Adaptive Fibonacci Search Method of Video Key Frame Extraction

Baomin Shao[#1], Hongyun Jia[*2]

[#]School of Computer Science and Technology, Shandong University of Technology, China

[*]School of Foreign Language, Shandong University of Technology, China

Abstract *- With the development of video techniques, one of the issues of building a proficient video processing system is to present a whole video with key frames to eliminate redundant information. This paper presented a novel video key frame extraction method using adaptive Fibonacci search algorithm. A pre-sampling was employed for selecting the suitable parameters for the sequence search process. After color and texture features were computed and combined, each frame was represented by a 92-dimensional feature vector, with the help of similarity measurement of frame combined feature vector, a divide-and-conquer searching method was executed. Experiments showed that this approach considerably reduced the processing time of each video while maintaining a similar precision and recall rates, at the same time, it could extract key frames from videos more effectively.*

Keywords: *Key frame extraction, Video analysis, Fibonacci search, Image feature vector*

I. INTRODUCTION

With the development of video capture, compression, storage and distribution technologies in recent years, the sharing of various video has gained extensive attention and is a widely used form of worldwide communication, meanwhile the video's extent and volume have increased rapidly. To effectively utilize such huge amount of data, proficient video processing system are needed to access these files with a friendly user interface [1, 2]. In order to achieve this target, one issue that needs to be addressed is the elimination of redundant information, which is to present a whole video with succinct summarization, and users can get aware of the content of any video without watching it entirely [3]. The objective of video summarization is to remove the redundant data which will significantly reduce the amount of information that needs to be processed. Video contains huge amount of information at different levels commonly referred as scenes, shots and frames, and it is necessary to discard the frames with repetitive or redundant information. Key-frames are defined as the representative frames of a video stream and the frames that provide the most accurate and compact summary of the video content. The key-frame detection process depends not only on the application but also on the personal "definition" of the user/developer of what a video summary should include. In this paper, the key frames represent the starting and ending points of any transition, a sequence of key frames can define whole changing of the video, with two or three key frames over the span of a transition, the remaining frames can be filled with in-betweens to create the illusion of movement.

There aren't commonly accepted key frame estimation techniques, different mathematical models and algorithms are developed according to different application fields, these algorithms can be roughly classified into three categories:

(1). In segment-based key frame extraction approaches, a video is segmented into higher-level video components, where each segment or component could be a scene, an event, a set of one or more shots, or even the entire video sequence. This key frame extraction method starts from decomposing the video into temporal segments (shots or scenes) and ends with extracting a fixed number of key frames per temporal segment. These representative frame(s) from each segment are selected as the key frames [4, 5].

(2). Another widely used approach is to use the low-level visual information of all video frames (or all shot frames or all scene frames) in order to group them using e.g. k-means and then select as key-frames that are more similar to the group centers of the groups. Typically, clustering-based models are used to extract key frames from features [6,7]. In clustering-based model, frames having similar features are grouped together and one or more frames from each cluster are selected to generate the desired number of key frames.

(3). The last class of approaches employ a sequential search to video stream. Such techniques start by a "root" key-frame (usually randomly selected as one of the first frames of the video) and then compare one by one the next frames, until a frame with significantly different low-level visual content is found. Then, this becomes the "root" key-frame and the process continues from the next video frame. Because of the enormous number of pixels present in videos, as a rule of thumb, first, feature vectors can be computed from video frames and then these features can be processed to accelerate the extraction of key frames.

In this paper, we proposed an adaptive Fibonacci series based search method for the key frames extraction, and an image combined feature based method to extract key frames from unstructured videos. In this proposed approach, an image combined feature was computed from unstructured videos frames and an information divergence based distance measure of the feature vector to measure dissimilarity between frames of the input video was applied. The combined feature preserved important visual information (texture, edge, color, etc.) of the frame image, and it had been shown to be shift and scale invariance and effective in terms of modeling the spatial structure. A procedure for selecting the key frames using Fibonacci sequence was proposed to reduce extraction time. We searched the frame sequence according to the Fibonacci series feature and located singular point faster, because different video has different characteristics. An adaptive process was used to set the parameters of the search algorithm to optimize the extraction results. This paper is organized as follows. Section 2 reviews Theoretical foundations of the method. In Section 3, the proposed key frame extraction algorithm is described, while Section 4 presents experiment results comparing to ground truth data. Finally, concluding remarks are given in Section 5.

