International Journal of Multimedia and Ubiquitous Engineering
Vol. 10, No. 1 (2015), pp. 89-100
http://dx.doi.org/10.14257/ijmue.2015.10.1.8

Optimized Identification Method for Digital Music Pieces in BitTorrent

Xun Jin[1] and Jongweon Kim[2]

[1]Dept. of Copyright Protection
Sangmyung University, Seoul, Korea
[2]Corresponding Author
Dept. of Intellectual Property
Sangmyung University, Seoul, Korea
jinxun@smu.ac.kr, jwkim@smu.ac.kr

Abstract

In this paper, we propose a methodology for identification of MP3 audio pieces downloaded by BitTorrent client program using cross-correlation of 12 chroma features extracted from audio signal. To prevent illegal distribution of copyrighted audio files from spreading over the internet, the file pieces split by torrent file must be detected as accurately as possible. In order to improve the probability of successfully identifying the audio pieces, we use Peak In Piece Position (PIPP) method which means the maximum peak appeared in a range corresponding to piece position and length is selected to reduce the probability of misidentification. The experimental results show probabilities of successfully identification of the pieces with different piece properties, comparisons between the proposed algorithms and conventional algorithm in terms of precisions and robustness of the proposed algorithms.

Keywords: *BitTorrent, Audio Piece, Chroma feature, Audio Identification, Cross-Correlation*

1. Introduction

BitTorrent protocol designed by Bram Cohen in 2001 is currently the most popular peer-to-peer (P2P) network for multimedia file sharing [1]. With the advent of BitTorrent protocol, internet users can easily share their video and audio files. Although the BitTorrent provides a highly efficient mechanism, the illegal use of BitTorrent on unauthorized distributing of copyright protected digital content is increasing rapidly. To prevent the illegal distribution of copyright-protected materials through the internet, we had batter extract the feature of the content and identify whether it is a copyright-protected content. But first of all, we have to analyze the files downloaded by the BitTorrent client whether it is available to extract the correct feature. For the purposes of transfer, files are split into fixed-size pieces which are all the same length except for possibly the last one which may be truncated. In this paper, we introduce four kinds of audio identification algorithms and compare performances of these algorithms with MP3 audio piece.

MPEG-1 more commonly referred to as MP3, is the most common audio file format used today. Most of audio files transferred through the BitTorrent are MP3 files. The structure of MP3 file consists of a number of independent frames and each frame has a header and side information which are used to decode the compressed audio data. If there is no header or side information in a piece, it is not only hard to make sure the format of audio file, but also difficult to find out where to start decoding. In this case, we have to analyze another piece. After the compressed data is decoded, then we extract the feature from the audio data.

ISSN: 1975-0080 IJMUE

There are several kinds of musical features can be extracted from audio data, such as timbre, tonality, rhythm and chroma [2], [3]. In this paper, we use the chroma feature to identify the audio pieces, because the chroma feature is a powerful and interesting representation for audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones of the musical octave. Our feature extraction method is based on the one described in [4], which is composed of beat tracking and chroma feature. A beat tracker is used to generate a beat-synchronous representation with one feature vector per beat and the representation of each beat is a normalized chroma vector which sums up spectral energy into 12 bins. The identification algorithm used in [4] is an approach of cross-correlating the entirety of the two feature matrices 12 times. But we found this method is not suitable for identifying the pieces of pop song files transmitted on BitTorrent network. Thus, we improved the identification algorithm to increase the probability of successfully identifying the pieces and found cross-correlation of each chroma feature once and two-dimensional (2D) cross-correlation of 12 chroma features gave better results. To reduce the probability of misidentification, we implemented PIPP method which removes peaks appeared out of correct range, and the experimental results show higher probabilities.

The overall system view of the proposed method is shown in Figure 1. After we read a downloaded audio piece, piece verification procedure will be activated to verify whether the piece contains valid audio frame. After verification, chroma feature will be extracted from decompressed audio data, if there exist valid frames, otherwise read next piece. Finally, audio identification is implemented by using chroma feature.

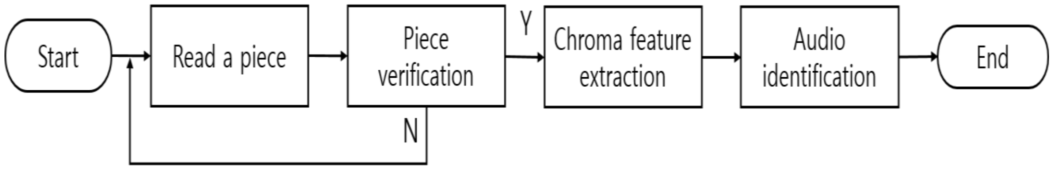


Figure 1. Overview of the Proposed System

The rest of this paper is organized as follows. To illustrate the unique characteristics of MP3 audio piece on BitTorrent in Section II, we present the working principle of BitTorrent technology and the structure of MP3 audio piece. Section III introduces the identification algorithms which are related to the presented methodology and presents comparisons of these algorithms with some songs. In Section IV, we evaluate the performances of the algorithms and discuss the evaluation of simulation results.

