International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.6 (2015), pp.165-176
http://dx.doi.org/10.14257/ijmue.2015.10.6.17

A Practical Video Fragment Identification System

Xun Jin and Jongweon Kim

Dept. of Copyright Protection, Dept. of Content and Copyright
Sangmyung University, Seoul, Korea, Sangmyung University, Seoul, Korea
jinxun0110@gmail.com, jwkim@smu.ac.kr

Abstract

In this paper, we present a practical identification approach of video fragment for digital video files. Before analyzing the video content, we must decode it based on its encoding format first. In order to effectively identify the format of a fragment, a format classification is performed before the format identification. The methods of format classification and identification are discriminative subspace clustering (DiSC) and the K-nearest neighbor (KNN).Because of losing the meta-information, we add a maximum similar header (MSH) to the front of the fragment to recover the video content. We adopt a simple key frame detection method using standard deviation and mean value. Motion vectors of macro blocks are utilized to classify the video features for effectively identifying the video. Several edges of frames are accumulated and compose a video feature. The experimental results show the evaluations of the video format classification and identification, fragment recovery, and content identification.

Keywords: *Video Identification, Subspace Clustering, Fragment Recovery, Motion Vector*

1. Introduction

With the rapid growth of multimedia transmission and evolution of network technology, video distribution has become much more popular and easier than ever. In the meantime, a lot of problems about video distribution have been building up for recently. For instance, invasion of privacy of personal video, copyright infringement of movie or computer forensic of video evidence. It is not an issue when the video files are intact. However, in reality, most video files are stored or distributed in the form of fragments. For instance, the movie files are split into several fragments to be distributed through the Bittorrent network, or some video files may be inadvertently or deliberately damaged, only a small fragment remains. The purpose of the proposed approach is to identify those fragments and find out the original videos for the fragments.

Many researchers have presented numerous methods of video identification as in [1-5]. However, those methods are performed under a hypothesis that the video content is extracted intactly. In practical terms, most arbitrarily split fragments are hard to be decomposed to get the inside content. Because all the video files are highly compressed and encoded, we have to decompress and decode them based on their compression and encoding methods first.

There are many types of compression and encoding methods, namely formats. In order to accurately recognize the format of a fragment, we classify the format features of numerous video samples in advance. Researchers have presented several methods of format classification based on statistical features, such as Shannon entropy, byte frequency distribution, N-gram, and Hamming weight as in [6-18]. It is difficult to achieve high precision of format classification only using statistical features, because all the video files involve high entropy coding. In this paper, we use a combination of statistical and structural features as the format feature as described in [19]. In [19], it was

ISSN: 1975-0080 IJMUE

used as audio format feature, however in this paper, we apply it to the video approach. The format classifier uses a kind of subspace clustering method, called discriminative subspace clustering as described in [20].The well-known K-nearest neighbor is employed in the procedure of format identification.

A video fragment indicates that it is randomly split from a video file. Because of losing the important meta-information, most fragments fail to be extracted the inside content. Therefore, we utilize a method of fragment recovery to recover the video content as in [19]. We collect numerous file headers from video samples and form a group of headers in advance. In the procedure of fragment recovery, the maximum similar header (MSH) is selected based on the identified format for recovering the fragment.

Because of the high capacity of video frames, it is inefficient to identify the entire frames of a video file. Hence, identifying a few of frames and key frame is a more desirable solution. Several types of video feature can be extracted from video data, such as color layout, edge, motion vector and binary pattern [1-5]. Two features are used in this paper: accumulated Canny edges [21-25] and motion vectors [26, 27]. The accumulated edges represent combination of temporal information and frame feature. We define the accumulated edges as major feature and the motion vectors as minor feature. We use minor features to classify the major features in advance for effective identification later.

Figure 1 shows the overall system view of the proposed approach. Before identifying a fragment, we set up a format feature database, video feature database and header database once for all. Once we receive a fragment, the procedure of format identification will be activated. Then the fragment will be recovered using the MSH. After decoding the fragment, the video content will be identified using motion vectors and accumulated edges.

