Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Identical-video retrieval using the low-peak feature of a video's audio information

Myoung-beom CHUNG, Il-ju KO
(*Department of Media, Soongsil University, Seoul 156-743, Korea*)
E-mail: {nzin, andy}@ssu.ac.kr

Received July 31, 2009; Revision accepted Dec. 4, 2009; Crosschecked Dec. 7, 2009

Abstract: The recognition and retrieval of identical videos by combing through entire video files requires a great deal of time and memory space. Therefore, most current video-matching methods analyze only a part of each video's image frame information. All these methods, however, share the critical problem of erroneously categorizing identical videos as different if they have merely been altered in resolution or converted with a different codec. This paper deals instead with an identical-video-retrieval method using the low-peak feature of audio data. The low-peak feature remains relatively stable even with changes in bit-rate or codec. The proposed method showed a search success rate of 93.7% in a video matching experiment. This approach could provide a technique for recognizing identical content on video file share sites.

Key words: Video retrieval, Video DNA, Audio signal processing, Audio feature extraction
doi:10.1631/jzus.C0910472 **Document code:** A **CLC number:** TN912.34

1 Introduction

With continuing developments in the computing environment and expansion of the broadband Internet, computer users have been gaining access to information of increasingly vast breadth and depth, available in many different multimedia formats, including image, audio, and video. Accordingly, the demand for more efficient searches of multimedia data has been increasing. Early search systems rely on text-based indexing methods that manually analyze and sort video files by title or content according to previously determined categories. These manual methods, however, cannot keep up with the vastly increasing amounts of information to be searched, and thus image-content-based search systems have been explored as a more efficient and advanced alternative. Content-based search systems analyze the given information mathematically to isolate in numeric values some representative features of the information, and afterwards catalog the data based on a systematic indexing method. This type of search system, when well-designed, can swiftly and accurately extract unique aspects of given information and present these to users in an understandable, usable format.

Video search systems have broadly followed the research trend described above. Early approaches examine only basic information such as the file name, size, and watermark. Over time, developments in image-processing technology have led to the rise of content-based search methods primarily focused on evaluating video image frames, which usually account for over two-thirds of the data of a given video. The features of image frames used in a content-based search system include color, shape, and texture (Swain and Ballard, 1991; Mehrotra and Gray, 1995; Manjunath and Ma, 1996; Huang *et al.*, 1997; Kaplan *et al.*, 1998). Research has proceeded not only along the lines of improved search algorithms, but also toward the creation of new features and similarity measures based on color, texture, and shape. One interesting addition to the set of features comes from

the MPEG-7 standard (Pereira and Koenen, 2001). The new color features, such as red-green-blue (RGB) and maximum likelihood (ML) color spaces, have specific benefits in areas such as lighting invariance, intuitiveness, and perceptual uniformity (Gevers, 2001). Sebe and Lew (2001) used a quantitative comparison of influential color models; Jafari-Khouzani and Soltanian-Zadeh (2005) proposed a new texture feature based on the Radon transform orientation, which has the significant advantage of being rotationally invariant; and Srivastava *et al*. (2005) described some novel approaches to the learning of shapes.

Technologies based on the image frames of video, however, are prone to erroneous classification of identical videos as different when they have merely been altered in resolution or converted using a different codec. Most technologies use features extracted from key frames selected by scene change detection. If the resolution or the codec of the video changes, however, the number of selected scene change frames and the position of the key frames will differ. For this reason, technologies based on the image frames of a video do not perform well in the search for identical videos.

Thus, in this paper we present an identical-video retrieval technique that relies on the audio information of a video. The audio information of a video can be sampled and quantized for pulse-code modulation (PCM) data, from which we can calculate the peak information. The peak information retains a similar wave form even with changes in bit-rate or codec (Chung *et al.*, 2007). Hence, we propose the extraction of a feature called low-peak from the peak information of a video, as it is in no way related to the resolution of the video, and tends to show similar values regardless of bit-rate or codec changes. We can therefore search for identical videos using this feature even if the video type changes. To confirm that the proposed method is effective, we carried out an experimental search for identical videos with a success rate of 93.7%. This result shows an improvement of 24% over the existing search method based on mel-frequency cepstral coefficients (MFCCs). Consequently, the proposed feature can provide an identical-video search system for video sharing sites such as YouTube, Yahoo Video, or MySpace.