2. BitTorrent Audio Piece

2.1. BitTorrent Protocol

BitTorrent is a P2P file distribution protocol which is designed to allow efficient distribution of large files, such as video and audio files. To share a file, BitTorrent splits the file into several fixed size pieces. The size of a piece is usually with byte sizes of a power of 2 and typically between 32 KB and 16 MB. BitTorrent users called peers connect to each other directly to send and receive these pieces. However, there is a central server called tracker coordinates the action of the peers [5], [6]. It acts as an information exchange center and sends a randomly chosen subset of peers who have pieces of the file. The subset of peers is called swarm list. Firstly, there should be at least one file owner generates a torrent file which contains metadata, and this user is called initial seeder. The torrent file also contains SHA1 hash values of each piece to verify integrity of the pieces. After the torrent file is created, it is registered into a tracker and the file owner places it on websites to make the torrent file available to other BitTorrent users. After downloading the torrent file from the BitTorrent website, the downloader opens it with a BitTorrent client program which connects to the tracker and manages the transfer of the pieces.

2.2. MP3 Audio Piece

MPEG Audio Compression is one of many methods to compress audio in digital form trying to consume as little space as possible but keep audio quality as good as possible. An MP3 file is built up from a number of small parts called frames. These frames are independent items and each frame has its own header, side information and compressed data. To decode these compressed data, we must need several parameters to represent compression strength, compression method, compressed data size and location of compressed data and they are allocated in header and side information. The header is constituted by the very first 4 bytes in a frame. The first 12 bits of a header are always set and they are called frame sync. Therefore, we can search through the file for the first occurrence of frame sync. After the frame sync is found, we verify next parameters and check if the values are correct. Then we read the whole header if there is no invalid value. There are 6 parameters in the header are very important, and they are Version ID, Layer, Bitrate, Frequency, Padding bit and Channel Mode. After a piece of a file is downloaded, we read these parameters to verify if the piece is valid. Then side information parameters are used to decode the audio data.

The piece length specifies the nominal piece size, and is usually a power of 2. The piece size is typically chosen based on the total amount of file data. If the piece size is too small, resulting in a large torrent metadata file, and if the piece size is too large, causing inefficiency. The most common piece size for music file is less than 512 KB in practical applications. BitTorrent clients download pieces in a random order to increase the opportunity to exchange data. When a peer finishes downloading a piece, the client checks that the hash matches, then passes the piece to the piece verification module. It is almost impossible to split MP3 file from the beginning of the header, which means the beginning of the piece may be frame data, side information or header. Thus, when a piece is downloaded, we begin to search the header immediately. If there is an available header, we decode the piece and begin the process of audio identification, otherwise we have to analyze next piece. Figure 2 shows the structure of one of downloaded pieces.

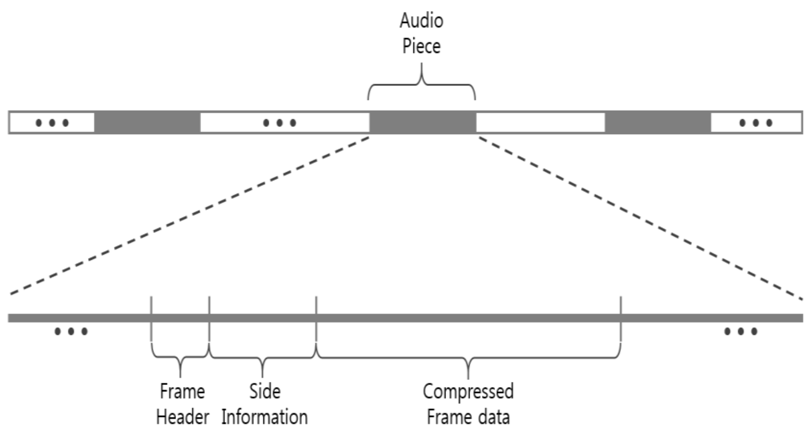


Figure 2. Structure of a MP3 Piece

3. Identification Algorithms

This section describes the conventional identification algorithm used in [4], and three more improved algorithms. It has been chosen to limit the discussion to research work focusing on audio identification , thus not including the extraction of the chroma feature, which has a focus on beat tracking, for more details of beat tracking please see [4, 7].

