Zhang *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2015) 2015:6
DOI 10.1186/s13636-015-0050-0

EURASIP Journal on
Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

SIFT-based local spectrogram image descriptor: a novel feature for robust music identification

Xiu Zhang[1], Bilei Zhu[2], Linwei Li[1], Wei Li[1,3*], Xiaoqiang Li[4], Wei Wang[5], Peizhong Lu[1] and Wenqiang Zhang[1]

Abstract

Music identification via audio fingerprinting has been an active research field in recent years. In the real-world environment, music queries are often deformed by various interferences which typically include signal distortions and time-frequency misalignments caused by time stretching, pitch shifting, etc. Therefore, robustness plays a crucial role in music identification technique. In this paper, we propose to use scale invariant feature transform (SIFT) local descriptors computed from a spectrogram image as sub-fingerprints for music identification. Experiments show that these sub-fingerprints exhibit strong robustness against serious time stretching and pitch shifting simultaneously. In addition, a locality sensitive hashing (LSH)-based nearest sub-fingerprint retrieval method and a matching determination mechanism are applied for robust sub-fingerprint matching, which makes the identification efficient and precise. Finally, as an auxiliary function, we demonstrate that by comparing the time-frequency locations of corresponding SIFT keypoints, the factor of time stretching and pitch shifting that music queries might have experienced can be accurately estimated.

1 Introduction

With the proliferation of a huge amount of digital music, online listening, downloading, and searching have become very popular applications among end users of the Internet in the past decade. Among the applications, music identification that is capable of recognizing unknown music segments has attracted much attention from both the research community and the industry. Music identification technique relies on an audio fingerprint which is defined as a unique and compact digest characterizing and summarizing the perceptually relevant audio content. Different methods have been proposed to construct a valid fingerprint, exploiting the properties of music characteristics as in [1-3] or applying computer vision techniques on the spectrogram of music signals as in [4,5]. Audio fingerprint is used in the music identification typically following the framework described in [6]. First, the algorithm calculates fingerprints of the original music signals and stores them together with affiliated metadata into a fingerprint database; then, when presented with an unlabeled and probably distorted music segment, it extracts a fingerprint from the query audio and compares it with those in the database. If a match is found for the query fingerprint, the unlabeled music segment is identified and the associated metadata such as information concerning singers, album, lyrics, and the like is returned.

The fingerprint for music identification should be highly discriminative over a large number of distinct fingerprints, compact for ease of storing and comparing, scalable to a large database of music records or a large number of concurrent identifications, and robust against a range of environmental distortions and transmission interferences. Among the above properties, robustness plays a central role. In the real world, a person might be interested to know the lyrics or singer of a song played in a noisy environment, then she/he records a short piece of music using a mobile phone and sends it to a remote server for identification through fingerprint matching. In this circumstance, to achieve a successful matching, the extracted fingerprint must be robust against serious distortions caused by, for example, poor

*Correspondence: weili-fudan@fudan.edu.cn
[1] School of Computer Science, Fudan University, Shanghai 201203, China
[3] Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 200433, China
Full list of author information is available at the end of the article

speakers, cheap microphones, background noise, echo, and wireless telecommunication. Moreover, many musical recordings played by TVs or radio broadcasts are often played at arbitrarily different speeds with the pitch changed or unchanged to comply with strict program schedule constraints, which constitutes the most challenging problem in the context of music identification. Specifically, such time-frequency distortions can be modeled by time scaling (or linear speed change) which modifies both duration (or speed) and pitch of music signals by resampling, time stretching (or time scale modification (TSM)) which changes only the duration using certain algorithms, and pitch shifting which merely causes the change of pitch. Roughly speaking, time scaling can be approximately deemed as the combination of time stretching and pitch shifting. Compared with other signal degradations which only influence the perceptual quality, time stretching/scaling and pitch shifting usually lead to a more significant drop of identification performance since they bring about desynchronization problems in the time and/or frequency domains.