2 Previous work

In a content-based video search, the volume of image frame data is greater than that of the text or image data, because it contains 15–30 frames per second. Therefore, it cannot be used for general indexing or searching of video systems. As a result, researchers often search for a key frame for each sequence. The key frame is the representative scene in a shot, and we search for scene change frames to extract the key frame from each sequence (Idris and Panchanathan, 1997; Lupatini *et al.*, 1998; Yusoff *et al.*, 1998; Pickering and Ruger, 2003). A shot means a set of frames taken from one camera's viewpoint until the scene jumps to another camera's viewpoint (Fig. 1). All frames within a shot show a high similarity to one another. For example, in Fig. 1, A, A, A and A′, A′, A′ are similar frames. The sequence of these frames is a scene. In contrast, A′, A′, A′ and B, B, B are not similar, reflecting a change from one camera's view to another's. At that point, the scene of the video data changes from A′ to B. Hence, the first B frame is a scene change frame.

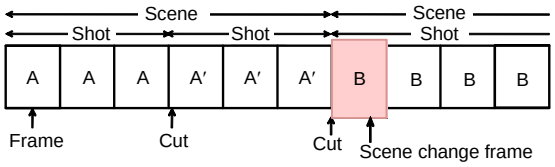

Fig. 1 Image sequence structure and scene change frames of video data

Researchers have proposed various methods of scene change detection. Methods that use only one feature have compared frames using such features as differences in pixel values, differences in histograms, comparison of edges, and block similarity (Stricker and Orengo, 1995; Zabih *et al.*, 1995; Eom and Choe, 2007; Kim and O′Connor, 2008). These single-feature methods perform well for single-setting videos such as sports or news, but have problems adjusting to videos that include a range of settings. To solve these problems, scene change detection methods using a mix of features have been proposed. Vadivel *et al.* (2008) proposed a method that generates histograms from the hue, saturation, and value (HSV) color space and texture. Lee *et al.* (2000) used a *k*-means algorithm based on differences in stet pixel

values and histograms, and Chung *et al*. (2009) used a mixed color histogram and the Karhunen-Loeve transform (KLT) algorithm. Scene change detection based on video images, however, often fails to detect identical scene change frames if the codec or resolution has been changed, as shown in Fig. 2.

Fig. 2 Scene cut comparison of videos written using different codecs

(a) Written using DivX MPEG-4 Video v4 (OpenDivX) codec at 320×240 resolution with a frame rate of 29.97 frames/s; (b) Written using Microsoft MPEG-4 Video v2 codec at 320×240 resolution with a frame rate of 29.97 frames/s

Figs. 2a and 2b are identical videos with the same run time. The scene change detection method uses the color histogram, texture, and KLT. Because they are encoded using different codecs, the numbers of scene change frames in Figs. 2a and 2b are different. The positions of the scene change frames are also slightly different. The dotted frames indicate exactly corresponding scene change frames whereas the other frames are unmatched frames.

In audio signal processing, conventional feature extraction involves MFCCs, zero-crossing rate (ZCR), and linear predictive coding (LPC) based on PCM data (Tzanetakis and Cook, 1999; McKinney and Breebaart, 2003). These features are often used for speech recognition, audio retrieval, and audio classification. MuscleFish of the Audible Magic Company originated the content-based audio classification genre (Wold *et al.*, 1996). Tzanetakis and Cook (2002) extracted an audio feature for the automatic classification of a music genre. They classified 10 genres such as blues, classical, country, and disco, and results showed a 70% success rate. Sung *et al.* (2008) then conducted a similar audio retrieval, proposing a method using MFCCs and dynamic time warping (DTW), which could find identical music content regardless of waveform changes. Following this method, Fig. 3 shows MFCC waveforms of the same codec and sampling rate, but with each waveform written using either a 64 or 128 or 192 kb/s MP3 audio. The waveforms of MFCCs for 64 kb/s and 128 kb/s show no differences, but at 192 kb/s they are somewhat different.