II. THEORETICAL FOUNDATIONS

A. *Fibonacci Series Search*

Fibonacci series are the sequence of values that are generated from a fixed pattern. Fibonacci series are defined by the recurrence relation as represented in Eq.1.

$$f_N = f_{N-1} + f_{N-2} \qquad (1)$$

The initial seeds are $f_1 = 1$ and $f_2 = 2$.

TABLE 1 FIBONACCI SERIES

N	1	2	3	4	5	6	7	8	9	10	11
F(N)	1	2	3	5	8	13	21	34	55	89	144

This Fibonacci series based method was used to select the key frames in the entire scene of the video. It was a sequential search algorithm for minimizing a unimodal function over a closed interval based on the Fibonacci series. Unlike other sequential search methods(i.e. dichotomous, or golden section search method), its main advantage is that for the same starting uncertainty interval and the number of iterations, no other sequential search technique can result in a smaller final uncertainty interval, and this search method requires a predetermined number of iterations.

B. *Combined Feature extraction*

Feature extraction is an important step to efficiently represent the video frames in multi-dimensional space. Color and texture features were used to represent the content of video frames in our proposed algorithm.

1) *Color feature extraction*

Color is the most expressive low-level feature. Each video frame is represented by a 72-dimensional feature vector, obtained from a color histogram. One key issue of such a histogram-based approach is the selection of an appropriate color space. In our case, it is important to remind that the color model reflects the human perception of colors. Compared with other color spaces, HSV color space is the closest to the characteristics of human vision[8]. So in this paper, the color histogram was obtained using the HSV color space, which was also found to be more resilient to noise [9, 10]. Since the human eyes are most sensitive to hue component, the HSV color space was divided into 72 color subspaces, the hue H was divided into 8 parts, the saturation S into 3 parts, and the brightness V into 3 parts. When S was small enough ($s < 0.2$), the perceptual color would turn to a black area, therefore, the range could be neglected. Similarly, when V was small enough (V < 0.2), it was neglected as a gray area. Using this technique, we could improve the computational efficiency and was also robust to small changes of the environment.

2) *Texture feature extraction*

As global color histogram alone is incapable of preserving spatial information presenting in the video frames, our method utilized texture feature along with color histogram to achieve higher semantic dependency between different video frames. Texture feature was extracted from the video frames using edge histogram descriptor [11]. A video frame was first sub-divided into 2 by 2 blocks, and then local edge histograms for each of these blocks were computed. Edges were broadly grouped into five categories: vertical, horizontal, 45°diagonal, 135°diagonal and isotropic. Thus, each local histogram had five bins corresponding to the above five categories. Finally, each frame was represented by a 20-dimensional feature vector corresponding to texture feature. After combining color and texture features, each frame was represented by a 92-dimensional feature vector.

III. PROPOSED METHOD

The proposed method consists of two main steps: (1) video frames pre-sampling to determine the parameter of Fibonacci search; (2) combined feature extraction and key frame extraction.

A. *Video frames pre-sampling*

The first step was to select some frames from the video stream with random time gaps. Only color feature was extracted to compute the similarity of two frames, the similarity was obtained by the χ^2 distance using Eq 2.

$$\chi^2 = \begin{cases} \sum_{i=1}^{k} \dfrac{(h_m(i) - h_n(i))^2}{\max(h_m(i), h_n(i))}, (h_m(i)! = 0 \,\|\, h_n(i)! = 0) \\ 0, else \end{cases} \qquad (2)$$

Where h_m and h_n are histograms of two frames and k was the value of histogram bins. The bigger the variance of all the similarity values was, the lower the parameter of Fibonacci search was set. In this step, very low sampling rate leaded to poor quality of video presentation and decreases the time required to obtain the summary at the same time. In contrast, very high sampling rate could miss important information contained in the video. Mutual information between two frames indicated the extent of similarity between those frames.