In this paper, we extend our previous work of [7] and propose a novel music identification algorithm that is highly robust to not only common audio signal distortions but also serious time- and frequency-domain synchronization warping simultaneously. The basic idea of our algorithm follows the line of applying computer vision techniques for music identification as did in [4] and [5]. Specifically, we first convert a music signal into a two-dimensional spectrogram image, then a powerful local descriptor, i.e., scale invariant feature transform (SIFT), is computed from the image to construct a sub-fingerprint. Thanks to the stability of the SIFT feature, the proposed algorithm exhibits a high discrimination and strong robustness. To our knowledge, this is the first algorithm that can simultaneously resist the abovementioned three challenging distortions, namely time scaling, time stretching, and pitch shifting. Moreover, introducing the SIFT feature into the spectrogram image brings an auxiliary contribution to this paper, i.e., a novel method of estimating the factor-of-time stretching and pitch shifting, which provides further information on how the query music has been wrapped in time and frequency. To make the identification efficient and precise, a locality sensitive hashing (LSH)-based nearest sub-fingerprint retrieval method and a matching determination mechanism are also integrated into this algorithm.

The remainder of the paper is organized as follows. Section 2 summarizes related works. Section 3 describes the processes of spectrogram image construction and robust audio fingerprint extraction. Section 4 details the LSH-based nearest sub-fingerprint retrieval method and the matching determination mechanism for robust sub-fingerprint matching. Section 5 introduces the principle of factor estimation of time stretching and pitch shifting. Finally, robustness and identification experiments are shown in Section 6 and the whole paper is concluded in Section 7.

2 Related work

Recently, a variety of audio fingerprinting algorithms have been proposed in the literature, each with a different degree of robustness. Most of them generate audio fingerprints from spectral features and obtain enough robustness against common audio signal deformations such as audio coding and noise addition and equalization. However, only few methods exhibit a certain capability of resisting time stretching/scaling and pitch shifting, as summarized below. Philips robust hash (PRH) [8] is one of the most significant methods and is usually deemed as a milestone. By segmenting audio signals into heavily overlapped (31/32) frames and extracting a 32-bit sub-fingerprint from 33 Bark-scale frequency sub-bands of each frame according to the energy differences between sub-bands, PRH exhibits a certain robustness when audio lengths are stretched from −4% to +4%[a] on a small dataset consisting of only four music excerpts. Unfortunately, the basic idea of this algorithm makes it susceptible to even a small amount (e.g., ±1%) of frequency misalignment caused by speed change, with significantly dropped performance. To overcome the pitch-sensitive problem, the Philips authors commented that two simple methods can be utilized. One is to store the original audio and its pitch-shifted versions into the database, and the other attempts to use multiple pitch-shifted queries for each audio clip to be identified. However, the exhaustive nature makes both methods inefficient. Namely, the first needs more memory space, and the second aggravates the retrieval complexity since it needs to exhaustively search within a set of possible scaling parameters.

To enhance the speed-change robustness of PRH, several extensions of this method have been developed. In [9], the Philips authors modified their original algorithm and achieved ±6% tolerance of speed change by exploiting shift invariance of the auto-correlation function of a densely sampled power spectrum, which logarithmically portions the energy from 300 to 2,000 Hz into 512 instead of the original 33 sub-bands. Seo et al. [10] extracted fingerprints from the phase components of the Fourier-Mellin transform of locally normalized audio spectrum. In a rather small testing dataset with only four different original excerpts, scale invariance of the transform renders the fingerprints robust against pitch shifting caused by speed changes up to ±10%, and local normalization ensures the robustness against other common audio manipulations. Bellettini and Mazzini [11] replaced the original 33 Bark-scale sub-bands of PRH with the sub-band division in terms of 12-tone equal temperament (12-TET). Under the constraints of musical scale, the authors assume that generally pitch shifting will only occur on integer-multiples of semitone, and their algorithm achieves as high as +41.42% (+6 semitones) resistance against frequency misalignment by shifting the fingerprint bits. As indicated above, the major drawback of this method is that it cannot handle random pitch shifting, which lowers its value.