The structures of four algorithms are shown in Figure 3. The conventional algorithm consists of feature correlation, accumulation of coefficients, chroma shift, post-processing and maximum peak selection. After many experiments of analyzing chroma property, we found the chroma shift has a negative impact on distinguishing the features, hence the first streamlined algorithm is proposed. In the process of improving identification accuracy, we found that 2D correlation of 2D chroma pattern is more accurate than one-dimensional (1D) correlation accumulation of 12 1D chroma bins. Therefore, the second improved algorithm with 2D cross correlation using 2D FFT is proposed. In order to achieve the purpose of high accuracy, we utilize the PIPP method to optimize the identification algorithm.

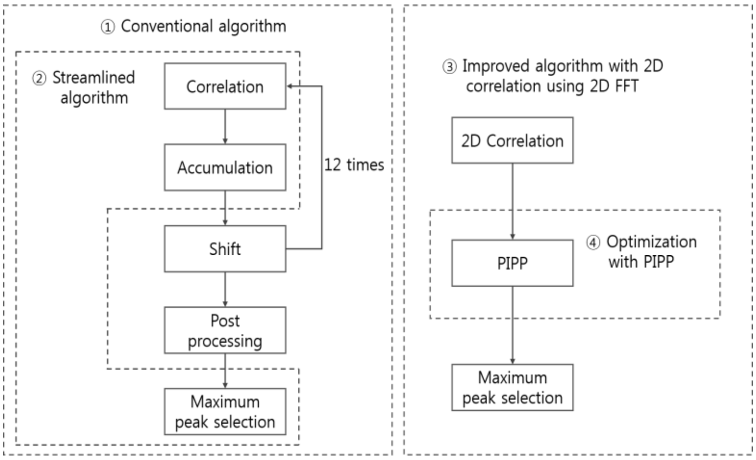
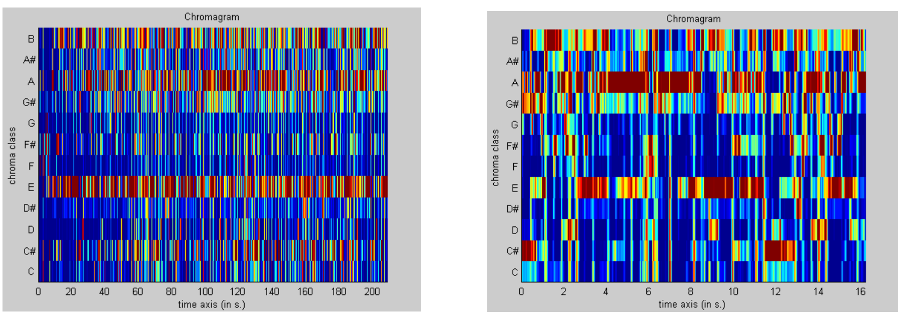


Figure 3. Structures of four Algorithms

3.1. Conventional Identification Algorithm

Useful musical information can be obtained from the distribution of chroma even without the absolute frequency. Chroma feature consist of 12 element vectors with each dimension representing the intensity which is associated with a particular semitone. Record a single feature vector per beat and 12 element chroma features are used to capture both dominant note and the broad harmonic accompaniment. Figure 4 shows chromagrams of a song and one piece of it.



(a) Chromagram of a Song (b) Chromagram of One Piece

Figure 4. Chromagrams of a Song and OnePiece of it

The conventional identification algorithm cross-correlates original audio feature F_o $(\mathrm{C} \times \mathrm{N}, \mathrm{C} = 12)$ and audio piece feature F_p $(\mathrm{C} \times \mathrm{n})$ one by one, which means each row vector of 12 chromas is cross-correlated. And it calculates the accumulation of 12 correlation coefficients rows, then shifts all the chroma features of F_o to the next chroma and calculates cross-correlation again. After 12 rounds of shifts, we can obtain correlation coefficients in 12 dimensions, and normalize them by shorter vector. The equation of the algorithm mentioned above is defined as follows:

$$R_{F_oF_p} = \begin{bmatrix} r(1,:) \\ r(2,:) \\ \vdots \\ r(12,:) \end{bmatrix}, \quad F_o = \begin{bmatrix} f_o(1,:) \\ f_o(2,:) \\ \vdots \\ f_o(12,:) \end{bmatrix}, \quad F_p = \begin{bmatrix} f_p(1,:) \\ f_p(2,:) \\ \vdots \\ f_p(12,:) \end{bmatrix}$$