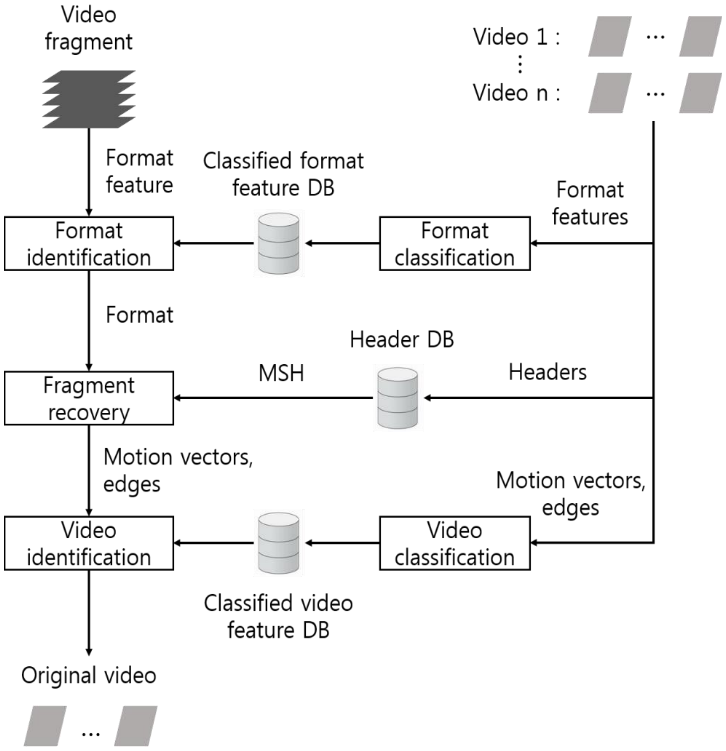


Figure 1. Overview of the Proposed System

The remainder of this paper is organized as follows. In Section II, we review the method of format feature extraction and introduce the format classification and identification. In Section III, we give an overview of fragment recovery. In Section IV, the algorithms of video classification and identification are presented. The evaluation results of format classification and identification, fragment recovery, and video identification are shown in Section V. Finally, in Section VI, we draw the main conclusions of this study.

2. Video Format Classification and Identification

2.1. Format Feature Extraction

The pattern recognition can be divided into two major categories: statistical and structural approaches. A statistical approach uses quantitative features, however, a structural approach uses morphological features [28]. Recently, most of researches in format feature extraction are statistical approaches, such as Shannon entropy, byte frequency distribution, and N-gram [6-18]. Because of all the video formats involve high compression method and entropy coding, those features are not suitable for distinguishing the formats of video fragments. Therefore, we employ a synchronization (SP) pattern which is a combination of structural and statistical features.

A video file consists of a large number of video frames. They are also called as blocks, packets or chunks in other formats. There is a special sync word in front of each frame. Each format of video has its own unique pattern for the sync word. In [12-13], researchers classified the formats with the quantity and existence of sync words. However, any non-video files can contain data with same value of the sync word. To solve this problem, we utilize the size information which can specify the length between two sync words, so that we can locate the next sync word. With the size information, the available sync words can be extracted. Then they compose a feature vector and are further processed by linear interpolation and quantization to obtain final format feature vector:

$$x = (v_1, v_2, \dots, v_D) \tag{1}$$

Where the D indicates the dimension of a feature vector.

2.2. Format Classification Using DiSC

The subspace clustering algorithms integrate clustering and feature evaluation to find clusters in different subspaces. They localize their search for the relevant dimensions and try to uncover the clusters which exist in multiple and possibly overlapping subspaces [20,29,30]. The DiSC mainly consists of four procedures: random processing, merging, quadratic classification, and weighted ensemble clustering.

We extract N format feature vectors x from N video samples. They compose a original data matrix X. To simplify the clustering problem, dimensionality reduction is applied in the first step. The X is projected into a lower dimensional ambient space with random projection matrix R as follows:

$$\tilde{X} = RX \tag{2}$$

A method of randomized local sampling is used for selecting subsets of data from union of subspaces to prevent over fitting and increase the robustness against noise and ensemble diversity. The number of random local clusters is defined as Q and each cluster consists of P points. We set $Q = 0.1N$ and $P = d + 3$, where d indicates the dimensionality of a subspace. Initially, Q random points are selected to form a random local clusters. Then composing the clusters around those points and their nearest neighbors.

Following the random processing, to provide more representative to the classifier, we merge the Q local clusters to K clusters. Since we need the distance between two clusters, mutual projection distance is used to measure the distance between the two cluster C_q and C_s as follows:

$$\mathrm{E}(q,s) = \frac{1}{2}\sqrt{\delta\left(x_q, L_s\right)} + \frac{1}{2}\sqrt{\delta\left(x_s, L_q\right)} \qquad (3$$

Where the $\delta\left(x_q, L_s\right)$ indicates the mean squared orthogonal distances to the subspace L_s. The clusters can overlap each other, however, before passing them to the classifier, the duplicate samples are removed. The distance is passed through the Radial Basis Function (RBF) kernel as follows:

$$\mathrm{A}(q,s) = exp\left(-\frac{\mathrm{E}(q,s)}{\alpha}\right) \qquad (4)$$

Where the α indicates the kernel width. The spectral clustering described in [31] is used for the merging. Then a set of points with consistent labels is obtained from each subspace, which will be used for the quadratic classifier. The quadratic classifier is a minimum Euclidean distance classifier for data modeled with projections onto the subspaces.

