CHAPTER 9

Using Low-Order Auditory Zernike Moments for Robust Music Identification in the Compressed Domain

Wei Li, Bilei Zhu, Chuan Xiao and Yaduo Liu

Methods based on moments and moment invariants have been extensively used in image analysis tasks but rarely in audio applications. However, while images are typically two-dimensional (2D) and audio signals are one-dimensional (1D), many studies have showed that image analysis techniques can be successfully applied on audio after 1D audio signal is converted into a 2D time-frequency auditory image. Motivated by these observations, in this chapter we propose using moments to solve an important problem of audio analysis, i.e., music identification. Especially, we focus on music

Wei Li, Bilei Zhu, Chuan Xiao
School of Computer Science, Fudan University
Shanghai, P. R. China
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
Shanghai, P. R. China
e-mail: weili-fudan@fudan.edu.cn

Yaduo Liu
China Electric Power Research Institute
Beijing, P. R. China

Editor: G.A. Papakostas, *Moments and Moment Invariants - Theory and Applications*
DOI: 10.15579/gcsr.vol1.ch9, GCSR Vol. 1, ©Science Gate Publishing 2014

identification in the compressed domain since nowadays compressed-format audio has grown into the dominant way of storing and transmitting music.

There have been different types of moments defined in the literature, among which we choose to use Zernike moments to derive audio feature for music identification. Zernike moments are stable under many image transformations, which endows our music identification system with strong robustness against various audio distortions. Experiments carried out on a database of 21,185 MP3 songs show that even when the music queries are seriously distorted, our system can still achieve an average top-5 hit rate of up to 90% or above.

9.1 Introduction

Moments and moment invariants have been widely used as pattern features in a number of image analysis applications [2, 23]. In contrast, only very few works have been reported in using moment-based methods for audio problems (see [30] for example). However, while images are typically two-dimensional (2D) and audio signals are one-dimensional (1D), studies in the field of machine hearing have showed that image analysis techniques can be successfully applied on audio after 1D audio signal is transformed into a 2D time-frequency auditory image [25, 19, 7]. This suggests that, after time-frequency transformation, methods based on moments and moment invariants may also be powerful tools for analyzing audio.

In this chapter, we investigate using moments to solve an important problem of audio analysis, i.e., music identification. Music identification is a technique that helps users recognize unknown music from a short (typically a few seconds) and probably distorted music segment. The technique relies on audio fingerprinting, and by comparing the fingerprint extracted from the input music segment with those previously calculated and stored in a fingerprint database, a music identification system can identify the unknown music and return its metadata such as the title and the singer's name. To date a number of music identification algorithms have been published in the literature, and some of them have even been deployed for commercial use [3]. However, most of these existing algorithms operate on the PCM wave format, in spite of the fact that nowadays compressed-format audio, especially MPEG-1 Layer III (MP3) music has grown into the dominant way of storing and transmitting music. Therefore, in this chapter we focus on music identification in the compressed domain.

So far, there have been only a few works that perform music information retrieval (MIR) directly on the compressed domain. Liu and Tsai [18] calculated the compressed-domain energy distribution from the output of polyphase filters as feature to index songs. Lie and Su [16] directly used selected modified discrete cosine transform (MDCT) spectral coefficients and derived sub-band energy and its variation to represent the tonic characteristic of a short-term sound and to match between two audio segments. Tsai and Hung [27] calculated spectrum energy from sub-band coefficients to simulate the melody contour and used it to measure the similarity between the query example and those database items. Tsai and Wang [28] used scale factors and sub-band coefficients in an MP3 bit stream frame as features to characterize and index the object. Pye [24] designed a new parameterization referred to as an MP3

cepstrum based on a partial decompression of MP3 audio to facilitate the management of a typical digital music library. Jiao et al. [10] took the ratio between the sub-band energy and full-band energy of a segment as intra-segment feature and the difference between continuous intra-segment features as inter-segment feature for robust audio fingerprinting. Zhou and Zhu [32] exploited long-term time variation information based on modulation frequency analysis for audio fingerprinting. Liu and Chang [17] calculated four kinds of compressed-domain features, i.e., MDCT, Mel-frequency cepstral coefficients, MPEG-7, and chroma vectors from the compressed MP3 bit stream to perform MIR.

However, most of the existing compressed-domain music identification algorithms have a common drawback: they do not consider or obtain convincing results to the most central problem of audio fingerprinting, i.e., robustness. In practical application scenarios, for example, recording music to a mobile phone and transmitting it through wireless telecom network, the audio might often be contaminated by various distortions and interferences like lossy compression, noise addition, echo adding, time stretching (or time scale modification, TSM) and pitch shifting. To provide promising identification results for unknown music queries, a music identification system should be insensitive to these distortions.

In this chapter, we propose a novel algorithm of compressed-domain music identification that is robust to common audio distortions and interferences. The algorithm is based on moments, and more specifically, we use Zernike moments to derive invariant audio features for music identification. Of various types of moments that have been investigated for image analysis, Zernike moments have been demonstrated to outperform the others (e.g., geometric moments, Legendre moments and complex moments) in terms of the insensitivity to image noise, information redundancy and capability for faithful image representation [29]. The applications of Zernike moments are widespread, including image recognition [12], human face recognition [8], image representation and matching [5], image watermarking [13] and more recently audio watermarking [30], etc. However, to the authors' knowledge, Zernike moments has not yet been applied to music identification, especially in the compressed domain.

In our algorithm, we first group 90 granules, the basic processing unit in decoding the MP3 bit stream, into a relatively big block for the statistical purpose, then calculate low-order Zernike moments from extracted MDCT coefficients located in the selected low to middle sub-bands, and finally obtain the fingerprint sequence by modeling the relative relationship of Zernike moments between consecutive blocks. Experimental results show that this low-order Zernike moment-based audio feature achieves high robustness against common audio signal degradations like recompression, noise contamination, echo adding, equalization, band-pass filtering, pitch shifting, and slight TSM. A 10-s music fragment, which is possibly distorted, can be identified with an average top-5 hit rate of 90% or beyond in our test dataset composed of 21,185 MP3 popular songs.

The remainder of this chapter is organized as follows. Section 9.2 introduces the basic principles of MP3, bit stream data format, the concept of Zernike moments, and its effectiveness as a robust compressed-domain feature of audio. Section 9.3 details the steps of deriving MDCT low-order Zernike moment-based audio fingerprint and the searching strategy. Experimental results on identification hit rate under various audio

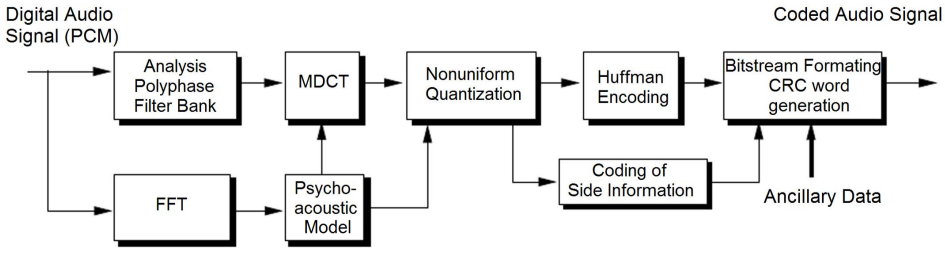

Figure 9.1: Block diagram of MP3 encoding.