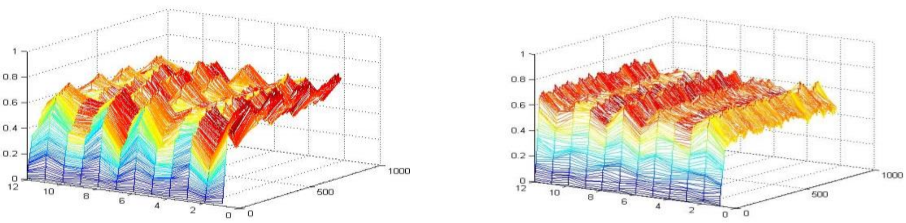
$$r(k+1,:) = \begin{cases} \frac{1}{n}\sum_{i=1}^{C} xcorr\left(f_o(i+k,:), f_p(i,:)\right), & i+k \le C \\ \frac{1}{n}\sum_{i=1}^{C} xcorr\left(f_o(i+k-C,:), f_p(i,:)\right), & i+k > C \end{cases} \tag{1}$$

$$k = 0,1,\cdots,C-1$$

where $xcorr$ denotes cross-correlation and $R_{F_oF_p}$ denotes correlation coefficients. In the post-processing step, a high pass filter is used to look for rapid variation on the best chroma which has the maximum peak in $R_{F_oF_p}$ and subtracts its mean value. At last, the maximum filtered peak is selected to represent the final correlation between audio piece and song. If there are L features of songs $\mathrm{S} = \{F_{o_1}, \ldots, F_{o_L}\}$ in database, then the one which has maximum peak in L peaks is identified to be the original audio.

3.2. Streamlined Algorithm

The similarity of songs will be increased when the number of songs L becomes larger, which means a few of chroma densities of different songs in S can be similar. The feature correlation between a little piece and its original song is not much difference from that between the piece and other songs using the conventional identification algorithm. For example, Figure 5 (a) shows the 12 chroma feature correlation coefficients $\boldsymbol{R_{F_{o1}F_{p1}}}$ between F_{p_1} which is the feature of a piece split from the 65th second to the 81th second of "With a little help from my friends" and its original song feature F_{o_1}. Figure 5 (b) shows the correlation coefficients $\boldsymbol{R_{F_{o2}F_{p1}}}$ between the F_{p_1} and F_{o_2} which is the feature of "Midnight train to Georgia". It can be seen from the two figures that $\boldsymbol{R_{F_{o2}F_{p1}}}$ is very close to $\boldsymbol{R_{F_{o1}F_{p1}}}$. Finally the F_{o_2} was misidentified as the most relevant audio after 12 rounds of shifts.



(a) Chroma Feature Correlation $\boldsymbol{R_{F_{o1}F_{p1}}}$ (b) Chroma Feature Correlation $\boldsymbol{R_{F_{o2}F_{p1}}}$

Figure 5. Feature Correlation between a Piece Feature F_{p_1} and its Original Song Feature F_{o_1} and other Song Feature F_{o_2} using Conventional Identification Algorithm

The maximum peak in $R_{F_{o2}F_{p1}}$ appeared in the 7th chroma after 6 shifts is just less than that in $R_{F_{o1}F_{p1}}$ appeared in the 1st chroma about 0.04. It means that the correlation between two different chroma features will be increased, when one of them shifts. Finally the maximum filtered peak between F_{p_1} and F_{o_2} is greater than that between F_{p_1} and F_{o_1}, and it leads to adverse consequences. To reduce such misidentification, we remove the procedure of shifting, eliminate the high pass filter and subtraction of mean value, so that the correlation results are much easier to distinguish. The maximum peak in $R'_{F_{o1}F_{p1}}$ is greater than that in $R'_{F_{o2}F_{p1}}$ about 0.2 using equation (2). Figure 6 shows the chroma feature correlation using improved algorithm after removing high frequency component of the coefficients.

$$R'_{F_oF_p} = \frac{1}{n}\sum_{i=1}^{C} xcorr\left(f_o(i,:), f_p(i,:)\right) \qquad (2)$$

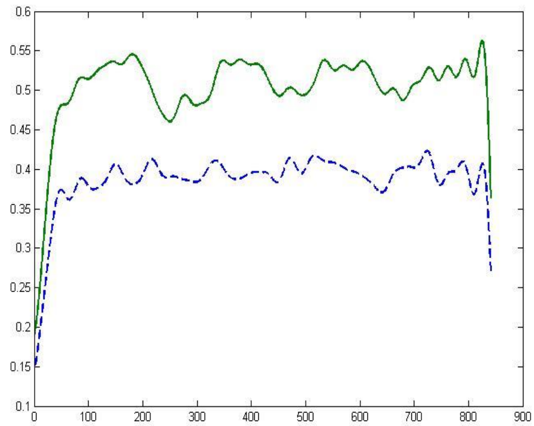
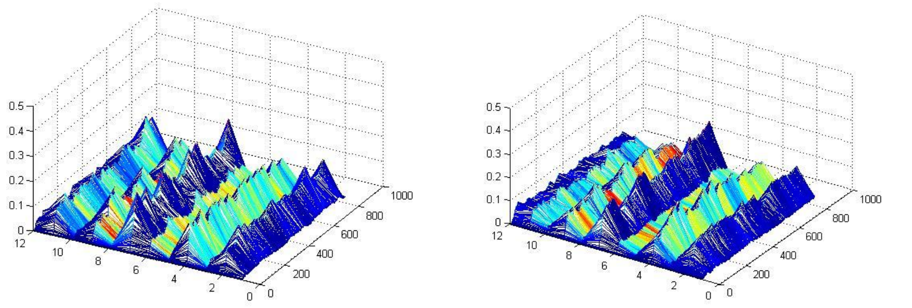