After the procedure of quadratic classification, we obtain the set of clusters which exhibit some diversity but also are highly correlated. Thus, those intermediate results need to be improved. In this case, the ensemble clustering is a desirable solution. For this purpose, the hybrid bipartite graph formulation (HBGF) described in [32] is applied for final clustering. The HBGF combines both the clustering information of ensembles and the pairwise information of points as the vertices of a bipartite graph. An ensemble of M clusterings with K classes is defined as $\{Y_m\}_{m=1}^{M}$. A connectivity matrix Z is created. It has $M \cdot K$ columns corresponding to the clusters and N rows corresponding to the points. Each row of Z is defined as follows:

$$Z(j, B_m) = 1(j,i,m), \quad j \in [1, N] \qquad (5)$$

$$B_m = Z(:, 1 + K(m-1): Km) \qquad (6)$$

Where $1(j,i,m)$ is a binary representation function. If point j has labe i in Y_m, the function takes the value 1, otherwise 0. The bipartite graph is enhanced with the information of subspace quality in a form of edge weights as follows:

$$w_m = \exp\left(-\sum_{i1}^{K}\sqrt{\delta(x_i, L_i)}/\alpha\right) \qquad (7)$$

Where the x_i indicates the points in Y_m with class label i. The L_i indicates the subspace. The w_m is applied to normalized row of Z as follows:

$$\hat{Z}(:, B_m) = w_m \frac{Z(:,B_m)}{\sum Z(:,B_m)} \qquad (8)$$

The final classified format feature vectors are obtained by applying the spectral clustering to the matrix $A_Z = \left(\hat{Z}\hat{Z}^T\right)^{\beta}$, where the parameter β is automatically selected to minimize the cluster distortion.

2.2. Format Identification Using KNN

In the procedure of format identification, the well-known KNN algorithms are used to identify the format of a fragment. The KNN algorithm measures the distance between a point and a set of points. We define the format feature vector of a fragment as a new point in D dimensional space and the classified format feature vectors as a set of trained points. We specify a positive integer k and select k points which are closest to the new point. The most common classification of those points is assigned to the new point. Figure 2 shows an example with $D = 2$.

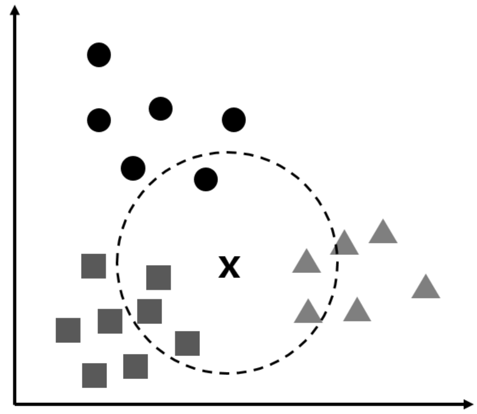


Figure 2. Format Identification Using KNN

3. Video Fragment Recovery

A video fragment is randomly split from a video file, which means it loses many video data, especially the important meta-information. Because of losing the meta-information, many video fragments failed to be decoded. To increase the accuracy of recovering the inside content, we add a MSH to the front of the fragment.

Many types of video file headers are collected to form a header group which has many parameters of various properties, such as frame rate, frame height, frame width, bit rate, etc. Each kind of parameter has several different values. The number of headers of the kth format is defined as H^k. The number of parameters in a header of the kth format is defined as D^k. We define the set of values which may appear in a parameter as $V^{k,d} = \left\{v_1^{k,d}, v_2^{k,d}, \dots, v_{J^{k,d}}^{k,d}\right\}$, $(d \in [1, D^k])$, where the J indicates the number of values may appear in a parameter. We define the number of headers with value of $v_j^{k,d}, (j \in [1, J^{k,d}])$ in the dth parameter as $\mathcal{N}_j^{k,d}$. The probability of a value occurrence in $V^{k,d}$ is equal to $\mathcal{N}_j^{k,d} / H^k$ and is defined as $\mathcal{P}_j^{k,d}$. We find out a value $v_j^{k,d}$ which has the maximum probability$\mathcal{P}_{max}^{k,d}$. The headers which have that value are moved to the front of the header group. In other words, the headers are rearranged according to the descending order of the probability of value occurrence. According to the identified format of fragment, we search the corresponding headers in the group and the one which can make the fragment decodable first is the MSH.