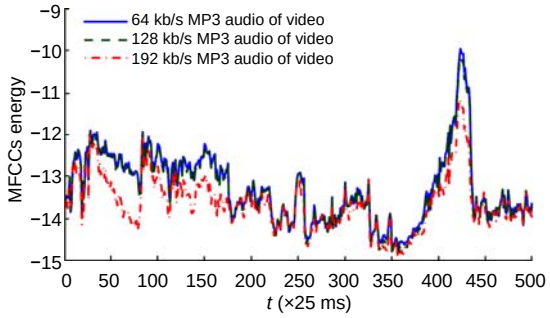

Fig. 3 Comparison of MFCC values of identical videos by bit-rate change

3 Extracting the low-peak feature and identical-video search

The audio information of a video can be sampled and quantized for PCM data. We can then calculate peak values from the PCM data. The formula for computing peak values is Eq. (1).

$$\begin{aligned}&\text{peak}[n]=\\&\max[\text{pcm}[1764(n-1)+1],\ \text{pcm}[1764(n-1)+2],\ ...,\\&\qquad\text{pcm}[1764(n-1)+1764]]\\&-\min[\text{pcm}[1764(n-1)+1],\ \text{pcm}[1764(n-1)+2],\ ...,\\&\qquad\text{pcm}[1764(n-1)+1764]].\end{aligned}\tag{1}$$

The number of 1764 discrete items of PCM data represents 20 ms of audio data at 44.1 kHz and Peak[n] is a peak value of n×20 ms. This number is widely used for peak data computation in audio signal processing. Before extracting the peak feature, we must detect a start-point for the audio data in the video file. This is necessary to overcome delays that occur when a video is changed by bit-rate or codec. Fig. 4 is an example of extraction start-points with delays caused by changes in bit-rate or codec.

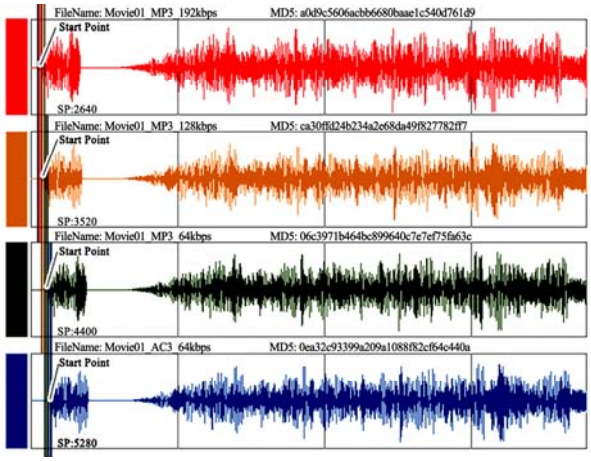

Fig. 4 Audio start-point delays caused by changes in bit-rate or codec

When a video comes in a different bit-rate or is written in a different codec, the start-point is gradually pushed out. The first waveform in Fig. 4 is 192 kb/s MP3 audio, and the start-point of the audio is in frame 2640. By contrast, the start-point of the second waveform at 128 kb/s is in frame 3520, and the start-point of the third waveform at 64 kb/s is in frame 4400. The fourth waveform of the identical video of a different codec with 64 kb/s AC3 audio has its start-point in frame 5280. Therefore, we need to detect the start-point for audio data in all video files and use the start-point detection method proposed by Kim *et al*. (2008).

Next, we explored an approach that focuses on spectral peak information, which remains relatively stable even with changes in the bit-rate or codec. Fig. 5 shows peak waveforms of videos with identical content and the same run time, although they were written using a different bit-rate or codec.

Each peak waveform in Fig. 5 is written using 64 kb/s MP3, 64 kb/s AC3, 128 kb/s MP3, or 192 kb/s MP3 audio. We can see that the vertical movements of the peak waveform are very similar in Fig. 5, even if the type of encoding has changed. We can, therefore, extract similar peak features from the audio peak information. In addition to the introductory background music that often serves as a prelude to a video, the main content of all video files contains various sounds that can be represented by unique wave patterns. These patterns contain silences or non-conversation sections that are low-peak sections. Low-peak sections are those sections where most peaks remain below a certain threshold α for the duration of the section, as shown in Fig. 6.