Figure 6. Feature Correlation between a Piece Feature F_{p_1} and its Original Song Feature F_{o_1} and other Song Feature F_{o_2} using Improved Algorithm

3.3. Improved Algorithm with 2D Cross Correlation Using 2D FFT

With the growth of audio files, the improved algorithm cannot satisfy high identification accuracy, which means about 20% of S are still misidentified when the L is equal to 50 and the piece size is equal to 256 KB. The density distribution of several chroma row vectors of 12 chromas of an audio piece can be usually close to that of other audio. This kind of phenomenon will have a negative impact to the probability of successfully identifying pieces. Figure 7 (a) shows the 12 chroma feature correlation coefficients between F_{p_3} split also from the 65th second to the 81th second of “Bridge over troubled water” and its original song feature F_{o_3}, and Figure 7 (b) shows the correlation between the F_{p_3} and feature of “Penny lover” F_{o_4} without the procedure of accumulation of 12 correlation coefficients rows.

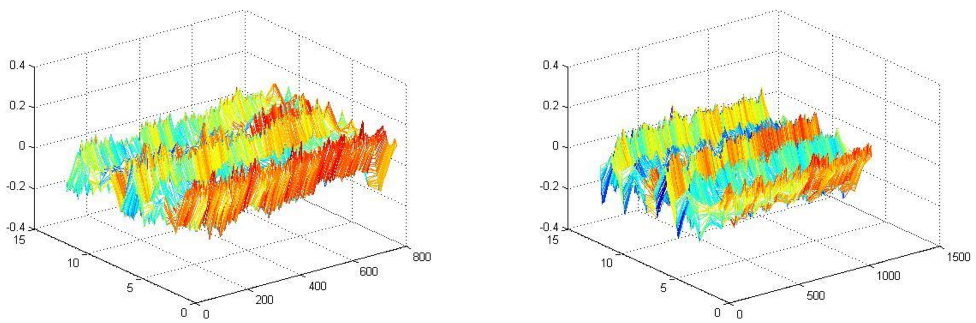


(a) Correlation between $\boldsymbol{F_{p_3}}$ and $\boldsymbol{F_{o_3}}$ (b) Correlation between $\boldsymbol{F_{p_3}}$ and $\boldsymbol{F_{o_4}}$

Figure 7. Feature Correlation between a Piece Feature $\boldsymbol{F_{p_3}}$ and its Original Song Feature $\boldsymbol{F_{o_3}}$ and other Song Feature $\boldsymbol{F_{o_4}}$ without the Procedure of Accumulation

The final maximum peak between F_{p_3} and F_{o_3} is less than that between F_{p_3} and F_{o_4} about 0.005, causing another misidentification. In order to avoid the occurrence of such error, we combine the 12 1D chroma bins to create a 2D chroma pattern, reduce the influence of similarity of 1D chroma bins on correlation and obtain more accurate identification results. The main advantage of the 2D correlation is to increase the spectral resolution by spreading overlapping peaks over two dimensions and can establish unambiguous assignments through correlation of bands, and determine specific sequential order of intensity changes. Using 2D FFT to compute 2D correlation can reduce the complexity and raise the matching speed. Figure 8 shows the 2D chroma pattern correlation results with 2D FFT based 2D cross correlation using equation (3). Because of the similarity of chroma row vectors, unfortunately, flux and reflux of peaks are appeared between correlated rows, but the maximum peak should be appeared on the first row as shown in Figure 8 (a), whereas in Figure 8 (b), it is appeared on the 11th row, and the maximum peak in $R''_{F_{o3}F_{p3}}$ is greater than that in $R''_{F_{o4}F_{p3}}$ about 0.01. With the proposed algorithm, the average probability of successfully identification is increased to 84%.

$$R''_{F_oF_p} = \frac{1}{n} R\left(IFFT2\left(FFT2(F_o).*\overline{FFT2(F_P)}\right)\right) \quad (3)$$



(a) 2D Correlation between $\boldsymbol{F_{p_3}}$ and $\boldsymbol{F_{o_3}}$ (b) 2D Correlation between $\boldsymbol{F_{p_3}}$ and $\boldsymbol{F_{o_4}}$

Figure 8. Feature Correlation between a Piece Feature $\boldsymbol{F_{p_3}}$ and its Original Song Feature $\boldsymbol{F_{o_3}}$ and other Song Feature $\boldsymbol{F_{o_4}}$ using 2D FFT based 2D Correlation