4. Video Classification and Identification

Because of a large number of frames in a video file, the key frames are used for extracting the video features in general. Key frame detection is a preliminary step for most video applications about indexing, classification and identification. In [33], researchers employed histogram difference, mean and standard deviation to determine the key frame. In this approach, we use a simple and fast shot boundary detection method to extract key frames using the standard deviation and mean value. Several types of video features can be extracted from video data, such as color layout, edge, motion vector and binary pattern. In this paper, there are two types of video features: the accumulated Canny edges and motion vectors. The accumulated edges can represent combination of temporal information and frame features. The motion vectors can represent an object movement in consecutive frames. We define the accumulated edges as major features and the motion vectors as minor features. The minor features are used to classify the major features in advance for effective identification later.

First, each $M \times N$ color frame is converted to a grayscale frame defined as GF. Then we begin to detect key frames. The mean value and standard deviation of the each GF are calculated. If the differences of them between the ith and the $i+1$th frame are greater than predefined threshold values, the $i+1$th frame is defined as a key frame as follows:

$$\mu_i = \frac{1}{M \times N} \sum_{m=1}^{M} \sum_{n=1}^{N} GF_{mn} \tag{9}$$

$$\sigma_i = \left(\frac{1}{M \times N - 1} \sum_{m=1}^{M} \sum_{n=1}^{N} (GF_{mn} - \mu_i)^2 \right)^{\frac{1}{2}} \tag{10}$$

$$\text{key frame} = \begin{cases} \text{ture}, & |\mu_i - \mu_{i+1}| > \tau_\mu, |\sigma_i - \sigma_{i+1}| > \tau_\sigma \\ \text{false}, & \text{otherwise} \end{cases} \tag{11}$$

If we find a key frame, the grayscale frame is resized to 128×128. Generally, the adjoining frames are used for motion estimation by analyzing the objects movements. The purpose of motion estimation in video compression standard is to compress video data, whereas in our approach, it is to represent unique feature information of video data. Most motion vectors between two adjoining frames contain minimal information, such as a scene of conversation in a movie, no object is moving except mouth. If the two frames are far from each other, they can be different scenes or contain different objects. Therefore, we use the ith key frame and the frame before the $i+1$th key frame to extract motion vectors.

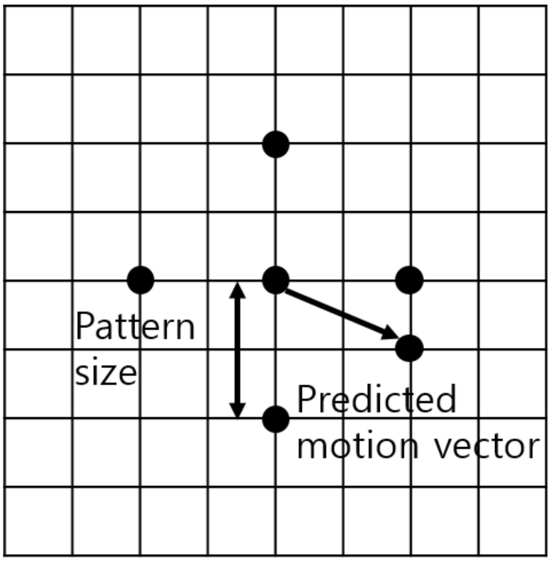


Figure 3. Adaptive Rood Pattern

The adaptive rood pattern search (ARPS) algorithm is used for searching the motion vectors in this approach as shown in Figure 3. The ARPS algorithm inspired by the fact that the general motions of objects are usually coherent. The motion vector of the current macro block is similar to those of macro blocks around it. The ARPS has a symmetrical rood shape. The size of the pattern implies the distance between the center point and vertex point. Because of most movements of camera are in vertical and horizontal directions, there are more motion vector distribution in these directions than that in other directions. Generally, we match a macro block with another based on the result of a cost function. The cost function is mean absolute difference which is the most popular and less computationally expensive method. We define a macro block size as 32×32, thus there are 16 macro blocks in a resized frame. Each motion vector of a macro block is defined as a node to construct a classification tree with 16 levels.