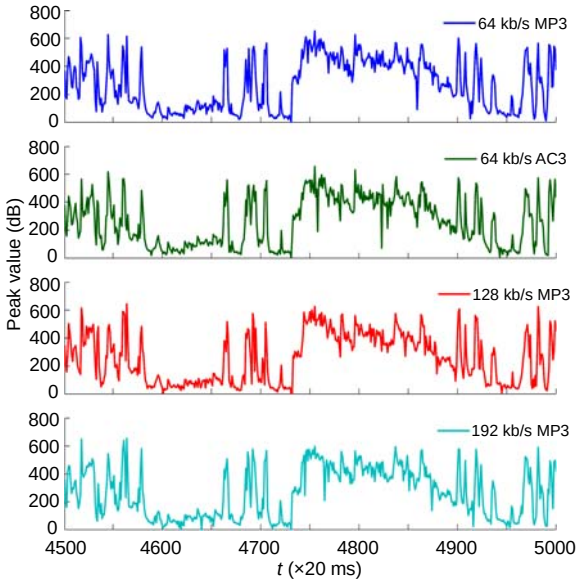

Fig. 5 Comparison of peak waveforms by bit-rate and codec change

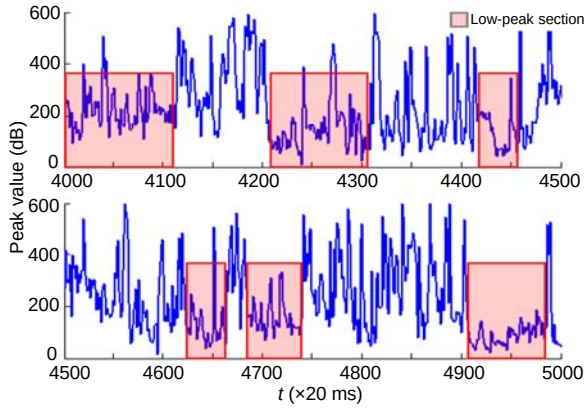

Fig. 6 Low-peak section that is silence or non-conversation of peak data

Fig. 6 shows the peak waveforms of 'Night at the Museum 2: Battle of the Smithsonian, 2009' in the introduction to the video from 80 s to 100 s. It is written using 64 kb/s 44.1 kHz MP3 audio. In Fig. 6, we can identify the boxed low-peak sections as silent sections. If the video content is identical, this section will appear in almost the same position even if the type of audio is different. Because the time basis of the x-axis is 20 ms, this will create too many features to search for, and it is difficult to use the low-peak section as a feature for identical-video searching. Hence, we change the timeline from milliseconds to seconds and count the number of low-peaks. Then, if the number of low-peaks is over a threshold β, we select this position as a low-peak feature, using Eqs. (2) and (3):

$$\begin{cases} \mathrm{LPCnt}[m]=\sum_{k=n}^{n+40} f(\mathrm{peak}[k]), \\ f(\mathrm{peak}[k])=\begin{cases} 1, & \mathrm{peak}[k]<\alpha, \\ 0, & \mathrm{peak}[k]\geq\alpha, \end{cases} \end{cases} \tag{2}$$

$$\begin{cases} \mathrm{LPF}[]=h(\mathrm{LPCnt}[m]), \\ h(\mathrm{LPCnt}[m])=\begin{cases} 1, & \mathrm{LPCnt}[m]\geq\beta, \\ 0, & \mathrm{LPCnt}[m]<\beta, \end{cases} \end{cases} \tag{3}$$

where LPCnt[m] is the number of low-peaks that are below threshold α, and LPF[] is the low-peak feature LPCnt[m] that is over threshold β. Then we can draw the low-peak features as in Fig. 7. Low-peak features are labeled 1 and non-low-peak features are labeled 0.

The graph is based on video data from 'Night at the Museum 2: Battle of the Smithsonian, 2009' written using 64 kb/s MP3, 64 kb/s AC3, 128 kb/s MP3, and 192 kb/s MP3. In Fig. 7, the boxes indicate low-peak features, which appear in almost the same positions even if the video has been changed by codec or bit-rate. Therefore, we can reliably extract the low-peak feature from the audio information of a video. A flowchart of the low-peak feature extraction process is shown in Fig. 8.

Table 1 shows a sample of the low-peak features of the video 'Night at the Museum 2: Battle of the Smithsonian, 2009'. We can see that the low-peak features are very similar.