3.4. Optimized Algorithm using PIPP

To achieve the purpose of obtaining higher precision, we utilize PIPP method to 2D correlation to implement optimization. The method to generate the PIPP is motivated by the fact that the correct maximum peak should be appeared at the position corresponding to the point where the piece was split, when the 2D feature correlation between the piece and its original audio was calculated. Even though the maximum peaks of correlations between the piece and other songs are greater than the correct one, the probability of peak occurrence at the same position is very low, unless they are the same audio. This is the kernel of the PIPP method. Therefore, we skip the maximum peak appeared outside the piece position p_p.

$$\left\{p_p \in Len_c \left| \frac{Num_p \cdot Len_c}{Total_p} - \alpha < p_p < \frac{Num_p \cdot Len_c}{Total_p} + \alpha \right.\right\} \quad (4)$$

The Num_p denotes the downloaded piece number and the $Total_p$ denotes the total number of pieces in a file, and both of them are given by the BitTorrent client program when we are downloading the piece. The Len_c is the length of the correlation coefficients and the factor α denotes the range about the position of peak occurrence.

As a matter of fact, experiment results show that the maximum peak appeared outside the correct range occasionally even if correlating with original audio. It is for the reason that, the arbitrary splitting of audio pieces causes several gaps of a few tenths of a second between beat times of a piece and those of original file before extracting chroma feature. Therefore, the both distributions of chroma feature densities are different, so there is no feature segment which is exactly the same as that of the piece in the original feature. These will eventually lead to correlation degradation. In order to minimize such a false negative and correlation degradation, we remove the redundant audio which influence synchronization of beat times. The redundant audio is located at the beginning of the piece and its duration t_r is calculated as follows:

$$t_r = min\{b \in B | \mathrm{b} > t_b\} - t_b \quad (5)$$

$$t_b = \frac{Offset_p \cdot \left[b_N + \frac{1}{N}\sum_{n=1}^{N}(b_{n+1} - b_n)\right]}{Len_f} \quad (6)$$

The $Offset_p$ denotes the offset of a downloaded piece of a file and Len_f denotes the length of the file, and both of them are also given values. The B is beat times with size of N of one song stored in the database. The t_b is the duration of blank located in front of the piece. Without the redundant audio, we can eliminate lots of irrelevant audio and precisely find the desired audio.

4. Performance Evaluation

4.1. Experimental Data

In the previous section, four identification algorithms have been presented clearly. In this section we evaluate performances of the proposed algorithms. To carry out the comparisons, we selected a total of 300 songs from a wide range of genres including hip pop, classical, country, rock and folk with bitrate of 128 kbps and sample rate of 44.1 kHz. The evaluation was investigated in terms of probability of successfully identifying pieces, namely $Precision$ defined as follows:

$$Precision = \frac{Correct}{Returned} \times 100\%$$

where $Correct$ denotes the correct retrieval set, $Returned$ denotes the resulting audio set. To analyze the performance of identification under different sizes of pieces, the evaluation was conducted with piece sizes of 32 KB, 64 KB, 128 KB and 256 KB. In order to evaludate robustness of the proposed algorithms, we also identified the audio pieces after recompression attack with bitrate of 64 kbps, 112 kbps and 192 kbps. The playback time of decompressed audio of a piece corresponding to different piece size and bitrate is shown in Table 1.

Table 1. Playback Time of a Piece

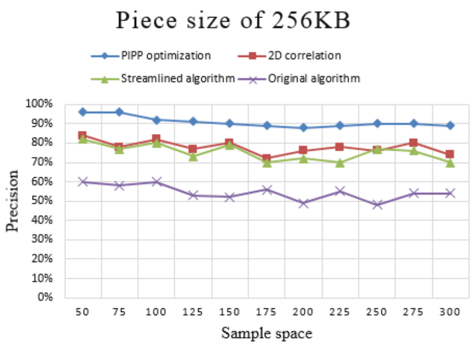
Bitrate	32 KB	64 KB	128 KB	256 KB
64 kbps	4s	8s	16s	32s
112 kbps	2.3s	4.5s	9s	18s
128 kbps	2s	4s	8s	16s
192 kbps	1s	2.5s	5s	10s

4.2. Results and Discussion

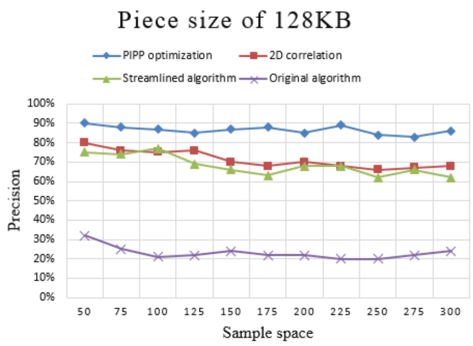
Basically, the evaluation was conducted on the purposes including: (1) the precisions for our proposed algorithms under different properties of pieces, (2) the comparisons of the proposed algorithms and the conventional algorithm in terms of precisions, (3) the robustness of the proposed algorithms for recompression attack. View the music database as the sample space of experiments in which a mash-up of 50 songs is selected at random. In each experiment, the size of sample space increased by 25 from 50 to 300. After many experiments, we found factor $\alpha = 40$ result in the best range.