The Canny edge detector detects a wide range of edges using a multi-stage algorithm in an image. It locates sharp intensity changes and finds object boundaries. The Canny edge detection algorithm applies Gaussian filter to smooth the image to remove the noises, finds intensity gradients of the image and uses a double threshold to determine edges. It produces high accuracy in edge detection and shows good performance. Therefore we accumulate the Canny edges to obtain feature data. First, we extract the Canny edges from resized grayscale frames. The edges are represented in binary. We accumulate the edges of the nth key frame to those of the next γ frames to obtain AE. Then we calculate two threshold values from the normalized AE using Otsu's method. The Otsu's method minimizes the intra-class variance. It turns out to be the same as maximizing the inter-class variance. It is an effective threshold algorithm. We define the threshold values as quantization levels and quantize the normalized AE to obtain QE. The QE consists of three values as shown in Figure 5. Figure 4 shows the original key frame. Finally, the QE is classified based on the motion vectors of 16 macro blocks and sent to the corresponding nodes.

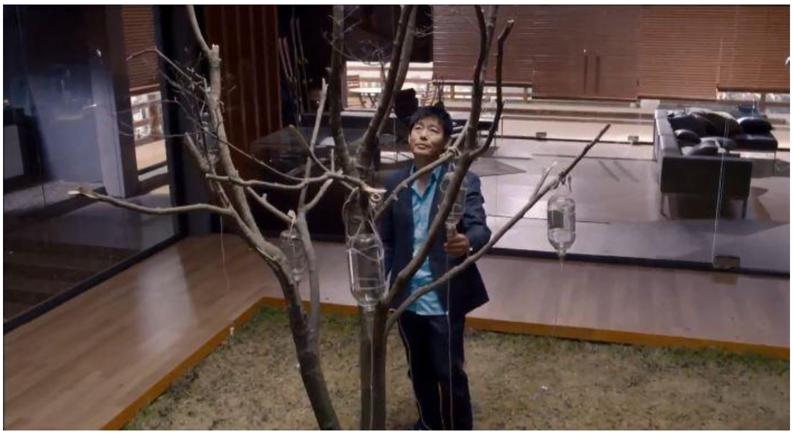


Figure 4. Original Key Frame

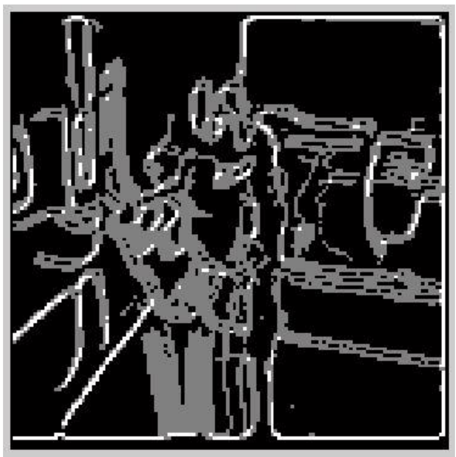


Figure 5. The Accumulated Edge after Normalization and Quantization

5. Performance Evaluation

The evaluation of the proposed approach consists of three kinds of experiments. One was to evaluate the format classification and identification error rates. Another was to evaluate the probability of video fragment recovery. The last one was to evaluate the precision of the video fragment identification. There were no file headers or footers in the test fragments. All the fragments were randomly split from the movie files such as action movie, drama and cartoon.

In the experiment of format classification, we evaluated three types of video formats: AVI, MP4 and RMVB. The samples used for classifying were randomly selected from 150 video files and 50 non-video files such as EXE, PDF and ZIP without header or footer. The samples used for identifying were randomly selected from other 100 video files and 30 non-video files. Figure 6 shows the normal probability density function of classification error rate of format classification. It illustrates the normal distribution with mean error rate of 0.25 and standard deviation of 0.19. Figure 7 shows the precisions of the format identification results. It can be seen from the figure that the precisions stay above 80% when the fragment size is greater than 256KB. If the fragment size is less than 256KB, the precision of RMVB is drop below 80%.