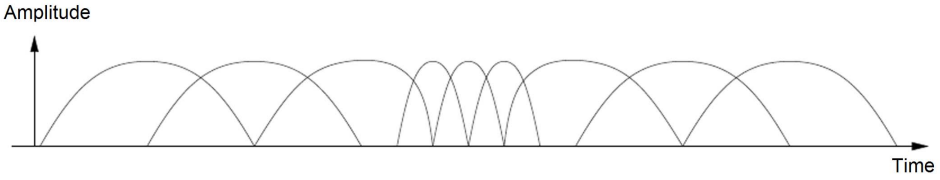

Figure 9.2: A typical sequence of windows applied to a sub-band.

distortions and interferences are given in Section 9.4. Finally, Section 9.5 concludes this chapter and points out some possible ways for future work.

9.2 Compressed-Domain Auditory Zernike Moments

9.2.1 Principles of MP3 Encoding and Decoding

The process of MP3 encoding is shown in Fig.(9.1). First, a sequence of 1,152 PCM audio samples are filtered through a polyphase filter bank into 32 Bark scale-like sub-bands, which simulate the critical bands in the human auditory system (HAS), and then decimated by a factor 32. Each sub-band will thereby contain 36 sub-band samples that are still in the time domain [20, 26]. Next, the sub-bands are further subdivided to provide better spectral resolution by MDCT transform. This starts with a windowing using long or short window depending on the dynamics within each sub-band. If the time-domain samples within a given sub-band show a stationary behavior, a long window (e.g., 25 ms) is chosen in order to enhance the spectral resolution in the following MDCT. If the sub-band samples contain transients, three consecutive short windows (e.g., each is 4 ms) are applied in order to enhance the time resolution in the following MDCT. Moreover, start window and stop window are also defined in order to obtain better adaption when window transients appear. Figure 9.2 shows an example of a sequence of windows applied to a sub-band.

MDCT transform performed on a sub-band will produce 18 frequency lines if a long window is used and 3 groups of 6 frequency lines (each group belongs to different time

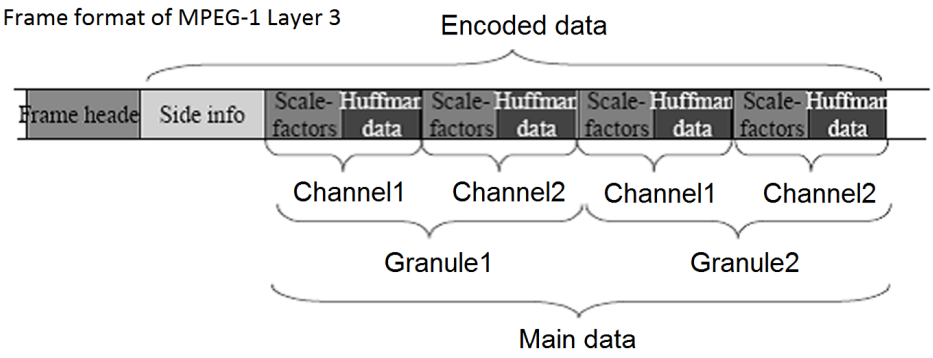

Figure 9.3: Frame format of MP3 bit stream.

intervals) if three consecutive short windows are used. 50% overlap between adjacent windows is adopted in both cases. Therefore, one MDCT transform will produce 576 frequency lines (referred to as a granule) which are organized in different ways in the cases of long windowing and short windowing.

Combined with other adjuvant techniques including psychoacoustic model, Huffman encoding, and quantization etc., the final compressed bit stream is generated. Figure 9.3 displays the frame format of MP3 bit stream [22]. As shown in the figure, each MP3 frame has two granules to exploit further redundancies, and each granule contains 576 samples.

In MP3 decoder, the basic processing unit of the input bit stream is a frame of 1,152 samples, approximately 26.1 ms at the sampling rate of 44.1 kHz (note that each granule can be dealt with independently) [14]. One granule of compressed data is first unpacked and dequantized into 576 MDCT coefficients then mapped to the polyphase filter coefficients in 32 sub-bands by inverse MDCT. Finally, these sub-band polyphase filter coefficients are inversely transformed and synthesized into PCM audio, as shown in Fig.(9.4) [26]. Therefore, access of transformation coefficients in Layer III can be either at the MDCT or the filter bank level, and the latter is obviously more time-consuming.

9.2.2 A Brief Introduction to Zernike Moments

In this subsection, we give a brief introduction to the basic concept of Zernike moments. Zernike moments are constructed by a set of complex polynomials which form a complete orthogonal basis set defined on the unit disk $x^2+y^2 \leq 1$. These polynomials have the form

$$P_{nm}(x,y) = V_{nm}(\rho,\theta) = R_{nm}(\rho)\exp(jm\theta), \tag{9.1}$$

where n is a non-negative integer, m is a non-zero integer subject to the constraints that $(n-|m|)$ is non-negative and even, ρ is the length of vector from the origin to the pixel (x,y), and θ is the angle between the vector and X-axis in counter-clockwise direction, $R_{nm}(\rho)$ is the Zernike radial polynomials in (ρ,θ) polar coordinates defined

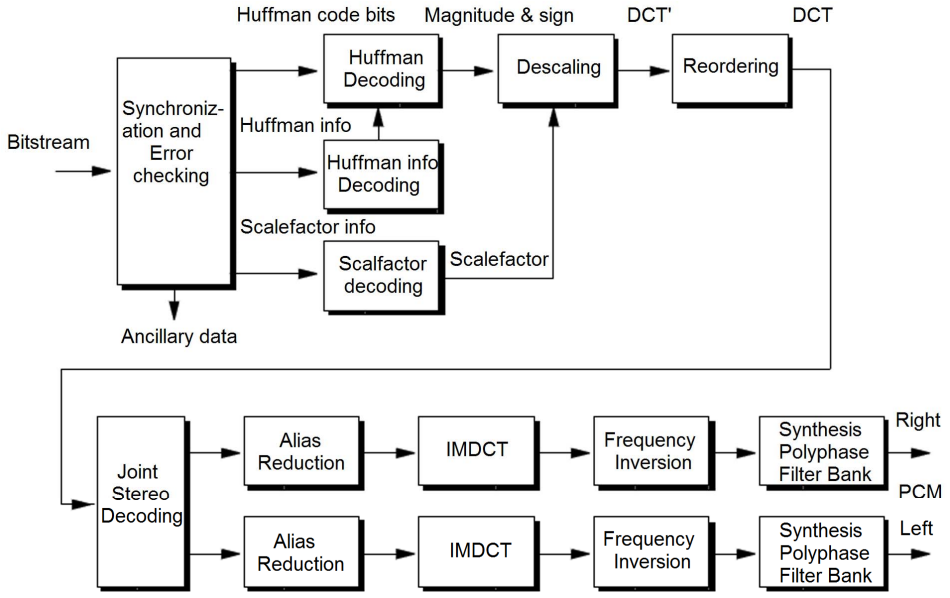

Figure 9.4: Block diagram of MP3 decoding.

as

$$R_{nm}(\rho) = \sum_{s=0}^{n-\frac{|m|}{2}} (-1)^s \frac{(n-s)!}{s!(\frac{n+|m|}{2}-s)!(\frac{n-|m|}{2}-s)!} \rho^{n-2s}. \tag{9.2}$$

Note that $R_{n,m}(\rho) = R_{n,-m}(\rho)$, so $V_{n,-m}(\rho,\theta) = V^*_{n,m}(\rho,\theta)$.