Figure 9 shows average precisions of four algorithms with different piece sizes, and it shows the precisions of all the algorithms presented in this paper will not drop precipitously when the size of database is increased, especially from sample space of 150, the precisions remain steady. When the piece size is equal to 256 KB, most of the precisions of PIPP optimized algorithm are greater than 90% as shown in Figure 9 (a). Figure 9 (d) shows all the precisions of conventional algorithm drop below 6% when the piece size is equal 32 KB, whereas those of the streamlined algorithm and the algorithm with 2D correlation maintain between 40% and 70 %, and those of PIPP optimized algorithm are greater than 70%.

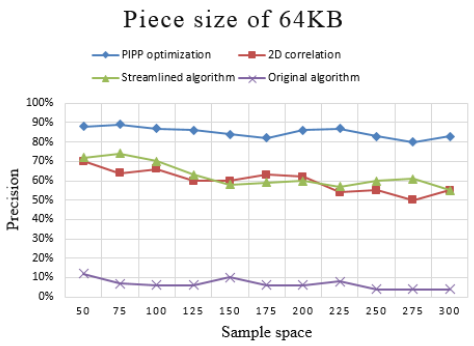
Figure 10 shows the precisions corresponding to different piece sizes when the size of sample space is equal to 300. With the reduction in piece size, the precision of conventional algorithm drops precipitously, whereas the other algorithms maintain steady. Obviously, these results reveal that the three more improved algorithms outperform the conventional algorithm significantly.



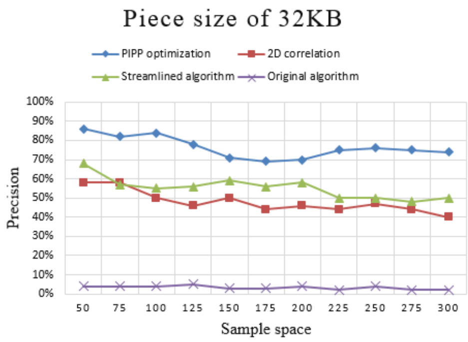
(a) Precisions with Piece Size of 256 KB



(b) Precisions with Piece Size of 128 KB



(c) Precisions with Piece Size of 64 KB



(d) Precisions with Piece Size of 32 KB

Figure 9. Average Precisions of Four Algorithms with Different Piece Sizes

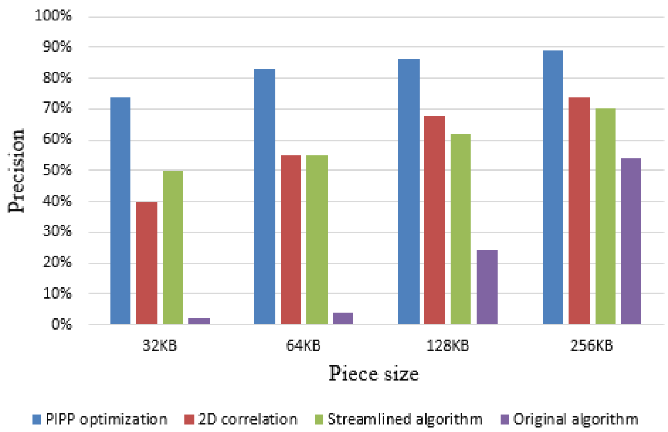
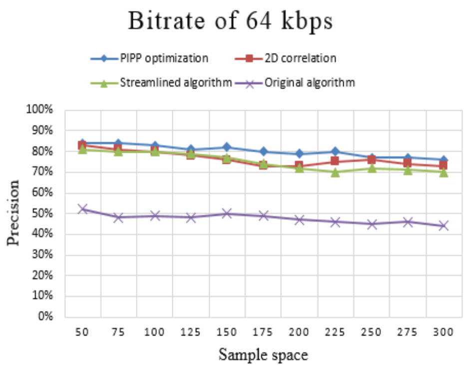
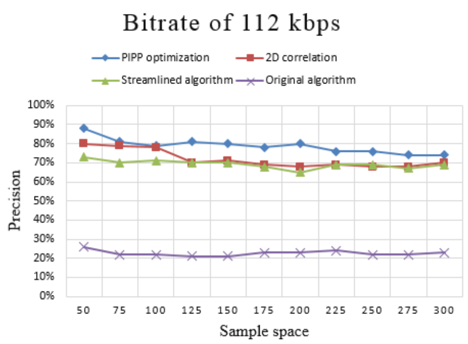