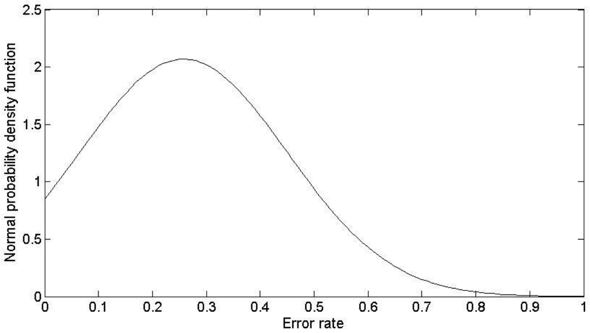


Figure 6. Normal Probability Density Function of Classification Error Rate

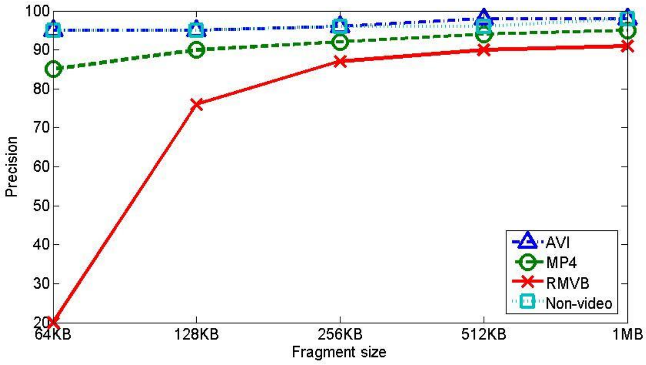
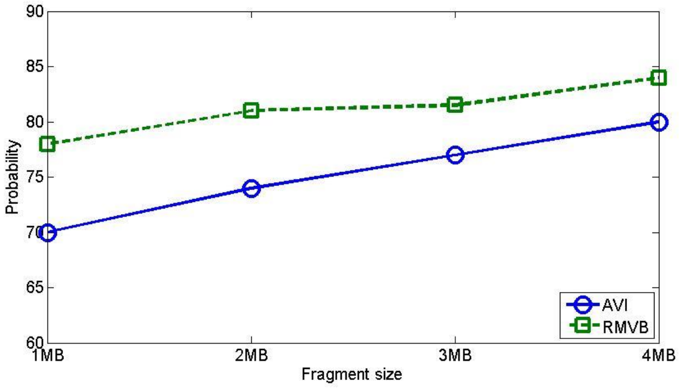
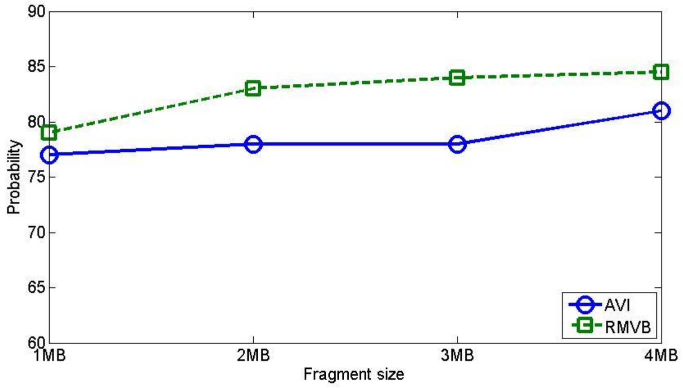


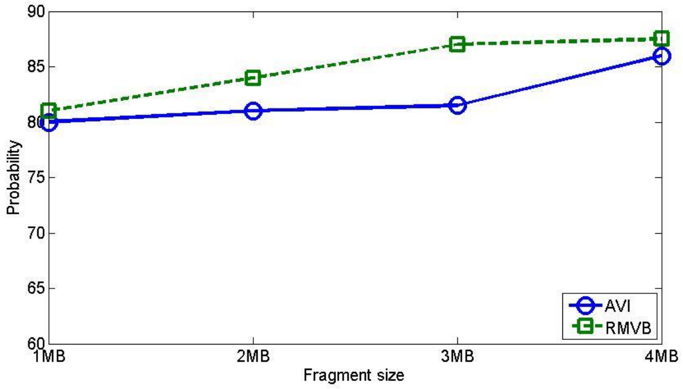
Figure 7. Precisions of the Format Identification



(a) Probability of Fragment Recovery with One Fragment



(b) Probability of Fragment Recovery with Two Fragments



(c) Probability of Fragment Recovery with Three Fragments

Figure 8. Probabilities of Fragment Recovery

Figure 8 shows the average probabilities of fragment recovery of AVI and RMVB with different fragment sizes after conducting dozens of experiments. The test samples are composed of 52 AVI and RMVB movie files. Figure 8 (a) and (b) illustrate the recovery probabilities of AVI maintain between 70% and 80% with one or two fragments, and that of RMVB maintain between 75% and 85%. Figure 8 (c) illustrates the recovery probabilities of the both formats maintain at around 85% with three fragments.

The precision of video identification is shown in Figure 9 with different fragment sizes. We evaluated 15 video files with the total duration of 20 hours in this experiment. The precisions of 4MB and 5MB fragments drop below 70 %, since there are usually no key frames in these fragments, such as the fragments of drama files. The precision of 8MB fragment is greater than 90%, showing good performance.

