Visual Video Retrieval System Using MPEG-7 Descriptors

Vojtěch Zavřel
Faculty of Informatics
Masaryk University
Brno, Czech Republic
xzavrel@fi.muni.cz

Michal Batko
Faculty of Informatics
Masaryk University
Brno, Czech Republic
batko@fi.muni.cz

Pavel Zezula
Faculty of Informatics
Masaryk University
Brno, Czech Republic
zezula@fi.muni.cz

Keywords

video retrieval system, content-based searching, metric space

1. INTRODUCTION

Video is one of the most popular media these days. However, the volume of the video content grows very fast, e.g. about 24 hours of video content is uploaded every minute to the most famous web site YouTube, and the necessity to search in this content is apparent. Generally, most of the video search systems are based on annotations or use additional text information, which is not always of high quality or lack precision. We present a system that allows user to search videos according to their visual content.

2. VIDEO SEARCHING BY CONTENT

There are several searching techniques commonly used by most of the video retrieval systems. Some of them are systems fully dependent on the text located near to the video like Google. Others (like YouTube) are catalog retrieval systems that usually use some additional information like description, categorization, relevance, relationships, comments and so. Such information can be used when looking for similar videos.

Different systems sometimes called *content based* (e.g. ANVIL[1], VARS[2] or VCode & VData[3]) use annotations to describe individual scenes by visual, audio, time and meta parameters. All those types of systems are highly dependent on the quality of the additional data, which is normally prepared by people [6].

Another possibility is to use automatic annotation and retrieval systems that use different techniques to build the describing information. Some of them are based on MPEG-7 descriptors or machine learning. Those are often restricted to a specific type of use – they are sometimes called *domain specific systems* [2, 4]. Others (e.g. SAPIR[4]) use metric space based similarity search principles where searching is based on an example query [3]. Basic idea is to search not on the level of the actual multimedia object but rather using characteristic features extracted from these videos. Features can be implemented as multidimensional vector-space descriptors and the similarity of the video content is then defined using these descriptors.

Although video contains several information carriers like image, audio, text, or time, for many people the image is still the most important part of the video. Therefore we demonstrate a system that allows users to search according an example video or a video scene. Visual part of the video in our system is represented by *keyframes*, i.e. still images from the video that best characterize a fraction of the video – typically one camera shot.

2.1 Keyframes extraction

We have prepared our own tool for keyframe extraction following the ideas of MPEG-7 standard's video summarization technique. This tool is based on MPEG-7 reference implementation library (eXperimental model) and its Summarization Client application [5].

The tool browses all the frames in the source video file and computes the frame change level according the specified precision parameters. Whenever a significant change is detected (and the given parameters are satisfied), a keyframe is extracted. Thus, each video is split into a sequence of keyframe images.

For just one minute of the video (even static) it is necessary to process 1500 frames (PAL format has 25 fps). The keyframe extraction allows us to reduce this number to 1-10% depending on the dynamicity of the source video. On our collection, we have observed about 4% reduction on average. It is still quite high amount of images, but using a scalable image retrieval system such as MUFIN [1], we have a potential to index millions of short videos. Moreover, the extraction parameters can be adjusted to further lower the ratio at the cost of losing some precision.

2.2 Keyframes indexing

To define the similarity between the videos (or the respective video scenes), we use the extracted key frames. Five MPEG-7 global visual descriptors – color structure (CS), color layout (CL), scalable color (SC), edge histogram (EH), and homogeneous texture (HT) [5] – are taken from each key frame. The dissimilarity of two key frames is defined as a

[1]http://www.anvil-software.de/
[2]http://sourceforge.net/projects/vars/
[3]http://social.cs.uiuc.edu/projects/vcode.html
[4]sapir.insti.cnr.it/index