Motivated by the human auditory algorithm, Sukittanon et al. [12] proposed to use long-term modulation scale features for audio content identification. Combined with channel compensation and sub-band normalization techniques, this method achieves certain insensitivity to distortions such as low-bit-rate MP3 and WMA, frequency equalization, dynamic range normalization, and TSM (±5%). Whereas, experiments on pitch shifting are not reported. Seo et al. [13] first divided the audio spectrum (300 $\sim$ 5,300 Hz) into 16 critical bands, then calculated a normalized frequency centroid for each critical band and used the 16 frequency centroids as the fingerprint of an audio frame. This fingerprint is able to resist moderate time stretching (±4%) and slight linear speed change (±1%). Malekesmaeili and Ward extracted audio fingerprints from adaptively scaled patches of the time-chroma representation, i.e., chromagram of the input audio signal [14]. The proposed fingerprint shows high robustness against tempo change and pitch shifting.

In [15], Wang described an audio fingerprinting algorithm whose ideas have been used in the famous Shazam music matching service[b]. This algorithm first identifies spectrogram peaks which are considered stable under noise and distortion. It then forms these peaks into pairs and uses the parameters of these pairs (frequencies of the peaks and the time interval between them) to generate fingerprints. Experiments show that the Wang algorithm is robust to noise addition and GSM compression, but its basic principle makes it sensitive to time and frequency synchronization distortions. In [16], Fenet et al. extended the Wang algorithm by using constant Q transform (CQT) and a new peak pair encoding mechanism. These modifications make the algorithm more robust to pitch shifting. In [17], Dupraz and Richard proposed a similar algorithm and used an ensemble of time-localized frequency peaks as the fingerprint for audio identification. By determining a constant pitch-shifting factor and multiplying all peak frequencies of the query signal by this factor prior to fingerprint matching, this method allows for promising audio identification performance with a +5% speed change.

AudioPrint [18], proposed by IRCAM, is a music recognition algorithm based on short-term and long-term frames (double-nested) short-time Fourier transform (STFT). Ramona and Peeters performed two-round improvements on this alogrithm in 2011 and 2013. In the first round, they improve the algorithm by introducing perceptual scales for amplitude and frequency (Bark bands) and then synchronizing the stream and database frames using an onset detection algorithm [19]. In the second round, cosine filters are introduced in the short-term spectral analysis to compensate the effect of pitch shifting. A simple solution is proposed to determine the frame positions, robust to audio degradations, with nearly no additional cost [20].

As opposed to the above audio identification algorithms based on fixed-length framing plus heavy overlap, which are usually more or less susceptible to time variations, Bardeli and Kurth proposed in [21] to divide audio signals into unequal-length disjoint time intervals. The basic idea is to acquire invariance against cropping and time scaling by picking out prominent local maxima of spectral features as segmentation boundaries. Experiments demonstrate that this algorithm allows identification of audio signals time-scaled up to ±15%, which notably outperforms most fixed-length framed methods.

Spectral features characterizing local spectral or harmonic behavior of a signal serve as the basis of most existing audio fingerprinting methods. However, several other types of interesting audio features have also been investigated. For example, Kurth et al. proposed in [3] a set of time-related features that capture local tempo, rhythm, and meter characteristics of music signals. By quantizing estimated tempos into certain modular tempo classes similar to the well-known pitch chroma classes, a so-called cyclic beat spectrum (CBS) invariant with respect to tempo doubling is obtained, which endows the designed algorithm with high-identification rates even under time scaling from −21% to +26%.

In [22], Lyon proposed a machine-hearing algorithm structure, which first converts the one-dimensional sound into a two-dimensional auditory image and then extracts features from the image to work with a following trainable classifier or decision module. By using this structure, a machine-hearing problem can be transformed into a machine vision problem, and the ideas and techniques from the vision field (e.g., sparse representation, compression, multi-scale analysis, and keypoint detection) can be used to solve the machine-hearing problem. As an illustration, Lyon et al. showed in [23] that sparse-coded auditory image features degrade less in interferences than vector-quantized Mel-frequency cepstral coefficients (MFCCs). In the literature, there have been several attempts that apply computer vision techniques for music identification. In [4], Ke et al. designed an algorithm that automatically learns local descriptors from the spectrogram via pairwise boosting. In contrast, Baluja and Covell [5] first divided the spectrogram into smaller spectral images and then decomposed these images using Haar wavelet. Audio fingerprints are finally obtained by binary quantization of retained significant wavelet components. Unfortunately, the algorithm in [4] is by nature very weak to time-varying distortions and there are no related experimental results reported, and the algorithm in [5] shows only certain robustness against ±10% TSM and slight resistance under ± 2% speed change.