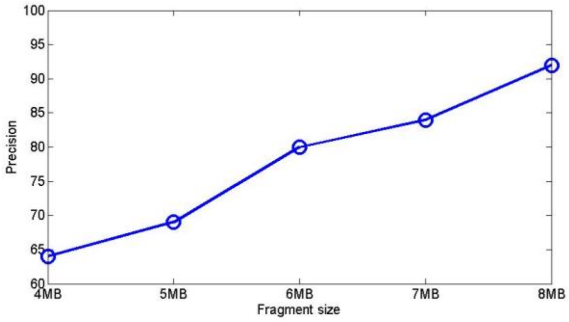


Figure 9. Precision of Video Identification

6. Conclusion

In this paper, we propose a practical system of video fragment identification. This system is composed of video format classification and identification, video fragment recovery and video classification and identification. The format feature is a synchronization pattern which combines the both statistical and structural features. The format classification algorithm is a kind of subspace clustering called DiSC. We use well-known KNN algorithm to identify the format of a fragment. In the procedure of fragment recovery, since the fragment loses many video data, especially the important meta-information, the MSH is added to the front of the fragment. The key frame is detected using the differences of the mean values and standard deviations. We extract two types of video content features from the frames, they are motion vectors and accumulated edges. The ARPS algorithm is used for searching the motion vectors from the ith key frame and the frame before the $i + 1$th key frame. The accumulated edges are normalized and quantized, then they are classified with motion vectors. The experimental results show the error rate of format classification, precision of format identification, probability of fragment recovery, and precision of video identification. The precision of format identification remains above 80% with the size of 256 KB, and that of video identification remains above 90% with the size of 8MB.

Acknowledgments

This research project was supported by the Ministry of Culture, Sports and Tourism (MCST) and the Korea Copyright Commission in 2014.

References

[1] J. Su, Y. Huang, H. Yeh and V. S. Tseng, “Effective content-based video retrieval using pattern-indexing and matching technique,” Expert Systems with Applications, vol. 37, **(2010)**, pp. 5068-5085.

[2] W. Hu, N. Xie, L. Li, X. Zeng and S. Maybank, “A survey on visual content-based video indexing and retrieval,” IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 41, **(2011)**, pp. 797-819.

[3] S. Kim, S. H. Lee and Y. M. Ro, “Rotation and flipping robust region binary patterns for video copy detection,” Journal of Visual Communication and Image Representation, vol. 25, **(2014)**, pp. 373-383.

[4] R. V. Martin and A. Bandera, “Spatio-temporal feature-based keyframe detection from video shots using spectral clustering,” Pattern Recognition Letters, vol. 34, **(2013)**, pp. 770-779.

[5] H. Zhou, A. H. Sadka, M. R. Swash, J. Azizi and U. A. Sadiq, “Feature extraction and clustering for dynamic video summarisation”, Neurocomputing, Vol. 73, **(2010)**, pp. 1718-1729.

[6] S. Fitzgerald, G. Mathews, C. Morris and O. Zhulyn, “Using NLP techniques for file fragment classification,” Digital Investigation, vol. 9, **(2012)**, pp. 44-49.

[7] W. C. Calhoun and D. Coles, “Predicting the types of file fragments”, Digital Investigation, vol. 5, **(2008)**, pp. 14-20.

[8] G. Conti, S. Bratus, A. Shubina, B. Sangster, R. Ragsdale, M. Supan, A. Lichtenberg and R. Perez-Alemany, “Automated mapping of large binary objects using primitive fragment type classification,” Digital Investigation, vol. 7, **(2010)**, pp. 3-12.

[9] M. C. Amirani, M. Toorani and A. A. Beheshti, “A New Approach to Content-based File Type Detection,” Proceedings - IEEE Symposium on Computers and Communications, **(2008)**, pp. 1103-1108.

[10] L. Zhang and G. B. White, “An Approach to Detect Executable Content for Anomaly Based Network Intrusion Detection,” 21th International Parallel and Distributed Processing Symposium, **(2007)**.

[11] S. Axelsson, “The Normalised Compression Distance as a file fragment classifier,” Digital Investigation, vol. 7, **(2010)**, pp. 24-31.

[12] V. Roussev and C. Quates, “File fragment encoding classification-An empirical approach,” Digital Investigation, vol. 10, **(2013)**, pp. 69-77.

[13] V. Roussev and S. L. Garfinkel, “File Fragment Classification-The Case for Specialized Approaches,” Systematic Approaches to Digital Forensic Engineering, Fourth International IEEE Workshop, Berkeley, CA, **(2009)**, pp. 3-14.