Zernike moments are the projection of a function onto these orthogonal basis functions. The Zernike moments of order n with repetition m for a continuous 2D function $f(x,y)$ that vanishes outside the unit disk is defined as

$$A_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2\leq 1} f(x,y) V^*_{n,m}(x,y) dxdy. \tag{9.3}$$

For 2D signal-like digital image, the integrals are replaced by summations to

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x,y) V^*_{n,m}(x,y), \quad x^2+y^2 \leq 1. \tag{9.4}$$

9.2.3 Compressed-Domain Auditory Zernike Moments

The inconvenience of directly applying Zernike moments on audio lies in that audio is inherently a time-variant 1D data, while Zernike moments are only applicable for 2D data. Therefore, we must map audio signals to 2D form before making them suitable for moment calculation. In our algorithm, we construct a series of consecutive granule-MDCT 2D images to directly calculate the Zernike moments sequence in the

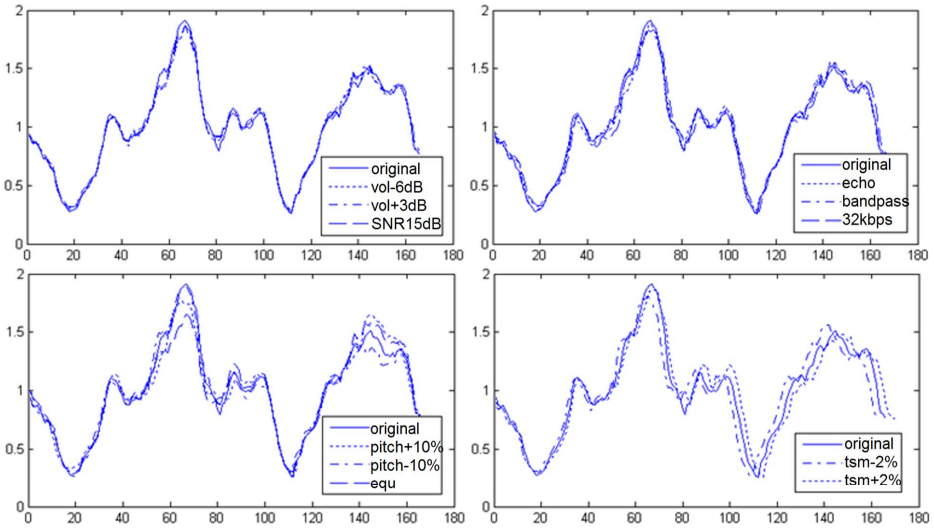

Figure 9.5: An example of MDCT Zernike moments under various audio signal degradations. Order = 2, block = 50 granules, hop size = 2 granules.

MP3 compressed domain. In light of the frame format of MP3 bit stream, one granule corresponds to about 13 ms, which means that it is indeed an alternative representation of time. On the other hand, MDCT coefficients can be roughly mapped to actual frequencies [4]. Therefore, the way we construct granule-MDCT images is virtually done on the time-frequency plane.

Human audition can be viewed in parallel with human vision if the sound is converted from a 1D signal to a 2D pattern distributed over time along a frequency axis, and the 2D pattern (frequency vs. time) constitutes a 2D auditory image [19, 25]. This way, we may seek to explore alternative approaches to audio identification by making recourse to mature technical means of computer vision. Although the link between computer vision and music identification has been made in several published algorithms, which all take short-time Fourier transform of time-domain audio to create spectrogram image [11, 1, 33], methods based on visualization of compressed-domain time-MDCT images have not yet been demonstrated for music identification. We argue that mature techniques in computer vision such as Zernike moments may in fact be useful for computational audition; the detailed calculation procedures of the proposed method will be described in the next section.

As stated in the introduction, the goal of calculating MDCT-based Zernike moment is to use it as an audio fingerprint after necessary modeling, for direct compressed-domain music identification. As an effective audio feature, will it be steady enough under various audio signal distortions? We did some experiments to check it. Figure 9.5 shows an example of MDCT 2-order Zernike moments sequence calculated from a 5-s clip of an MP3 song. The calculation includes several steps like granule grouping, sub-bands selection, and auditory image construction, which will be depicted

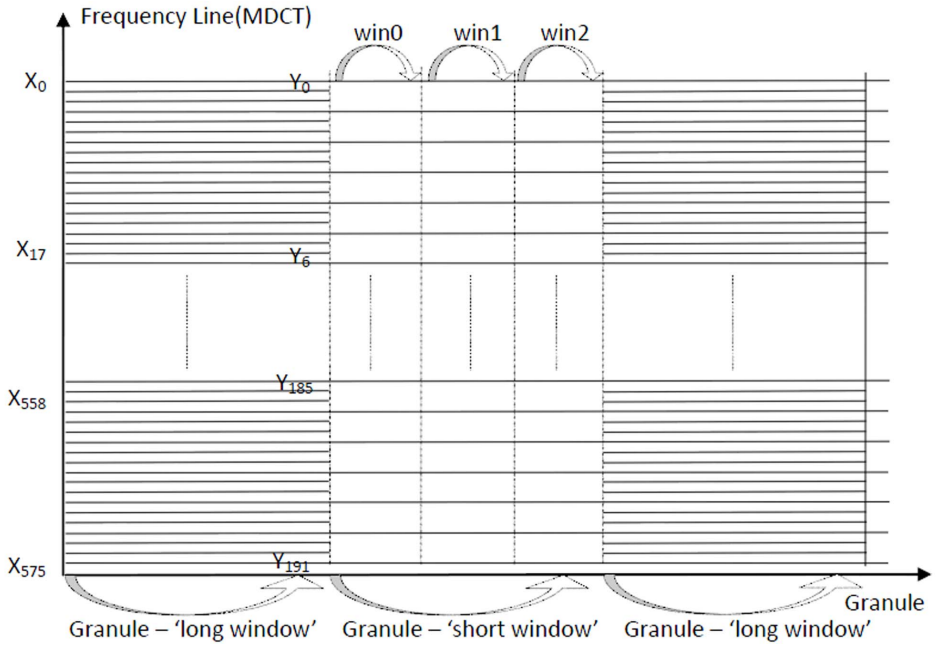

Figure 9.6: Distribution of MDCT coefficients in 'long window' and 'short window' types of granule.

in detail in the next section. It can be clearly seen that the Zernike moment curve is rather stable, keeping its basic shape at the same time positions under common audio signal distortions like MP3 recompression at 32 kbps, echo adding, band-pass filtering, noise contamination, volume modulation, equalization, and pitch shifting up to $\pm 10\%$. When the sample excerpt is slightly time scale-modified, the curve only translates a small distance along the time axis with little change to the basic shape. These observed phenomena confirm our initial motivation. Herein, low-order Zernike moment of time-MDCT auditory image displays great potential to become a powerful audio fingerprint.