3 Robust audio fingerprint extraction

Music signals are often contaminated by various distortions in the real-world environment. Therefore, creating highly robust feature representation is a prerequisite and challenging task for music identification. In this paper, we propose to use a SIFT local feature originating from the computer vision field for music identification. Although the link between music identification and computer vision has been made in several published algorithms such as [4] and [5], we argue that a SIFT descriptor calculated in a spectrogram image is indeed a novel and rather robust feature. The details of calculating a SIFT-based audio fingerprint are described as follows.

3.1 Spectrogram image construction

The first step of our algorithm is to construct a spectrogram image from the input music signal. This is accomplished as follows.

1) Perform STFT on the music signal to obtain the linear spectrogram, using Hanning-windowed frames of 185.76 ms (8,192 points) with a three-fourth overlap. The frame length and overlap are selected based on the following considerations. First, a long frame length endues the spectrogram with a low time resolution, which makes the representation insensitive to time variations. Second, under the framework of fixed framing, heavy overlap is a prerequisite to deal with the lack of synchronization between the short query music and the long original signal [24], since excerpts only a few seconds long are used to identify the whole audio signals. Classical PRH algorithm [8] uses an overlap up to 31/32; herein, we experimentally adopt three fourths to balance the desynchronization resistance and searching speed.

2) Quantize the linear spectrogram obtained above into 64 logarithmically spaced frequency sub-bands in terms of Equation 1 so that frequency multiplication can be reduced to addition:

$$f_i = f_{\min} \times 2^{\frac{i-1}{12}}, \tag{1}$$

where f_i is the central frequency of the i^{th} sub-band, $i = 1, \ldots, 64$ is the sub-band index, and $f_{\min} = 318$ Hz is the minimum frequency. Therefore, the spectrogram adopted ranges from 318 to 12,101 Hz, which covers the five medium-to-high perceptually important octaves and is large enough to extract more local image features described in the next section for robust matching.

3) Convert the logarithmic spectrogram into a gray image where image features can be extracted. To achieve this end, the spectrogram is first transformed into a log-magnitude representation as follows:

$$S(i,j) = \log|X(i,j)|, \tag{2}$$

where X is the spectrogram and i and j are the frequency sub-band index and the frame index, respectively. Compared with the linear-magnitude version, the log-magnitude spectrogram reveals more about small-magnitude components where robust local features can also be extracted. After obtaining S, the spectrogram image I is then generated as:

$$I(i,j) = \frac{S(i,j) - \min(S)}{\max(S) - \min(S)} \times 255, \tag{3}$$

where $\min(S)$ and $\max(S)$ are the minimum and maximum values of S, respectively.

3.2 Relationships between audio manipulations and spectrogram image transformations

As mentioned in the introduction, time stretching, pitch shifting, and time scaling are the three most arduous audio distortions for music identification algorithms to resist. Since time scaling can be roughly deemed as the combination of time stretching and pitch shifting, in this subsection, we only take time stretching and pitch shifting into consideration and reveal that they can be distinctly described as corresponding spectrogram image transformations. Remember that time stretching merely changes the speed of an audio signal without affecting its pitch. Therefore, when an audio signal is time-stretched, its spectrogram image remains stable in the frequency axis with only the time axis lengthened or shortened, see sub-figures (a), (b1), and (c1) in Figure 1 for example. By contrast, pitch shifting just modifies the pitch of an audio signal with no influence on its duration. When an audio signal is pitch-shifted, its spectrogram image remains unchanged in the time axis with only frequency components translated upwards or downwards; see sub-figures (a), (d1), and (e1) in Figure 1 for instance.