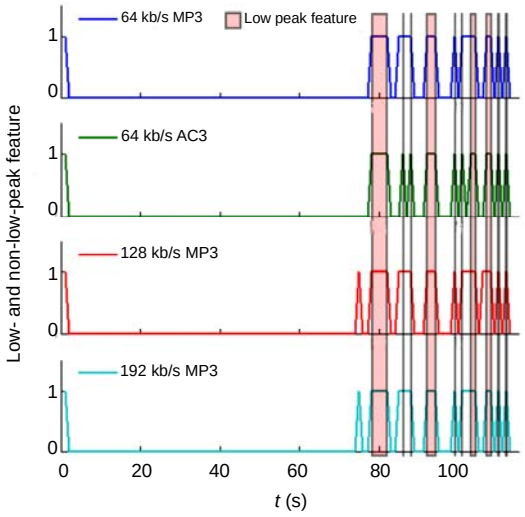

Fig. 7 Low-peak feature extraction from the peak data

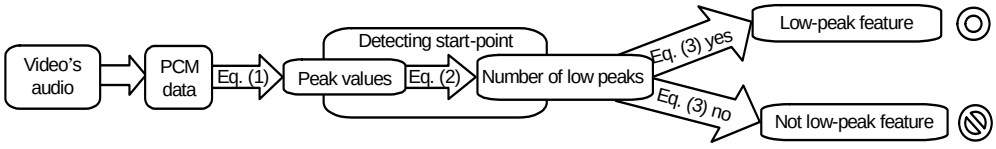

Fig. 8 Flowchart of the low-peak feature extraction from video's audio

Table 1 A sample of low-peak features of the same video at different bit-rates

File type	Low-peak feature	Feature count
64 kb/s, MP3	1, 78, 79, 80, 81, 82, 85, 86, 87, 88, 92, 93, 94, 99, 101, 102, 103, 104, 107, 108, 110, 112	22
64 kb/s, AC3	1, 78, 79, 80, 81, 82, 86, 88, 92, 93, 94, 99, 101, 103, 104, 107, 108, 110, 112	19
128 kb/s, MP3	1, 75, 78, 79, 80, 81, 82, 85, 86, 87, 88, 92, 93, 94, 99, 101, 102, 103, 104, 106, 107, 108, 110, 112	24
192 kb/s, MP3	1, 75, 78, 79, 80, 81, 82, 85, 86, 87, 88, 92, 93, 94, 99, 101, 102, 103, 104, 107, 108, 110, 112	23

In the proposed identical-video-search method, we record the low-peak features of the 64 kb/s MP3 to the database, because this is the most frequently used codec and the lowest bit-rate. In addition, we record the number of low-peak features per file. For example, if a video file has 22 low-peak features, the file records 22 items of data to the database (Fig. 9).

In Fig. 9, the database fields are composed of the index number (IDX), the serial number of file (number), video title (title), the value of low-peak feature (LPF), and run time of video (runtime). The identical-video search conducts a similarity search by comparing the search file data to the database record. Because the low-peak features will not match exactly when the video encoding is changed, the similarity search returns the result with the most closely matching features. Fig. 10 shows an example of a similarity search, where the search file features consist of 1, 78, 79, 99, 110, and 112. '4. Night at the Museum 2' is the search result, because it has the greatest number of matching features such as 1, 78, 79, 99, and 112. Therefore, we can search the identical video using information about the low-peak audio features of a video.

4 Experiment and analysis

We conducted the identical-video-retrieval experiment using the low-peak features of the audio as follows: firstly, we collected a set of 1000 video files whose genres were situation comedies, TV dramas, and movie films. The video files were downloaded randomly from Torrent or 4shared (http://www.4shared.com). Because the video files were written using various codecs, bit-rates, and resolutions, we re-encoded all the video files to 64 kb/s MP3. Then, following the low-peak feature extraction method described above, we recorded the low-peak features of these video files to a database. We set the threshold values as follows: threshold α=380 dB and threshold β=25. The threshold α=380 dB was the average of the peak values of the video dataset, and the threshold β=25 was half the value of the low-peak measurement section (50/s). We then ran an identical-video-retrieval search against the database using video files that matched the ones recorded in the database. The search video files had a same bit-rate and codec as the ones in the database. This experiment aimed to confirm whether the low-peak feature is unique to each