9.3 Algorithm Description

As described above, the main difficulty of applying Zernike moments to audio is the dimension mismatching. So, we first depict how to create 2D auditory images from 1D compressed-domain MP3 bit stream. The detailed procedure is described as follows.

9.3.1 MDCT-Granule Auditory Image Construction

The construction of MDCT-granule auditory image is composed of the following steps:

Y-axis construction: MP3 bit stream consists of many frames, which are the basic processing unit in decoding. Each frame is further subdivided into two independent granules, each with 576 values. If a granule is encoded using long window, these 576 values represent 576 frequency lines and are assigned into 32 Bark scale-like sub-bands, that is, each sub-band includes 18 frequency lines. If a granule is compressed via short window, these values stand for 3 groups of 192 frequency lines, and each group corresponds to one of the three consecutive windows respectively, see Fig.(9.6).

In order to construct the Y-axis of auditory images, we must unify the frequency distribution of both long- and short-window cases by adapting the original MDCT-granule relationship to achieve approximately the same frequency resolution. For long-window cases, we group every three consecutive MDCT coefficients of one granule into a new sub-band value, which is equal to the mean of the absolute value of the original three MDCT coefficients, see the upper part of Eq.(9.5). For short-window cases, we substitute the original three MDCT values belonging to different windows at the same frequency line with the mean of their absolute value, see the lower part of Eq.(9.5). In this way, all MDCT values in a granule are uniformly divided into 192 new sub-bands for both long- and short-window cases; this forms the basis for further construction of auditory image.

$$sn(i,j) = \begin{cases} \frac{1}{3}\sum_{n=3i}^{3i+2} |s(n,j)| & \text{for the case of long window} \\ \\ \frac{1}{3}\sum_{m=0}^{2} |s^m(i,j)| & \text{for the case of short window} \end{cases} \quad i = 0, 1, \ldots, 191, \tag{9.5}$$

where $sn(i,j)$ is the new MDCT coefficient at the ith sub-band and jth granule, $s(n,j)$ is the original MDCT coefficient at the nth frequency line and jth granule for the long-window case; $s^m(i,j)$ is the original MDCT coefficient at the ith frequency line, jth granule and mth window for the short-window case.

X-axis construction: After the above Y-direction construction, the next step is to set up the X-axis to form the final auditory images. In our algorithm, N continuous granules ($N = 90$ in experiment) are partitioned into a block and act as the X-axis of one auditory image. Overlap of blocks is taken to improve the robustness of our algorithm against time desynchronization. The hop size between adjacent blocks is M granules ($M = 1$ in experiment).

Auditory image construction: With the above definition of X- and Y-axes, we are now to construct the auditory images for moment calculation. Figure 9.7 is an image for illustration, where its pixels constitute an $192 \times N$ matrix. 192 pixels along the Y-axis represent 192 new MDCT coefficients calculated in terms of Eq.(9.5), and N pixels at the X-axis mean the N time-domain granules, i.e., a block. It is known that sounds located in the low-middle frequency area cover the main content most vital to the HAS and are usually much more robust against various audio distortions than high frequency components. Therefore, we pick out the second to the fifty-first new sub-band MDCT values in this method to act as the Y-axis, which roughly correspond to 300 to 5,840 Hz of real frequency according to Table 9.1 [31]. N is set to 90 granules to form the X-axis and mitigate the problem of desynchronization.

Consequently, the (x, y) coordinates of a pixel in the kth constructed auditory image

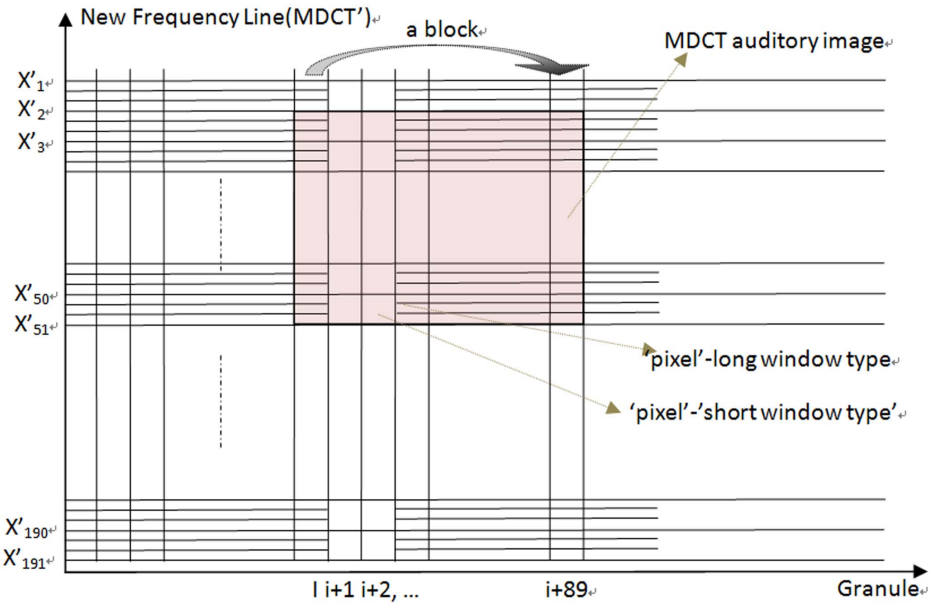

Figure 9.7: An illustration of the constructed auditory image.

are shown in Eq.(9.6)

$$f^k(x,y) = sn(i,j) \quad \begin{array}{l} k = 0, 1, \ldots, N_{\text{block}} - 1 \\ x = i = 2, 3, \ldots, 51 \\ y = 0, 1, \ldots, 89 \\ j = k \times \text{hop size} + y \end{array}, \tag{9.6}$$

where k means the kth auditory image, and N_{block} is the total number of blocks of the query clip or the original music piece, which is variable and determined by the audio length.