Figure 10. Precisions with Sample Space of 300

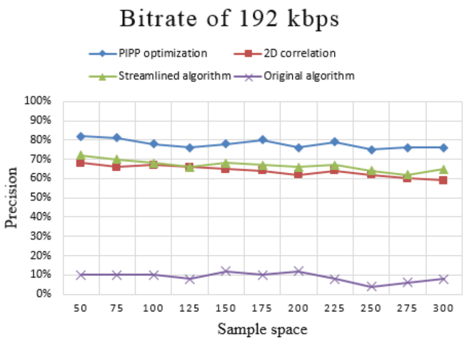
In practical application, internet users may distribute recompressed music file through the BitTorrent network. Thus we evaluated the robustness of proposed algorithms after recompression attack. Figure 11 shows the average precisions of the proposed algorithms with piece size of 128 KB after recompression with bitrate of 64 kbps, 112 kbps, 192 kbps. It can be seen from the figures that the precisions of conventional algorithm are not only instable but also less than 60%, whereas those of the streamlined algorithm and the algorithm with 2D correlation are maintained at around 70 %, and those of PIPP optimized algorithm are maintained at around 80%, showing good performance.



(a) Precisions with bitrate of 64 kbps



(b) Precisions with bitrate of 112 kbps



(c) Precisions with bitrate of 192 kbps

Figure 11. Average Precisions of Four Algorithms with Different Compression Strengths

A major factor that has a big impact on misidentification is a few seconds of silent audio. Now many songs contain silent part including the sample songs evaluated in our research. Once a piece is split around silent part, it is very difficult to identify original song correctly, because many other songs also have silent part.

5. Conclusion

In this paper, we proposed a methodology for identification of MP3 audio pieces downloaded by BitTorrent client program based on 12 chroma features. The pieces split by torrent file are randomly ordered, and passed to the piece verification module to verify whether the piece is valid by checking the important parameters of frame header. Then we decode the piece and extract the feature to identify the piece. The conventional matching algorithm is not suitable for identifying the pieces of pop song files transmitted on BitTorrent network. Thus we proposed streamlined version of conventional algorithm, improved algorithm with 2D cross correlation using 2D FFT and optimized algorithm using PIPP method to increase the identification accuracy. The experimental results show the precisions for our proposed algorithms under different properties of pieces and the comparisons of the proposed algorithms and the conventional algorithm in terms of precisions. We also evaluated the robustness of the proposed algorithms for recompression attack.

In the future work, we are going to improve the speed of classification because there are over 10 million songs in the world. We will study more formats of audio and construct an audio identification system to identify any formats of audio piece. We will also develop the proposed technology and apply it to the applications of forensic.

Acknowledgments

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

References

[1] P. Sharma, A. Bhakuni and R. Kaushal, “Performance Analysis of BitTorrent Protocol”, National Conference on Communications, NCC, **(2011)**; New Delhi, India.

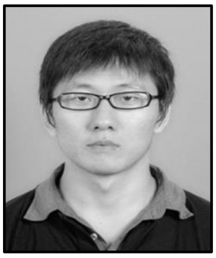
[2] O. Lartillot and P. Toiviainen, “A Matlab Toolbox for Musical Feature Extraction from Audio”, Proceedings of the 10th International Conference on Digital Audio Effects, **(2007)**; Bordeaux, France.

[3] T. Jehan, “Creating Music by Listening”, PhD thesis, MIT Media Lab, Cambridge, MA, **(2005)**.

[4] D. P. W. Ellis and G. E. Poliner, “Identifying ‘Cover Songs’ with Chroma Features and Dynamic Programming Beat Tracking”, IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, **(2007)**; Honolulu, HI.

[5] K. Bauer, D. McCoy, D. Grunwald and D. Sicker, "BitStalker: Accurately and efficiently monitoring bittorrent traffic", IEEE International Workshop on Information Forensics and Security, WIFS, **(2009)**; London.
[6] K. P. Chow, K. Y. Cheng, L. Y. Man, P. K. Y. Lai, L. C. K. Hui, C. F. Chong, K. H. Pun, W. W. Tsang, H. W. Chan and S. M. Yiu, "BTM - An Automated Rule-based BT Monitoring System for Piracy Detection", International Conference on Internet Monitoring and Protection, ICIMP, **(2007)**; San Jose, CA.
[7] D. P. W. Ellis, "Beat Tracking by Dynamic Programming", Journal of New Music Research, vol. 36, pp. 51-60, **(2007)**.

Authors



Xun Jin 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, digital signal processing, and information security.

JongWeon Kim received the Ph.D. degree from University of Seoul, major in signal processing in 1995. He is currently a Professor of Dept. of Intellectual Property 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.