SISAP'10 September 18–19, 2010, Istanbul, Turkey

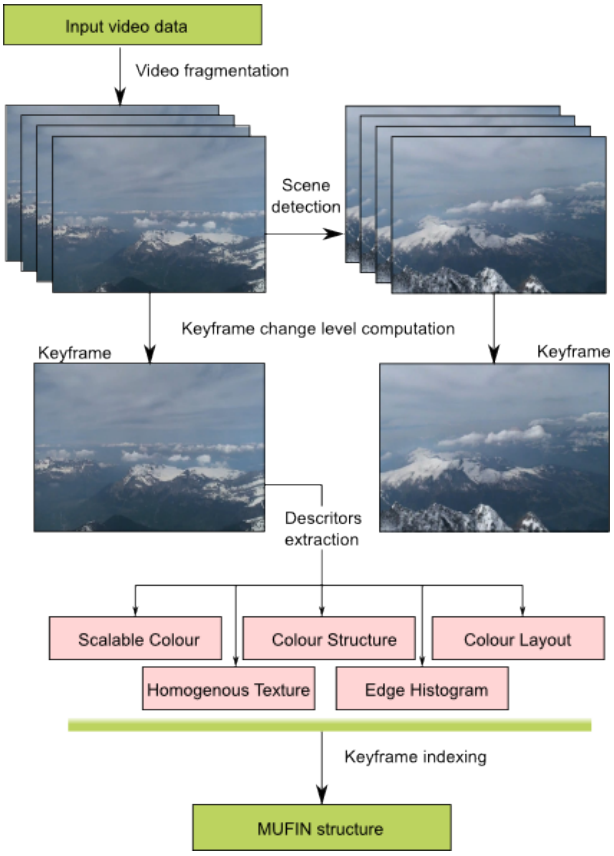

Figure 1: System architecture

weighted sum of the distance of each of the five descriptors in both the key frames:

$$d(o_1, o_2) = 3 * d_{CS}(CS(o_1),CS(o_2))+ 2 * d_{CL}(CL(o_1), CL(o_2))+3 * d_{SC}(SC(o_1), SC(o_2))+ 4 * d_{EH}(EH(o_1), EH(o_2))+0.5 * d_{HT}(HT(o_1), HT(o_2))$$

Then, the similarity of two videos is defined as a Jaccard coefficient [7], where the cardinality of the intersection is the number of key frames that have a similar key frame in the other video. This is defined as the nearest-neighbor key frame that has a distance below a given threshold. Moreover, only images within a given time window are considered, so the similar scenes in different parts of the videos are not compared and also the number of distance computations was reduced. Both the parameters were set experimentally by testing the results on a small sample set.

2.3 Demo collection

Our test database consists of more than 2000 videos. During the extraction phase more than 4 million of image frames had to be processed. The indexed key frames collection is about 60 000 images. The whole process of extraction on a personal computer with Intel Core2 family CPU took 1275 hours of CPU time. The most time-consuming part of the process was the extraction of key frames that took about 1120 hours of CPU time and the extraction of visual descriptors lasted about 85 hours. The video resolution is mainly 720 x 576 pixels.

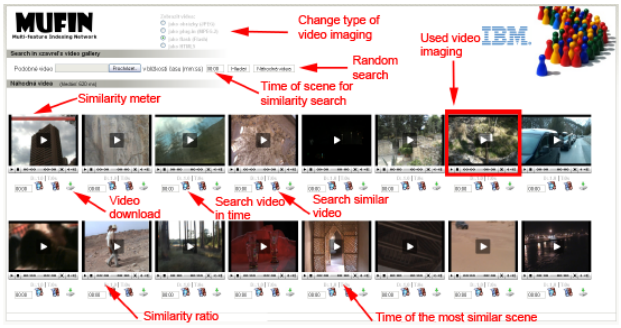

Figure 2: User interface example

2.4 Video visualization

Our system offers several ways of video visualization. The videos can be shown as images (timeline of key frames), as an embedded player, Flash player or HTML5 video tag.

Users have several possibilities of searching for videos. A user can upload his/her own video and search for a similar video as a whole or for a particular scene. Also a single image can be used instead of a whole video to search for key scenes. Then it is possible to use an iterative search of currently found videos - also based on a whole video or a single image specified using the video time. Finally, the interface allows to get a random selection of videos. The demo is available on webpage: **http://mufin.fi.muni.cz/videos**

3. ACKNOWLEDGMENTS

This work has been partially supported by the national research projects GACR 201/08/P507 and GACR 201/09/0683.

4. REFERENCES

[1] M. Batko, V. Dohnal, D. Novak, and J. Sedmidubsky. Mufin: A multi-feature indexing network. *Similarity Search and Applications, International Workshop on*, 0:158–159, 2009.

[2] A. Dorado, J. Calic, and E. Izquierdo. A rule-based video annotation system. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(5):622 – 633, may 2004.

[3] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995.

[4] Y. Liu and F. Li. Semantic extraction and semantics-based annotation and retrieval for video databases. *Multimedia Tools Appl.*, 17(1):5–20, 2002.

[5] B. S. Manjunath, P. Salembier, and T. Sikora, editors. *Introduction to MPEG-7: Multimedia Content Description Language*. Wiley, April 2002.

[6] H. Miyamori and S. ichi Iisaku. Video annotation for content-based retrieval using human behavior analysis and domain knowledge. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:320, 2000.

[7] P. Zezula, G. Amato, V. Dohnal, and M. B. atko. *Similarity Search: The Metric Space Approach*, volume 32 of *Advances in Database Systems*. Springer-Verlag, 2006.