9.3.2 Compressed-Domain Audio Features: MDCT Zernike Moments

Fragment input and robustness are known to be two crucial constraints on audio fingerprinting schemes. If modeling with audio operations, this is equal to imposing random cropping plus other types of audio signal processing on the input query example. Random cropping causes serious desynchronization between the input fingerprint sequence and those stored ones, bringing a great threat to the identification hit rate. Usually, there are two effective mechanisms to resist time-domain misalignment, one is invariant feature, and the other is implicit synchronization which might be more powerful than the former [6]. However, in the MPEG compressed domain, due to its compressed bit stream data nature and fixed frame structure, it is almost impossible to extract meaningful salient points serving as anchors as in the uncompressed domain

Table 9.1: Map between MDCT coefficients and actual frequencies for long and short windows sampled at 44.1 kHz

Long window		Short window	
Index of MDCT coefficient	Frequency (Hz)	Index of MDCT coefficient	Frequency (Hz)
$0 \sim 11$	$0 \sim 459$	$0 \sim 3$	$0 \sim 459$
$12 \sim 23$	$460 \sim 918$	$4 \sim 7$	$460 \sim 918$
$24 \sim 35$	$919 \sim 1{,}337$	$8 \sim 11$	$919 \sim 1{,}337$
$36 \sim 89$	$1{,}338 \sim 3{,}404$	$12 \sim 29$	$1{,}338 \sim 3{,}404$
$90 \sim 195$	$3{,}405 \sim 7{,}462$	$30 \sim 65$	$3{,}405 \sim 7{,}462$
$196 \sim 575$	$7{,}463 \sim 22{,}050$	$66 \sim 191$	$7{,}463 \sim 22{,}050$

[15]. Therefore, designing a statistically stable audio feature becomes the main method to fulfill the task of fragment retrieval and resisting time-domain desynchronization in audio fingerprinting.

With the preparations above, we substitute $f(x, y)$ in Eq.(9.4) with $f^k(x, y)$ in Eq.(9.6) and calculate the Zernike moments of the kth auditory image as below

$$A_{nm}^k = \frac{n+1}{\pi} \sum_x \sum_y f^k(x, y) V_{n,m}^*(x, y), \tag{9.7}$$

where n is the moment order, and m must be subject to the condition that $(n - |m|)$ is non-negative and even.

Note that n, the order, plays a crucial role in the moment calculation. A carefully selected order will directly determine the robustness of this feature and the running speed. Generally speaking, low-order moments characterize the basic shape of an audio or image signal, while higher-order ones depict the high-frequency details [21]. Thereby, we naturally conjecture that low-order Zernike moments will perform better than high-order moments in our application. In order to verify this assumption and help obtain the most suitable order number for strong robustness, we did some comparative experiments. As shown in Fig.(9.8), the Zernike moments of orders 2, 6, 10, and 16 are first calculated and then compared for the original audio segment and the audio under two typical distortions, i.e., equalization and noise addition. It can be clearly seen in the figure that with the order increasing, the moment envelope fluctuates more and more dramatically. The Zernike moment curve of the order 2 is the most stable one in the experiment and is chosen as the final value in our algorithm. An affiliated benefit brought by this order is that the computation speed of its corresponding Zernike moment is much faster than any other higher-order situations.

9.3.3 Fingerprint Modeling

On the basis of the Zernike moments calculated from a series of auditory images sliding along the granule axis, we sum up all Zernike moments with order $n \leq 2$ as the final feature to further increase the invariance as shown in Eq.(9.8). The final audio

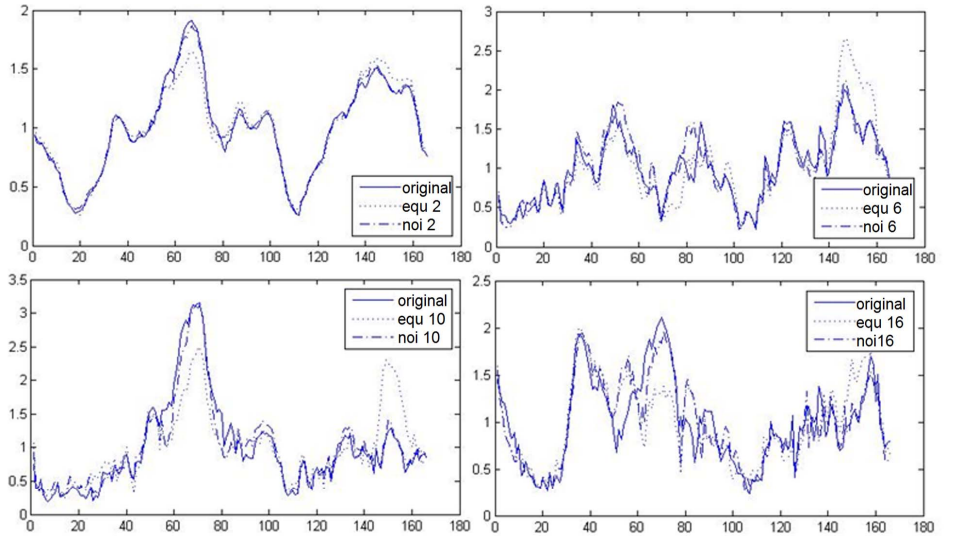

Figure 9.8: Stability of MDCT Zernike moments at the order of 2 (upper left), 6 (upper right), 10 (lower left) and 16 (lower right).

fingerprint sequence is derived according to Eq.(9.9). This method is straightforward yet effective by omitting the exact moment values and only retaining their relative magnitude relationship. Similar methods have been used in query-by-humming systems to model the progressive tendency of the melody line.

$$Z^k = \sum_{\substack{0 \le n \le 2 \\ n - |m| \ge 0 \\ (n-|m|)\%2 = 0}} A_{nm}^k \tag{9.8}$$

$$S(k) = \begin{cases} 0 & \text{if } Z^k < Z^{k+1} \\ 1 & \text{if } Z^k \ge Z^{k+1} \end{cases} \quad k = 0, 1, \ldots, N_{\text{block}} - 1. \tag{9.9}$$

9.3.4 Fingerprint Matching

The emphasis of this chapter is to take compressed-domain audio Zernike moments as the key features for audio fingerprinting. As stated in Section 9.2, such kind of feature is rather stable under common audio signal distortions and slight time-domain misalignment like TSM. By further modeling with the fault-tolerant magnitude relationship between moments of successive auditory images, the steadiness of the derived fingerprints is further reinforced. Therefore, by right of the power of the stable fingerprint, we can adopt a relatively straightforward yet effective measure, i.e., Hamming distance, to perform exhaustive matching between the query example and those stored recordings. An illustration of the matching procedure is shown in

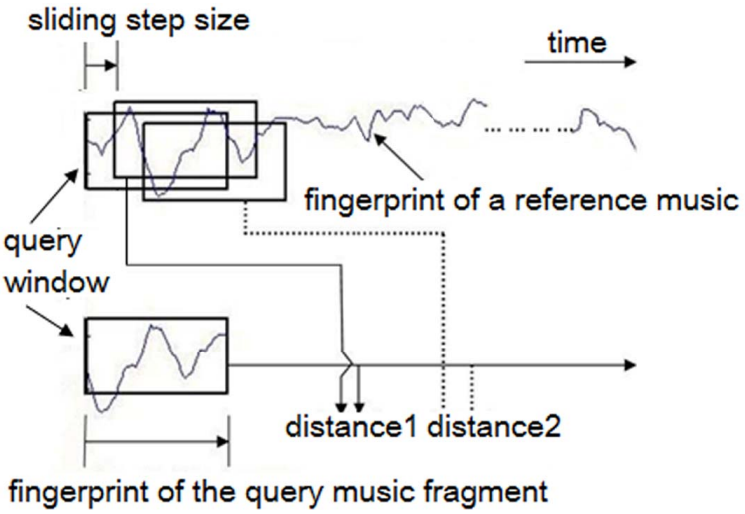

Figure 9.9: An illustration of the fingerprint matching procedure.

