use XML [10] (eXtensible Markup Language) as the format of feature description files in our system mainly because of its high degree of freedom (self-definition of nodes).

Figure 6 shows the design of feature description file. It is divided into the main feature description file and the secondary feature description file. Each video segment corresponds to a secondary feature description file. The hierarchical file design can reduce memory consumption and improve the efficiency of query.

"*Additional information*" in Figure 6 is defined as an action provided to the user after clicking the metadata on the video [11]. The additional information can be texts, images, video, etc. The most promising approach in modeling the Application domain is data-centric [12]. The additional information is organized on two levels distinguishing between:

• Information fragments (or atomic concepts), at the lower level, like texts, images, videos, etc;

• Presentation units (or pages), composed of fragments, that are presented to the user;

Figure 7 and 8 give the examples of the main and secondary feature description files based on XML specification. The main feature description file provides main video file information and segment information. And segment information means the start and end time of the segment. The secondary feature description files store information of hotspot and additional description. Our System currently only supports overlaying of texts. For system extension, <*type*> (text, video, image) is necessary. Spatio-temporal information of hotspots is saved in key frame nodes.

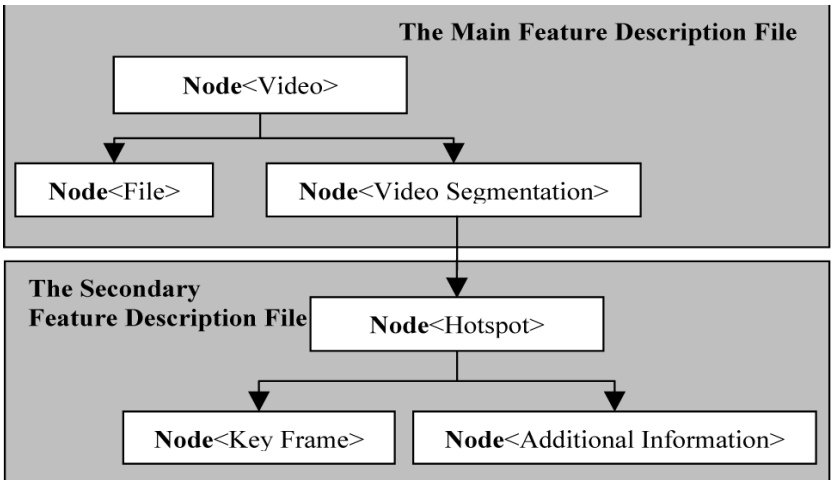


Figure 6. Node design of feature description file

```
<?xml version = " 1.0" encoding = " utf-8"?>
<Root>
<Video>
<VideoFile SavePath = " "/>
<VideoSeg Num = " "  Start = " " End = " " />
……
<VideoSeg Num = " "  Start = " " End = " " />
</Video>
</Root>
```

Figure 7. The main feature description file

```
<?xml version = ”1.0" encoding = ”utf-8"?>
<Root>
<VideoSeg Num = ” ” Start = ”” End = ” ” HotSpotCount = ” ”>
<HotSpot Num = ” ” Start = ” ” End= ” ” KeyFrameCount = ” ”>
<KeyFrame Num = ” ” Time=”” Duration = ” ” Location = ” ”/>
……
<KeyFrame Num = ” ” Time = ” ” Duration = ” ” Location = ” ”/>
<AddInfo Type = ”text” Num = ” ” TextColor = ” ” LTop  = ” ”
overlayText = ” “ size = ” ” font = ” “/>
</HotSpot>
……
<HotSpot>
……
</HotSpot>
</VideoSeg>
</Root>
```

Figure 8. The secondary feature description file

4.3 Dynamic Display of Additional Information

Video processing system based on DirectShow [13] has advantages of efficiency, conveniency and scalablity. Our system should complete tasks of image overlay, file storage and retrieval and mouse event capture while video playing. DirectShow is suitable for our system.