B. Key frames extraction

Key frames were detected based on the feature vector difference. The larger the difference, more likely the frame was to be the key frame, and this method could avoid the motion of the lens inside the difference better, and improve certain robustness. At the same time, the following aspects of the problem should be taken into account: (1) two adjacent key frames should not be too close to each other; (2) the difference between the key frame and the previous frame should be the largest of all the frame difference values in the current segment; (3) in the following shot, the difference between the two frames near the part of the shot should no larger than the difference between the the key frame and the previous frame. In other words, the key frame should reflect a maximum value in the difference list of video frames. To seek the singular value which stands for the key frame of video in the list, these steps were followed:

a. *Set k = 1, direction=1, Cf=F(k), Nf=F(k+1);*

b. *If Nf>=Len(video) stop; else compare feature vector of Cf with Nf, if less than a threshold T, go to step c; else go to step d;*

c. *set direction=1, k=k+1, Cf= Nf, Nf= Cf +Fk, go to step b;*

d. *startRewind=1, direction=-1, Cf=Nf, Nf=Cf-F(k-1), k=k-1, go to step e;*

e. *if k=1, key frame found, go to step g; else compare feature vector of Cf with Nf, if less than a threshold T, go to step d; else go to step f;*

f. *startRewind=1, direction=1, k=k-1, Cf= Nf, Nf= Cf +F(k), go to step e;*

g. *Set k = 1, direction=1, Cf= Nf, Nf= Cf+F(k), go to step b.*

The proposed adapted Fibonacci search method of video key frame extraction is illustrated in Fig. 1.

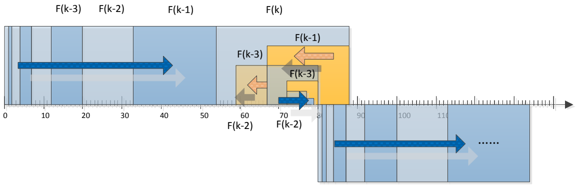

Figure 1 Proposed Fibonacci search method of video key frame extraction

IV. EXPERIMENT AND DISCUSSION

This paper evaluated the key frame extraction approach on a subset of Rai Scuola video archive dataset, which was a collection of ten randomly selected broadcasting videos mainly including documentaries and talk shows (see Fig 2). Key frames of the ten videos had been manually annotated to define the ground truth.

Abbiamo sempre avuto il problema del benzene
Che muta l'equilibrio ambientale di una zona

Figure 2 Some frames extracted from video dataset

In this experiment, we focused on the extraction performance in terms of F1-score for all videos. Table 2 summarized the achieved results and comparison with the frame by frame extraction method. The frame by frame method had a better performance on most videos, while the proposed method behaved better on the fourth and the eighth videos. In these two videos, there were more vague shots, such as fade in and fade out effect, object zoom in and zoom out movement. When dealing with these videos, it was hard to tell where was the key frame based on similarity measure of adjacent two frames, the proposed method could jump a certain step under the guide of Fibonacci series, and skipped the accumulation of tiny changes to get a better F1-score.

TABLE 2 F1-SCORE COMPARISON OF PROPOSED METHOD AND FRAME BY FRAME METHOD

Video number	Total duration	F1-score of our method	F1-score of frame by frame method
1	09:51	0.90	0.93
2	09:50	0.92	**0.97**
3	09:41	0.93	**0.94**
4	09:31	**0.92**	0.89
5	09:30	0.88	**0.96**
6	10:00	0.90	**0.94**
7	10:00	0.82	**0.90**
8	10:00	**0.86**	0.77
9	10:00	0.80	**0.85**
10	10:00	0.91	**0.93**
Average	09:50	0.88	**0.91**

Regarding the time performance, we performed an experiment on a PC with Intel i7-6600U processor @2.60G Hz. By using a divide-and-conquer strategy for searching a sequence by narrowing possible locations to progressively smaller intervals, the Fibonacci search algorithm had time complexity of O (log (n)) and, due to its access pattern for the array elements was much faster compared to the traditional binary search when the arrays being searched were large. In our case, the searching procedure was interrupted when a key frame was found, the consuming time couldn't reach the maximum performance, it could deal with 400 frames per second while 100 frames with frame by frame method, 4 time faster than the latter method. The time consuming of ten videos of both algorithms is shown in Fig 3.