Fig.(9.9). More specifically, let $\{x_0, x_1, \ldots, x_{n-1}\}$ be the fingerprint sequence of the query example, $\{y_0^i, y_1^i, \ldots, y_{N-1}^i\}$ the fingerprint sequence of the ith database song ($n \ll N$), N_{song} be the number of songs stored in the database, and Eq.(9.10) be the minimum bit error rate (BER) of matching within a song.

$$\begin{aligned} \text{BER}(i) = \tfrac{1}{n} \min\{(x_0, x_1, \ldots, x_{n-1}) \otimes (y_j^i, y_{j+1}^i, \ldots, y_{j+n-1}^i) \\ \mid j = 0, \ldots, N-n\} \\ i = 0, \ldots, N_{\text{song}} - 1, \end{aligned} \quad . \tag{9.10}$$

The total number of comparison within the database is $(N - n + 1) \times N_{\text{song}}$.

Given a reasonable false positive rate (FPR), the threshold of the BER T can be acquired from both theoretical and practical ways to indicate under what condition a match can be called a hit. Let $\text{BER}(i')$ be the ascending reordered form of $\text{BER}(i)$, namely $\text{BER}(0') < \text{BER}(1') < \text{BER}(2') < \text{BER}(3') < \text{BER}(4') < \ldots < \text{BER}(N_{\text{song}} - 1')$, then the final retrieval results are summarized in Eq.(9.11).

$$result = \begin{cases} \text{top1} & \text{if } k = 0' \\ \text{top5} & \text{elseif } k \in \{1', 2', 3', 4'\} \\ \text{top10} & \text{elseif } k \in \{5', 6', 7', 8', 9'\} \\ \text{failed} & \text{else} \end{cases} \tag{9.11}$$

where k is the ascending rank of BER of the database song where the query example is cut.

9.4 Experiments

The experiments include a training stage and a testing stage. In the training stage, three parameters (i.e., hop size, block size, and BER threshold) that affect the algorithm's performance are experimentally tuned to get the best identification results. To achieve this end, a small training music database composed of 100 distinct MP3 songs is set up. In the testing stage, the algorithm with the obtained parameters from training is tested on a large dataset composed of 21,185 different MP3 songs to thoroughly evaluate the identification performance and robustness. All songs in the two databases are mono, 30 s long, originally sampled at 44.1 kHz, and compressed to 64 kbps, with a fingerprint sequence of 672 bits. In both stages, audio queries are prepared as follows. For each song in the training (testing) database, a 10-s query segment is randomly cut and distorted by 13 kinds of common audio signal manipulations to model the real-world environment, and hence, 1,400 (296,590) query segments (including the original segments) are obtained, respectively.

9.4.1 Parameter Tuning

First, we describe the parameter tuning procedure. Note that when the combination of parameters varies, the associated fingerprint database is named using the following rule, i.e., FPDB_<hop-size>_<block-size>_<order-number>.

Effect of hop size: Hop size is the interval between two adjacent blocks in the time axis. Smaller hop size is beneficial to alleviate the desynchronization between the query segment and its true counterpart in the original audio. Since each block is concatenated by granules, theoretically, one granule of hop size will lead to the minimal displacement. This conclusion is also experimentally demonstrated in Fig.(9.10), where the hop size varies from 1 to 4, the block size is fixed at 30 or 40, and the Zernike moment order is fixed at 2 or 4. It can be clearly seen that when the hop size is 1, the corresponding blue curves are always above other curves. More precisely, when the hop size becomes bigger, the top-1 hit rate curve moves downwards, namely the identification accuracy becomes worse.

Effect of block size: As stated in Section 9.3, a block is assembled by a set of granules in order to endure small variations in the time domain. Generally, longer block will generate steadier Zernike moment value at the cost of lowering local sensitivity and discriminability of fingerprints. To investigate the effect of block size on top-1 hit rate, we first fix the hop size at 1 and Zernike moment order at 2 and then vary the block size from 30 to 120 by increment of 10. From Fig.(9.11), it can be seen that for the common audio signal distortions such as lossy compression, echo adding, and resampling, the top-1 hit rates are not obviously affected by the increase of the block size. However, for TSM ($\pm$2% and $\pm$3% in the experiment), the corresponding four curves (in the middle of the figure) go up monotonically with the increase of the block size and reach a stable status when block size is equal to 90 granules. Therefore, the parameter block size is set as 90 in the experiment.

BER thresholding: Since we use BER as the metric to test fingerprint similarity (discrimination) and robustness, we have to first determine a reasonable threshold T based on the desired FPR in real applications. It is insignificant to claim the robustness

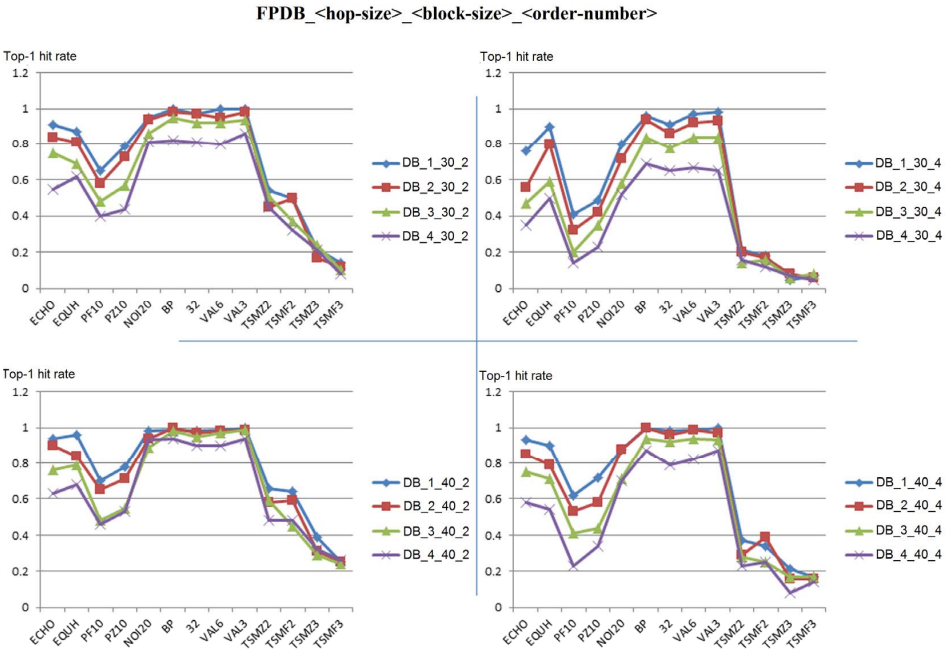

Figure 9.10: Influence of various hop sizes on top-1 hit rate.