The frameworks of Edit-End and Play-End built with DirectShow are the same. Just as Figure 9 shows, Filter Graph Manager is used in DirectShow for managing the connection of Filters. The video stream read and decoded by Source Filters will be processed by Transform Filters and then be rendered and displayed in Rendering Filters. Meanwhile, Filter Graph Manager controls the capture and response of user events. Callback function in DirectShow obtains mouse event and determines whether to display additional information. Transform Filters display the information dynamically based on the judgment result.

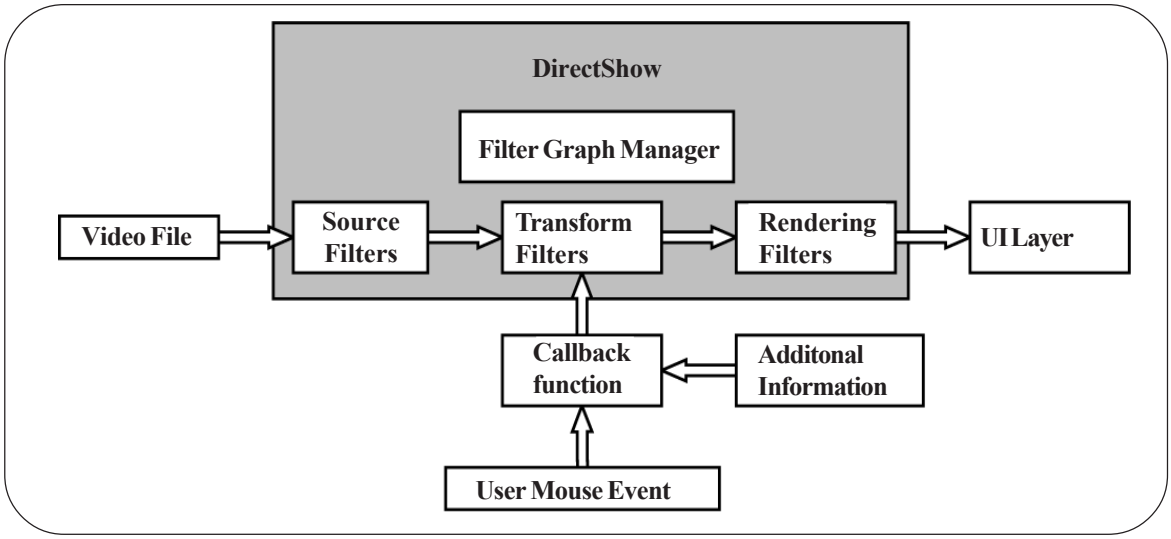


Figure 9. Module of video playback and interactive operation

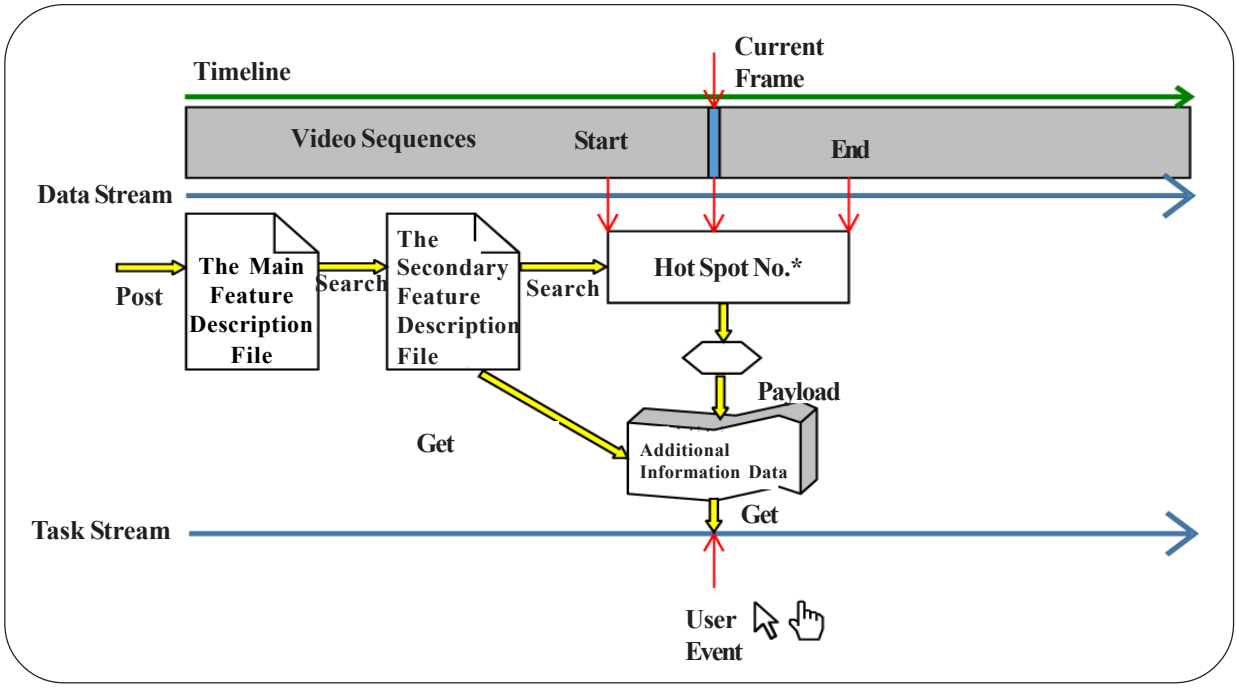


Figure 10. Preview & Playback workflow

To ensure synchronization between playback and response to mouse events, task flow and data flow should be synchronized. A data-driven scientific workflow is proposed in [14]. We specify our workflow depends on it. Figure 10 is the description of preview and playback workflow. Ask for reading the main feature description file while video playback, and search for the currently secondary feature description file according to timer. Search the secondary feature description file for the currently hotspots, and get specific additional information to memory according to hotspot ID. Task flow monitors user mouse events and displays additional information.

5. Functional Analysis of The System

5.1 Implementation of the Architecture

We combined feature description files storage and retrieval, real-time response to user events, the improved Camshift visual tracking with the framework in DirectShow to achieve video hotspots editing.

Figure 11 is the main interface of Edit-End. Edit-End can achieve the function of video editing project management, hotspots rectangular marquee, playback controls, information display, etc.