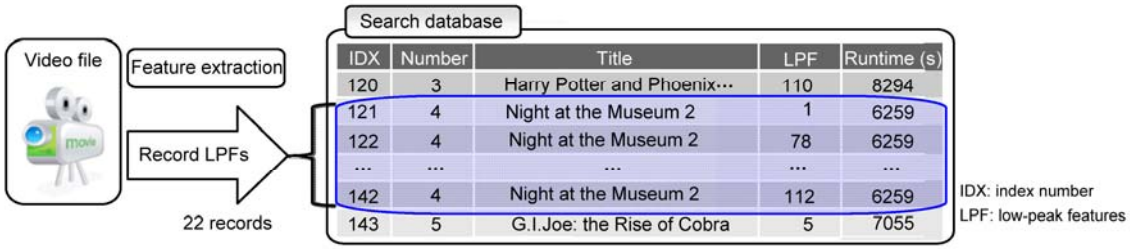

Fig. 9 Database structure of low-peak features for an identical video search

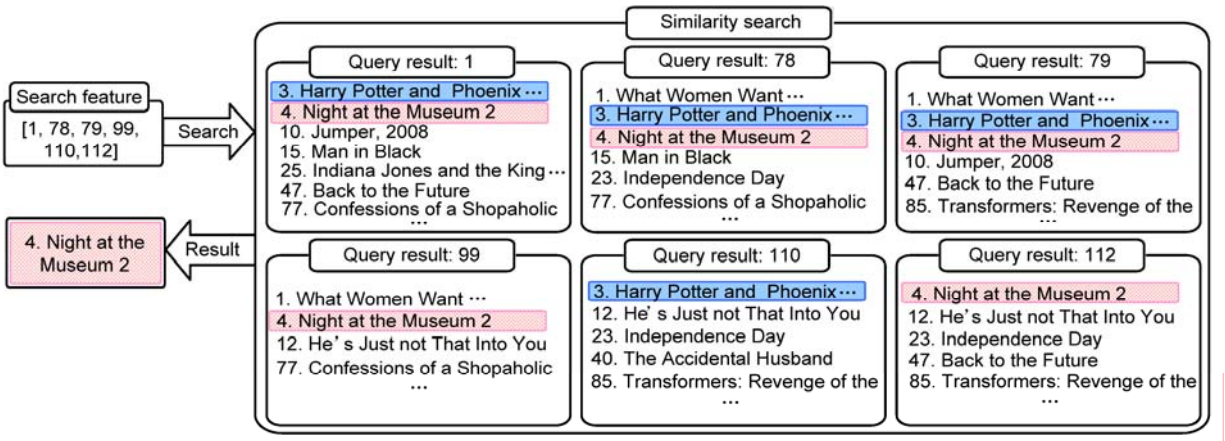

Fig. 10 Example of a similarity search using the low-peak audio feature

individual video. The accuracy of the search result was judged by whether the titles of the search file and the result file from the database were the same. To compare the proposed method against an existing search method, we tested the method of Sung *et al*. (2008) using MFCCs with DTW. The reason we compared this method is that it is rare for methods based on image frames to achieve identical-video retrieval after a change in encoding such as codec or bit-rate. Table 2 shows the results of this experiment.

Table 2 Experiment for validation of uniqueness using the low-peak feature method

Method	Correct	Error	No finding	Accuracy (%)
Low-peak feature	1000	0	0	100
MFCCs with DTW	1000	0	0	100

The results of both search methods showed a success rate of 100%, and we could see that the proposed low-peak feature of each video is unique. In the experiment, the computer we used was an Intel Core2duo E6300 with a 1.86 GHz Conroe and 2 GB RAM. We used the C++ programming language to extract the features and MySQL 5.1.39 for the database. The time of process for low-peak feature extraction from one video was 0.8254 s, while the time of process for MFCCs feature extraction was 1.0271 s. The time of process for identical-video searches from 1000 video files was about 0.0715 s in the experiment using the low-peak feature method, and the time of process for identical-video searches was about 0.0832 s in the experiment using MFCCs and DTW. The proposed method was slightly faster than the existing method using MFCCs and DTW for the feature extraction and identical-video search. Next, we re-encoded five distinct sets from the selected 1000 video files for the set of search files. The first set contained 1000 video files with the audio written using 128 kb/s MP3, and the second set contained 1000 video files with the audio written using 192 kb/s MP3. In both sets the codec was the same as those recorded to the database, but the audio bit-rate was different. The third set contained 1000 video files written using 64 kb/s AC3 audio, the same bit-rate as the database records, but with a different codec. The fourth set contained 1000 video files written using 128 kb/s AC3 audio. The codec and bit-rate of this set were different from the database records. The fifth set contained 1000 video files written using the same bit-rate and codec, but the video resolution was different. Table 3 shows the results of an identical-video search using the low-peak feature method and the MFCCs with DTW method.