without taking FPR into consideration. For a query fingerprint and an equal-length part of a stored fingerprint, they are judged as similar in a perceptual sense if the BER is below the threshold T. In this section, we adopt experimental method to estimate the FPR. First, a set of fingerprint pairs combined from different songs are constructed, then the BER of each pair is calculated. From the result we find out that all BER values exhibit a bell-shaped distribution around 0.5 (this result is similar to that of [9]). Given a specific threshold T, FPR is determined by dividing the number of falsely matched queries by that of all fingerprint pairs. We further observe that experimental FPRs corresponding to most thresholds, for example from 0.2 to 0.4, are acceptable in practice. Then, which threshold is most appropriate? To help make this selection, we did some experiments from another point of view to investigate the relationship between top-1 identification hit rate and the BER threshold T as shown in Fig.(9.12). It can be seen that when T increases from 0.30 to 0.40, the hit rate lines under common audio signal distortions, pitching shifting, and TSM first successively go upwards monotonously and then keep steady after 0.34; in other words, bigger thresholds do not significantly contribute to the identification hit rate any more. In conclusion, 0.34 is adopted as the BER threshold in the experiment.

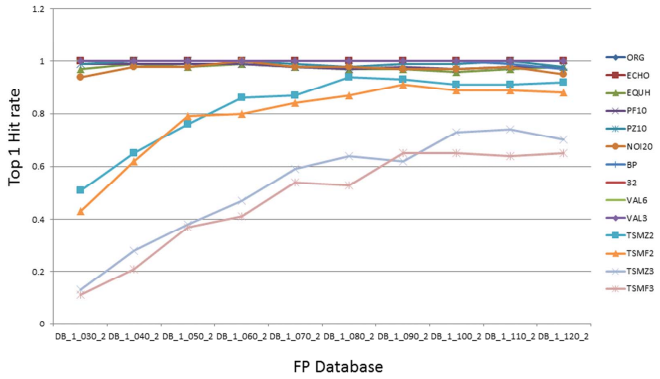

Figure 9.11: Influence of various block sizes on top-1 hit rate.

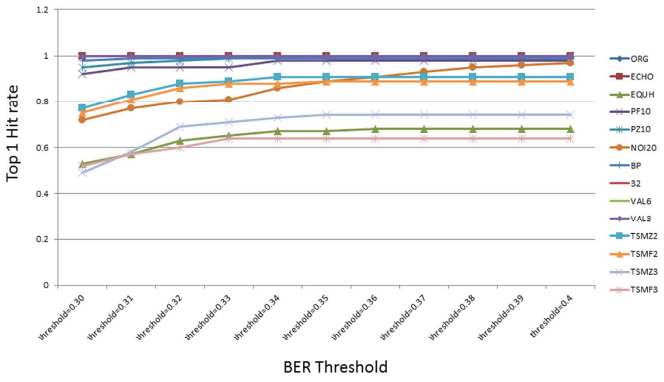

Figure 9.12: Relationship between BER threshold and top-1 hit rate.

9.4.2 Identification Results under Distortions

To simulate the real-world interference, we apply various audio signal operations on the compressed query examples using audio editing tools Cool Edit (Adobe Systems Inc., CA, USA) and Gold Wave (GoldWave Inc., Canada). Since music identification is done in a fragmental way, the processing procedure is actually equivalent to a mixture of random cut plus signal processing. For each song in the testing database, where 21,185 distinct songs are collected all together, a 10-s segment is first randomly cut and then manipulated by 13 various audio signal distortions. Accordingly, the query set amounts to 296,590 audio excerpts. With the parameters set as above (i.e., block size = 90, hop size = 1, and BER threshold = 0.34), the top-1, 5, and 10 identification rates of the queries within the testing dataset are averaged and illustrated in Fig.(9.13). The horizontal axis lists the abbreviation of audio signal distortions adopted in the experiment. ORG means original audio signal which is not distorted. ECHO means echo addition with 100-ms delay and 50% decay. EQUH means 10-band equalization.

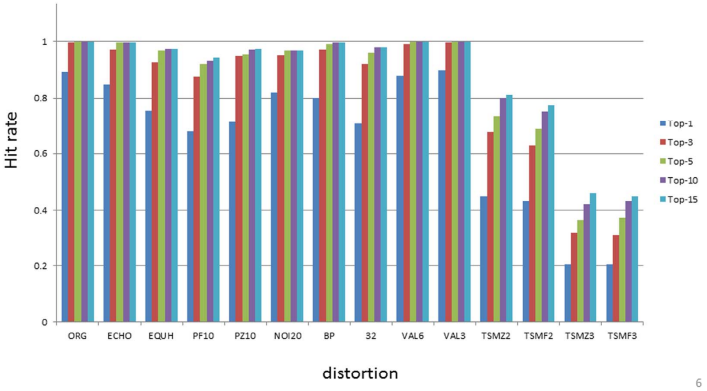

Figure 9.13: Identification performance under various distortions.

PF10 and PZ10 mean pitch shifting by -10% and $+10\%$, respectively. NOI20 means noise addition at signal-to-noise ratio (SNR) of 20 dB. BP means band-pass filtering from 100 to 6,000 Hz. 32 means MP3 recompression under 32 kbps. VAL6 and VAL3 mean volume change under -6.02 and $+3.52$ dB, respectively. TSMZ2, TSMF2, TSMZ3, and TSMF3 mean TSM under $+2\%$, -2%, $+3\%$, and -3%, respectively.

It can be seen that our proposed MDCT Zernike moment-based fingerprint shows satisfying identification results, even under severe audio signal processing like heavy lossy recompression, volume modulation, echo adding, noise interference, and various frequency wrappings such as band-pass filtering, equalization, and pitch shifting ($\pm 10\%$). To be more specific, when the queries are original or only distorted by echo adding, band-pass filtering, and volume modulation, the top-5 hit rates (green bars) are almost not influenced and all come close to 100%. Under other more severe signal manipulations such as equalization, pitch shifting, noise addition, and MP3 compression, the top-5 hit rates are pretty good and still above 90%.

The only deficiency is that under pitch-reserved TSM, which can be modeled as a kind of cropping/pasting to relatively smooth local parts in between music edges [15], the identification results drop quickly with the increase of scaling factors and become unacceptable when $\pm 3\%$ TSM are performed. This weakness is essentially caused by the fixed data structure of the MP3-compressed bit stream. In this case, implicit synchronization methods based on salient local regions cannot be applied. The only way to resist serious time-domain desynchronization is to increase the overlap between consecutive blocks and design more steady fingerprints; however, the overlap has an upper limit of 100% (98% has been used in our algorithm), and discovering more powerful features is not an easy work.