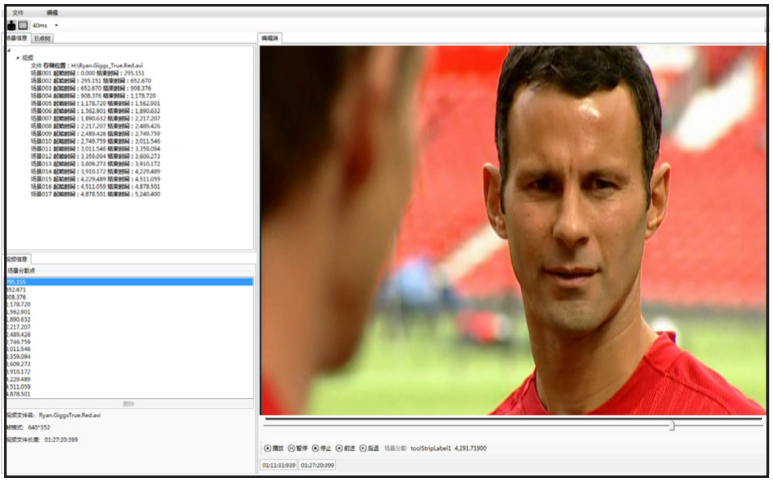


Figure 11. Main interface of Edit-End

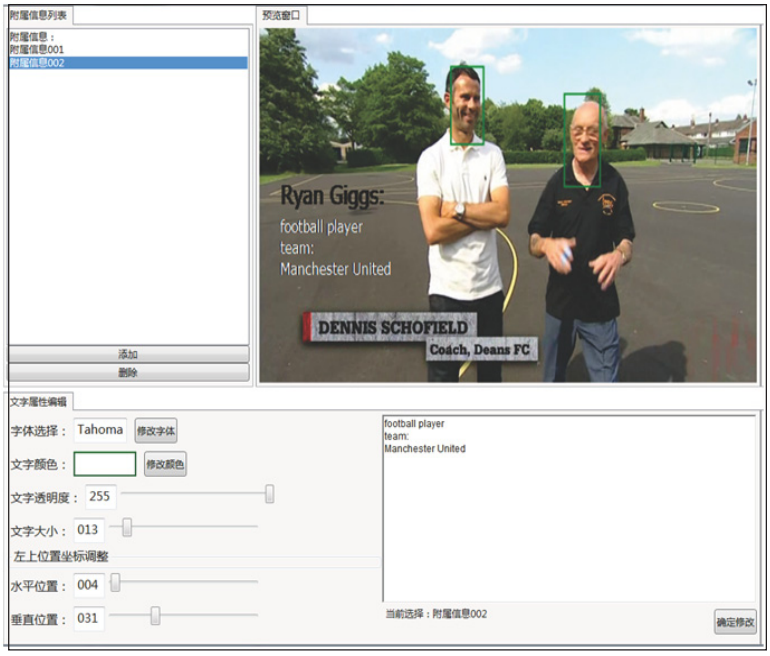


Figure 12. The interface of additional information edit

Additional information is shown as presentation units. Figure 12 shows the edit and combination of additional information block. The key of this step is the flexibility of the layout and repeatability of the operation.

The improved Camshift can be adapted to a variety of applications with enhancement of the standard deviation in hue. Our algorithm is better than the classical Camshift and Stolkin's improved Camshift – ABCshift [15] in terms of the center location error, the overlap rate [16] and the success rates [17]. Specific principle and performance test of our algorithm are in our another paper. Figure 13 shows the tracking results of 2 test sequences using the improved Camshift. The selected pictures for each test sequence are obtained by equal interval extraction.

5.2 Streaming Media Format Extension

Particularly for streaming media, we should package main video, feature description files and additional media into data packets, and send them to the clients by the streaming media server according to some real-time transport protocol [18]. Popular streaming media formats are MOV (Apple QuickTime), ASF (Microsoft, Advanced Streaming Format) and SWF (Adobe, Shock Wave Flash). Multimedia data is divided into several data packets in streaming media formats (such as atom in ASF and packet in MOV). Basic data packets contain at least two types: records file information and the real media data. A typical example is ASF. ASF consists of three standard objects: Header Object, Data Object and Index Object [19]. Header Object and Data Object are the two basic data packets.

Feature description files generated in our system also need to be packaged. The main feature description file is packaged and stored before the real video stream to support the parsing of the secondary feature description files. The secondary feature description files and additional media are packaged into real media data packets. And data type and timestamp are stored in the packet header. Packets representing a certain hotspot need to be placed in front of the current main video packets.