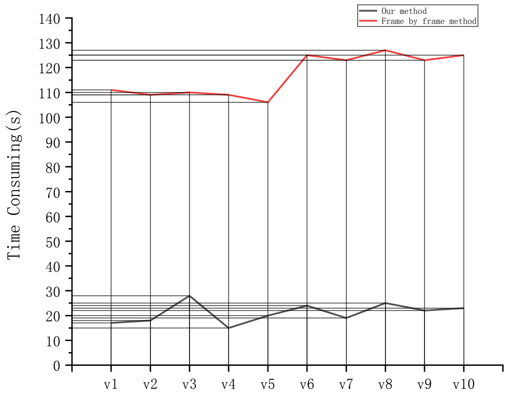

Figure 3 Time consuming comparison of two methods on ten videos

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel video key frame extraction method using adaptive Fibonacci search algorithm and similarity measure of frame combined feature vector. A pre-sampling was employed for selecting the suitable parameters for the sequence search process. This approach considerably reduced the processing time of each video compared to frame by frame search method, while maintaining a similar precision and recall rates. We undertook a comprehensive evaluation of the proposed method on video dataset of RAI using three subjective and three objective measures. The detailed experimental results clearly demonstrated qualitatively and quantitatively that the proposed method produces video key frames with similar quality and high user satisfaction as compared to commonly used technique.

In future, we will focus on using a more extensive set of features like color, motion, shape and texture along with an efficient feature fusion strategy to obtain more meaningful video key frames. Another direction of future research will be the implementation of video key frame extraction system of high user interaction. The execution of algorithm needs to be adjusted to different video type and user requirements to achieve better results.

REFERENCES

[1] D.B. Ponceleon, S. Srinivasan, A. Amir, D. Petkovic, D. Diklic, Key to effective video retrieval: effective cataloging and browsing, in: Proceedings of the ACM International Conference on Multimedia, 1998, pp. 99–107.

[2] B.T. Truong, S. Venkatesh, Video abstraction: a systematic review and classification, ACM Transactions on Multimedia Computing, Communications, and Applications 3 (1) (2007) 1–37.

[3] A.G. Money, H.W. Agius, Video summarization: a conceptual framework and survey of the state of the art, Journal of Visual Communication and Image Representation 19 (2) (2008) 121–143.

[4] S. Uchihashi and J. Foote, Summarizing video using a shot importance measure and a frame-packing algorithm, in IEEE ICASSP, 1999, vol. 6, pp. 3041-3044.

[5] Z. Rasheed and M. Shah, Detection and representation of scenes in videos, IEEE Trans. Multimedia, vol. 7, no. 6, pp. 1097-1105,Dec. 2005

[6] Y. Zhuang, Y. Rui, T. S. Huang and S. Mehrotra, Adaptive key frame extraction using unsupervised clustering, In Proceedings of IEEE International Conference on Image Processing (ICIP), 1998.

[7] Vasileios Chasanis, Aristidis Likas and Nikolaos Galatsanos,Video rushes summarization using spectral clustering and sequence alignment, Proceedings of the 2nd ACM TRECVid Video Summarization Workshop, 2008.

[8] G. Paschos, "Perceptually uniformcolor spaces for color texture analysis: an empirical evaluation," IEEE Transactions on Image Processing, vol. 10, no. 6, pp. 932–937, 2001.

[9] J. Almeida, N.J. Leite, R.S. Torres, VISON: video summarization for online applications, Pattern Recognition Letters 33 (4) (2012) 397–409.

[10] G. Paschos, Perceptually uniform color spaces for color texture analysis: an empirical evaluation, IEEE Transactions on Image Processing 10 (6) (2001) 932–937.

[11] B.S. Manjunath, J.R. Ohm, V.V. Vasudevan, A. Yamada, MPEG-7 color and texture descriptors, IEEE Transactions on Circuits and Systems for Video Technology 6 (11) (2000).