9.4.3 False Analysis

In a practical identification system, two important false statistics must be taken into account to thoroughly evaluate the overall performance. The first is called false negative, which refers to the fail of detecting correct songs even though the query is

Table 9.2: False statistics of identification results

Actual	Predicted	
	Positive	Negative
Positive (27,378)	23,336	4,042
Negative (29,256)	106	29,150

included in the database. The second is false positive, which refers to the return of wrong matched results for a query that does not belong to the database and is more annoying in commercial applications. Below, a confusion matrix is adopted to analyze the two types of errors. To achieve this aim, we prepare 27,378 queries that exist in the testing database and 29,256 queries that come from outside the database. In all the true queries, 4,042 of them are not successfully retrieved from the database (i.e., the false negative rate is 14.7%), while for all the false queries, 106 of them are falsely judged to be within the database and get wrong results (i.e., the false positive rate is 3.6×10^{-3}), as shown in Table 9.2. The false positive rate is acceptable in practical application, while the false negative rate is relatively big. The reasons are twofold, one is that the above numbers are top-1 identification results, the other is that many database songs of a same singer have quite similar musical aspects in rhythm, harmonic progression, instrument arrangement etc., so that the queries are confused.

9.5 Conclusion

In this chapter, a novel music identification algorithm is proposed, which directly works on the MP3-encoded bit stream by constructing the MDCT-granule auditory images and then calculating the auditory Zernike moments. By virtue of the short-time stationary characteristics of such feature and large overlap, 10-s long query excerpts are shown to have achieved promising identification hit rates from the large-scale database containing intact MP3 songs and distorted copies under various audio signal operations including the challenging pitch shifting and TSM. For future work, combining the MDCT Zernike moments with other powerful compressed-domain features using information fusion will be our main approach to improve the identification performance and robustness against large time-domain misalignment and stretching. Cover song identification performed right on the compressed domain is our final aim to be accomplished.

References

[1] S. Baluja and M. Covell. Waveprint: Efficient wavelet-based audio fingerprinting. *Pattern Recognition*, 41(11):3467–3480, 2008.

[2] S.O. Belkasim, M. Shridhar, and M. Ahmadi. Pattern recognition with moment invariants: A comparative study and new results. *Pattern Recognition*, 24(12):1117–1138, 1991.

[3] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 41(3):271–284, 2005.

[4] T.Y. Chang. Research and implementation of MP3 encoding algorithm. Master's thesis, Department of Electrical and Control Engineering, National Chiao Tung University (NCTU), Hsinchu, Taiwan, 2002.

[5] Z. Chen and S.K. Sun. A Zernike moment phase-based descriptor for local image representation and matching. *IEEE Transactions on Image Processing*, 19(1):205–219, 2010.

[6] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. *Digital watermarking and steganography*. Morgan Kaufmann, 2007.

[7] J.W. Dennis. *Sound event recognition in unstructured environments using spectrogram image processing*. PhD thesis, Nanyang Technological University, 2014.

[8] J. Haddadnia, M. Ahmadi, and K. Faez. An efficient feature extraction method with pseudo-Zernike moment in RBF neural network-based human face recognition system. *EURASIP Journal on Applied Signal Processing*, 2003(9):890–901, 2003.

[9] J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *International Society for Music Information Retrieval Conference*, pages 107–115, 2002.

[10] Y. Jiao, B. Yang, M. Li, and X. Niu. MDCT-based perceptual hashing for compressed audio content identification. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 381–384, October 2007.

[11] Y. Ke, D. Hoiem, and R. Sukthankar. Computer vision for music identification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, June 2005.

[12] A. Khotanzad and Y.H. Hong. Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.

[13] H.S. Kim and H.K. Lee. Invariant image watermark using Zernike moments. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8):766–775, 2003.

[14] K. Lagerstrom. Design and implementation of an MPEG-1 layer III audio decoder. Master's thesis, Chalmers University of Technology, Department of Computer Engineering Gothenburg, Sweden, 2001.

[15] W. Li, X. Xue, and P. Lu. Localized audio watermarking technique robust against time-scale modification. *IEEE Transactions on Multimedia*, 8(1):60–69, 2006.

[16] W.N. Lie and C.K. Su. Content-based retrieval of MP3 songs based on query by singing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 929–932, 2004.

[17] C.C. Liu and P.F. Chang. An efficient audio fingerprint design for MP3 music. In *International Conference on Advances in Mobile Computing and Multimedia (MoMM)*, pages 190–193, Hue City, Vietnam, December 2011.

[18] C.C. Liu and P.J. Tsai. Content-based retrieval of MP3 music objects. In *ACM International Conference on Information and Knowledge Management (CIKM)*, pages 506–511, USA, November 2001.

[19] R.F. Lyon. Machine hearing: An emerging field. *IEEE Signal Processing Maga-*

zine, 27(5):131–139, 2010.

[20] T. Painter and A. Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515, 2000.

[21] G.A. Papakostas, Y.S. Boutalis, D.A. Karras, and B.G. Mertzios. A new class of Zernike moments for computer vision applications. *Information Sciences*, 177(13):2802–2819, 2007.

[22] S. Pfeiffer and T. Vincent. Formalisation of MPEG-1 compressed domain audio features. Technical report, CSIRO Mathematical and Information Sciences, 2001.

[23] R.J. Prokop and A.P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, 54(5):438–460, 1992.

[24] D. Pye. Content-based methods for the management of digital music. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 2437–2440, 2000.

[25] R. Rifkin, J. Bouvrie, K. Schutte, S. Chikkerur, M. Kouh, T. Ezzat, and T. Poggio. Phonetic classification using hierarchical, feed-forward, spectro-temporal patch-based architectures. Technical report, Computer Science and Artificial Intelligence Laboratory, MIT, 2007.

[26] K. Salomonsen, S. Søgaard, and E.P. Larsen. Design and implementation of an MPEG/Audio layer III bitstream processor. Master's thesis, Aalborg University, Institut of Electronic Systems, 1997.

[27] T.H. Tasi and J.H. Hung. Content-based retrieval of MP3 songs for one singer using quantization tree indexing and melody-line tracking method. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 14–19, 2006.

[28] T.H. Tasi and Y.T. Wang. Content-based retrieval of audio example on MP3 compression domain. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 123–126, 2004.

[29] C.H. Teh and R.T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.

[30] X.Y. Wang, T.X. Ma, and P.P. Niu. A pseudo-Zernike moment based audio watermarking scheme robust against desynchronization attacks. *Computers & Electrical Engineering*, 37(4):425–443, 2011.

[31] Y. Wang and M. Vilermo. A compressed domain beat detector using MP3 audio bitstreams. In *ACM International Conference on Multimedia (MULTIMEDIA)*, pages 194–202, Canada, 2001.

[32] R. Zhou and Y. Zhu. A robust audio fingerprinting algorithm in MP3 compressed domain. *WASET Journal*, 5(2011-07-20):608–612, 2011.

[33] B. Zhu, W. Li, Z. Wang, and X. Xue. A novel audio fingerprinting method robust to time scale modification and pitch shifting. In *ACM International Conference on Multimedia (MM)*, pages 987–990, Italy, 2010.