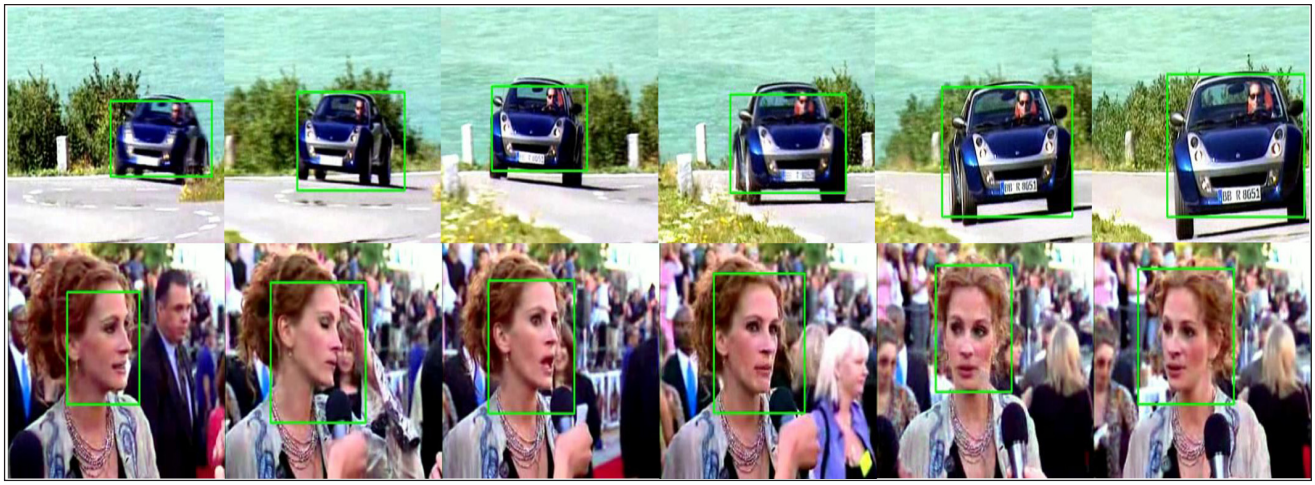


Figure 13. Tracking results of hotspot

6. Conclusion

In this paper, we design a comprehensive interactive video editing and playback system based on hotspot labeling after studying commercial interactive video sites. The system architecture, key techniques and functional analysis of our system are detailed in this paper.

References

[1] Peng, Jian. (2012). Research and implementation of interactive video technology. University of Electronic Science and technology, Computer software and theory.

[2] Guo, Xingji. (2013). Investigation on hot zone of irregular figure, *Computer Applications and Software*, 30 (5) 295-297.

[3] Li, Yulin. (2013). Exploring of interactive video used in the future development. *Radio & TV Broadcast Engineering*, 40 (5) 30-32.

[4] Yatabe, T., Kawasaki, H., Sakauchi, M. (1999). Interactive video description on the network-interactive video representation

of real world based on digital city map, Multimedia Computing and Systems. IEEE International Conference on, 2, p.194, 198 2, July.

[5] Wang, Ying. (2013). Application of three layer software architecture in Web System. Silicon Valley, (11) 76-77.

[6] G. R. Bradski . (1998). Computer Vision Face Tracking as a Component of a Perceptual User Interface. Proc. IEEE Workshop Applications of Computer Vision, p. 214-219, Oct.

[7] Cheng Y. Z. (1995). Mean shift, mode seeking, and clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17 (8) 790-799.

[8] Gary Bradski., Adrian Kaehler. (2009). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, 04.

[9] Cheng, Xueqi. (1996). Bo Han., Research and implementation of hypermedia authoring tools. *Computer and digital engineering*, 05:8-11.

[10] Wu, Yi (2012). Depth understanding XML of C#. Beijing: Tsinghua University press.

[11] Abe, H., Shigeno, H., Okada, K. (2006). Media Synchronization Method for Video Hypermedia Application Based on Extended Event Model, Multimedia and Expo, 2006 IEEE International Conference on, p.1289,1292, 9-12.

[12] Cannataro, M., Pugliese, A. (2000). An XML-based architecture for adaptive Web hypermedia systems using a probabilistic user model, Database Engineering and Applications Symposium, International, p. 257, 265, 2000.

[13] Qiming, Lu., (2003). DirectShow Development Guide. Beijing: Tsinghua University press.

[14] Balis, B. (2012). Hypermedia Workflow: A New Approach to Data-Driven Scientific Workflows, High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion. , p. 100,107, 10-16 Nov. 2012.

[15] Stolkin, R., Florescu, I., Baron, M., Harrier, C., Kocherov, B. (2008). Efficient visual servoing with the ABCshift tracking algorithm, Robotics and Automation. ICRA 2008. IEEE International Conference on, p. 3219, 3224, 19-23 May 2008.

[16] Lin, Dajun., Zheng, Huicheng., Ma, Donghong. (2013). Robust visual tracking using local salient coding and PCA subspace modeling. Information Forensics and Security (WIFS), 2013 IEEE International Workshop on, p. 25, 30,18-21 Nov. 2013.

[17] Shengfeng, He., Qingxiong Yang., Lau, R.W. H., Jiang, Wang., Ming- Hsuan Yang, (2013). Visual Tracking via Locality Sensitive Histograms, Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, p.2427, 2434, 23-28 June 2013.

[18] Ma., Jie., Tian., Jinwen., Liu, Jian. (2003). Streaming Media and File Format. *Computer Engineering and Applications*, 39 (23) 49-52.

[19] Tao, Hongjiu., Liu, Jian., Tian, Jinwen (2001). Analyzing ASF of Streaming Media Format of Windows Media. *Computer and Communications,* 19 (6) 52-55.