[14] V. A. Cicirello and W. C.Regli, “A flexible and extensible approach to automated CAD/CAM format classification,” Computers and Graphics, vol. 37, **(2013)**, pp. 484-495.

[15] M. Karresand and N. Shahmehri, “File Type Identification of Data Fragments by Their Binary Structure,” IEEE Information Assurance Workshop, **(2006)**.

[16] S. J. Moody and R. F. Erbacher, “SADI – Statistical Analysis for Data type Identification,” Third International Workshop on Systematic Approaches to Digital Forensic Engineering, **(2008)**, pp. 41-54.

[17] M. McDaniel and M. H. Heydari, “Content Based File Type Detection Algorithms,” 36th Annual Hawaii International Conference on System Sciences, **(2003)**.

[18] M. C. Amirani, M. Toorani and S. Mihandoost, “Feature-based Type Identification of File Fragments,” Security and Communication Networks, vol. 6, **(2013)**, pp. 115-128.

[19] X. Jin and J.Kim, “Audio Fragment Identification System,” International Journal of Multimedia and Ubiquitous Engineering, vol.9, **(2014)**, pp. 307-320.

[20] V. Zografos, L. Ellisy and R. Mester, “Discriminative Subspace Clustering,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **(2013)**, pp. 2107-2114.

[21] J. Canny, “A Computational Approach to Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, **(1986)**, pp. 679-698.

[22] L. Ding and A. Goshtasby, “On the Canny edge detector,” Pattern Recognition, vol. 34, **(2001)**, pp. 721-725.

[23] S. Vijayarani and M. Vinupriya, “Performance Analysis of Canny and Sobel Edge Detection Algorithms in Image Mining,” International Journal of Innovative Research in Computer and Communication Engineering, vol. 1, **(2013)**.

[24] M. Juneja and P. S. Sandhu, “Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain,” International Journal of Computer Theory and Engineering, vol. 1, **(2009)**.

[25] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk and B. Girod, “Robust Text Detection in Natural Images with Edge-Enhanced Maximally Stable Extremal Regions,” 18th IEEE International Conference on Image Processing, **(2011)**, pp. 2609-2612.

[26] Y. Nie and K. Ma, “Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation,” IEEE Transactions on Image Processing, vol. 11, **(2002)**, pp. 1442-1449.

[27] A. Barjatya, “Block-Matching Algorithms For Motion Estimation,” DIP 6620 Final Project Paper, **(2004)**.

[28] R. T. Olszewski, “Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data Thesis Committee,” Engineering Education, vol. 35, **(2001)**, pp. 1-125.

[29] L. Parsons, E. Haque and H. Liu, “Subspace Clustering for High Dimensional Data: A Review,” ACM SIGKDD Explorations Newsletter, vol. 6, **(2004)**, pp. 90-105.

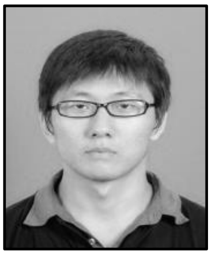
[30] C. Domeniconi, D. Papadopoulos, D. Gunopulos and S. Ma, “Subspace Clustering of High Dimensional Data,” Society for Industrial and Applied Mathematics, **(2004)**, pp.517-521.

[31] A. Ng, M. Jordan and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” Advances in Neural Information Processing Systems, **(2001)**.

[32] X. Z. Fern and C. E. Brodley, “Solving Cluster Ensemble Problems by Bipartite Graph Partitioning,” In Proceedings of the 21st International Conference on Machine Learning, vol. 36, **(2004)**.

[33] G. I. Rathod and D. A. Nikam, “An Algorithm for Shot Boundary Detection and Key Frame Extraction Using Histogram Difference,” International Journal of Emerging Technology and Advanced Engineering, vol. 3, **(2013)**, pp. 155-163.

Authors



Xun Jin, he received his B.S. degree in Computer Science and Technology from Fujian Agriculture and Forestry University, China, in 2011. He is currently pursuing the Ph.D. degree in Copyright Protection, Sangmyung University, Korea. His research interests are digital image/video watermarking, multimedia forensics, pattern recognition, digital signal processing, and information security.

JongWeon Kim, he received the Ph.D. degree from University of Seoul, major in signal processing in 1995. He is currently a Professor of Dept. of Content and Copyright at Sangmyung University in Korea. He has a lot of practical experiences in the digital signal processing and copyright protection technology at the institute, the industry, and College. His research interests are in the areas of copyright protection technology, digital rights management, digital watermarking, and digital forensic marking.