Table 3 Identical-video search results using the low-peak feature and MFCCs with DTW with bit-rate, codec or resolution change*

Method	Setting	Correct	Error	No finding	Accuracy (%)
Low-peak feature	128 kb/s, MP3	900	99	1	90.0
	192 kb/s, MP3	904	95	1	90.4
	64 kb/s, AC3	964	35	1	96.4
	128 kb/s, AC3	916	83	1	91.6
	Different resolution	1000	0	0	100.0
	Total	4684	312	4	93.7
MFCCs with DTW	128 kb/s, MP3	607	393	0	60.7
	192 kb/s, MP3	595	405	0	59.5
	64 kb/s, AC3	682	318	0	68.2
	128 kb/s, AC3	601	399	0	60.1
	Different resolution	1000	0	0	100.0
	Total	3485	1515	0	69.7

* Each set contains 1000 video files; the resolution of the video files in the database records was 320×240 pixels, but the resolution of the fifth set was 640×480 pixels

Overall, our method yielded an accuracy rate of 93.7%. In contrast, the search that used the existing MFCCs and DTW method showed an accuracy rate of 69.7%. The proposed method showed an improvement of 24% over the existing method. The search method using MFCCs and DTW performed well for identical music content, but not well for an identical-video search. This is likely because the search method using MFCCs uses all the audio data, whereas the low-peak method uses only a small part of the audio data. Most of the errors were matching videos that were labeled ‘error’ because the search video’s low-peak value was less than that recorded to the database. The third set written using 64 kb/s AC3 codec, however, yielded a higher accuracy rate than those with different bit-rates. Thus, it could be concluded that the low-peak feature technique is robust even with codec changes.

5 Conclusion

We hereby propose an audio-based algorithm for identical-video retrieval. The low-peak feature of audio patterns is more effective for recognizing identical videos as it remains relatively stable with resolution or codec changes compared to image-analysis methods. This proposed technique can form the core of a copyright protection process by searching for content that requires protection. The technique could be used, for instance, in constructing copyright protection systems for services that offer video-sharing, such as YouTube, Yahoo Video, and MySpace.

The proposed feature can recognize an identical video file even if the video is changed in resolution, bit-rate, or codec. Gradually, however, users also need to be able to recognize similar videos that have been reprocessed. Thus, further research is needed to find an upgraded search method that can recognize reprocessed video files using the proposed feature. Furthermore, a system is needed to decide automatically whether the identical-retrieval result using the low-peak feature is correct or not.

References

Chung, M.B., Sung, B.K., Ko, I.J., 2007. Pretreatment for the problem solution of contents-based music retrieval. *J. Korea Soc. Comput. Inf.*, **12**(6):97-104 (in Korean).

Chung, M.B., Ko, I.J., Jang, D.S., 2009. Scene Change Detection Algorithm on Specific Movie. Proc. 11th Int. Conf. on Advanced Communication Technology, p.2286-2290.

Eom, M.Y., Choe, Y.S., 2007. Scene Change Detection on H.264/AVC Compressed Video Using Intra Mode Distribution Histogram Based on Intra Prediction Mode. Proc. 6th Conf. on Applications of Electrical Engineering, p.140-144.

Gevers, T., 2001. Color-Based Retrieval. *In*: Lew, M.S. (Ed.), Principles of Visual Information Retrieval. Springer-Verlag, London, p.11-49.

Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R., 1997. Image Indexing Using Color Corrlograms. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.744-749. [doi:10.1109/CVPR.1997.609412]

Idris, F., Panchanathan, S., 1997. Review of image and video indexing techniques. *J. Vis. Commun. Image Represent.*, **8**(2):146-166. [doi:10.1006/jvci.1997.0355]

Jafari-Khouzani, K., Soltanian-Zadeh, H., 2005. Radon transform orientation estimation for rotation invariant texture analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(6):1004-1008. [doi:10.1109/TPAMI.2005.126]

Kaplan, L.M., Murenzi, R., Namuduri, K.R., 1998. Fast texture database retrieval using extended fractal features. *SPIE*, **3312:**162-175. [doi:10.1117/12.298440]

Kim, C.Y., O′Connor, N.E., 2008. Low complexity video compression using moving edge detection based on DCT coefficients. *LNCS*, **5371**:96-107. [doi:10.1007/978-3-540-92892-8_11]

Kim, J.S., Sung, B.K., Ko, I.J., 2008. Music Starting-Point Detection Method Using Min-Wave-Shape. Korea Society of Computer Information Conf., p.137-141 (in Korean).

Lee, H.C., Lee, C.W., Kim, S.D., 2000. Abrupt Shot Change Detection Using an Unsupervised Clustering of Multiple Features. IEEE Int. Conf. on Accustics, Speech, and Signal Processing, p.2015-2018. [doi:10.1109/ICASSP.2000.859228]

Lupatini, G., Saraceno, C., Leonardi, R., 1998. Scene Break Detection: A Comparison. Proc. 8th Int. Workshop on Research Issues in Data Engineering Continuous-Media Databases and Applications, p.34-41. [doi:10.1109/RIDE.1998.658276]

Manjunath, B.S., Ma, W.Y., 1996. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**(8):837-842. [doi:10.1109/34.531803]

McKinney, M., Breebaart, J., 2003. Features for Audio and Music Classification. Proc. Int. Symp. on Music Information Retrieval, p.151-158.

Mehrotra, R., Gray, J.E., 1995. Similar-shape retrieval in shape data management. *Computer*, **28**(9):57-62. [doi:10.1109/2.410154]

Pereira, F., Koenen, R., 2001. MPEG-7: a standard for multimedia content description. *Int. J. Image Graph.*, **1**(3):527-546. [doi:10.1142/S021946780100030X]

Pickering, M.J., Ruger, S., 2003. Evaluation of key-frame based retrieval techniques for video. *Comput. Vis. Image Understand.*, **92**(2-3):217-235. [doi:10.1016/j.cviu.2003.06.002]

Sebe, N., Lew, M.S., 2001. Color-based retrieval. *Pattern Recogn. Lett.*, **22**(2):223-230. [doi:10.1016/S0167-8655(00)00092-1]

Srivastava, A., Joshi, S.H., Mio, W., Liu, X., 2005. Statistical shape analysis: clustering, learning, and testing. *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(4):590-602. [doi:10.1109/TPAMI.2005.86]

Stricker, M.A., Orengo, M., 1995. Similarity of color images. *SPIE*, **2420**:381-392. [doi:10.1117/12.205308]

Sung, B.K., Chung, M.B., Ko, I.J., 2008. A feature based music content recognition method using simplified MFCC. *Int. J. Princ. Appl. Inf. Sci. Technol.*, **2**(1):13-23.

Swain, M.J., Ballard, D.H., 1991. Color indexing. *Int. J. Comput. Vis.*, **7**(1):11-32. [doi:10.1007/BF00130487]

Tzanetakis, G., Cook, P., 1999. Multifeature Audio Segmentation for Browsing and Annotation. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, p.103-106. [doi:10.1109/ASPAA.1999.810860]

Tzanetakis, G., Cook, P., 2002. Musical genre classification of

audio signal. *IEEE Trans. Speech Audio Process.*, **10**(5):293-302. [doi:10.1109/TSA.2002.800560]

Vadivel, A., Sural, S., Majumdar, A.K., 2008. Temporal video segmentation using a colour-texture histogram. *Int. J. Signal Imag. Syst. Eng.*, **1**(1):78-87. [doi:10.1504/IJSISE.2008.017777]

Wold, E., Blum, T., Keislar, D., Wheaton, J., 1996. Content-based classification, search and retrieval of audio. *IEEE Multim.*, **3**(3):27-36. [doi:10.1109/93.556537]

Yusoff, Y., Christmas, W., Kitter, J., 1998. A study on automatic shot change detection. *LNCS*, **1425**:177-189. [doi:10.1007/3-540-64594-2]

Zabih, R., Miller, J., Mai, K., 1995. A Feature-Based Algorithm for Detecting and Classifying Scene Breaks. Proc. 3rd ACM Int. Conf. on Multimedia, p.189-200. [doi:10.1145/217279.215266]