To make things clearer, below we give some formalized explanations on the relations between pitch shifting and spectrogram image translation. Given a signal component with frequency f, its energy distributes around the sub-band with index $Y(f)$, which is calculated by inverting Equation 1 as below:

$$Y(f) = \text{Round}\left(12 \times \log_2 \frac{f}{f_{\min}} + 1\right), \tag{4}$$

where Round(x) rounds x to the nearest integer. If the signal component is pitch-shifted by a factor k, which is negative when the pitch decreases and positive when the pitch increases, it will move to a new frequency $(1+k)f$, with its

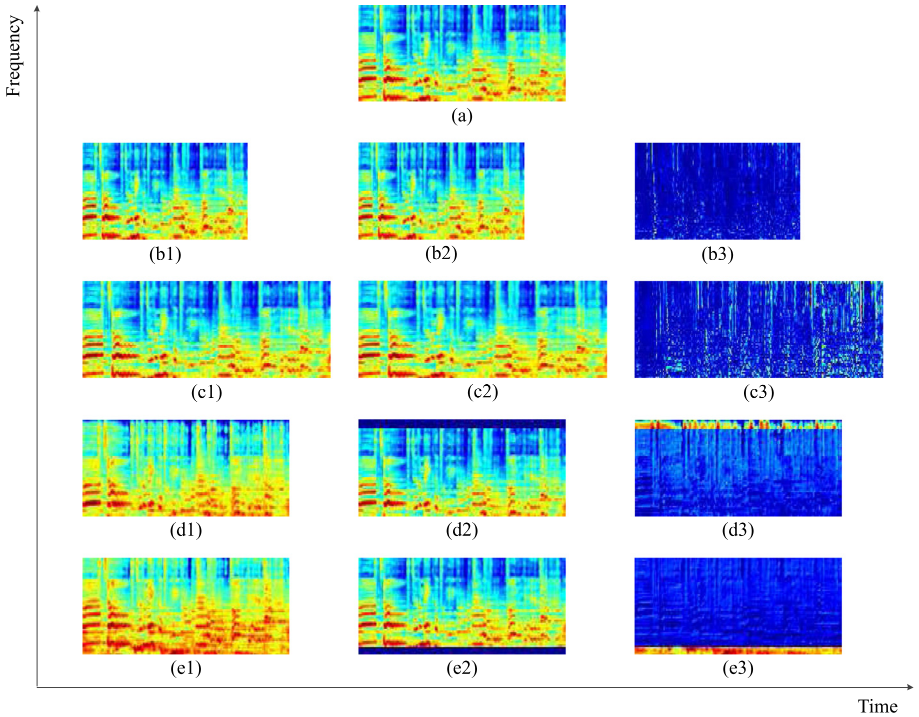

Figure 1 Relationships between audio manipulations and corresponding spectrogram image transformations. (a) is the spectrogram image of an original 10-s music clip. From the second row, the leftmost column displays spectrogram images of four audio excerpts distorted from the original clip: **(b1)** −20% time stretching, **(c1)** +20% time stretching, **(d1)** −30% pitch shifting, and **(e1)** +30% pitch shifting. The middle column displays corresponding images after spectrogram image **(a)** is modified with image transformations: **(b2)** 20% time-axis shortening, **(c2)** 20% time-axis lengthening, **(d2)** six frequency bins downshifting, and **(e2)** five frequency bins upshifting. The rightmost column **(b3, c3, d3, e3)** accordingly illustrates the differences between corresponding sub-figures of the leftmost and the middle columns. Note that warmer colors represent larger spectral differences while cooler colors represent smaller ones.

energy moved to the vicinity of the $Y((1+k)f)^{\text{th}}$ sub-band. Note that the frequency-axis hopping is independent of the absolute frequency f, as shown in Equation 5:

$$Y((1+k)f) - Y(f) \approx \text{Round}(12 \times \log_2(1+k)), \quad (5)$$

which means that pitch shifting applied to an audio signal can be approximately modeled as a constant vertical translation of its spectrogram determined only by coefficient k.

Figure 1 verifies the above deduction, where we can see that spectrogram images calculated from differently time-stretched or pitch-shifted audio signals exhibit high similarity with correspondingly transformed spectrogram images. For example, sub-figures (b3) and (c3) in Figure 1 are chiefly composed of cool-color components, meaning that (b1) and (c1), spectrogram images calculated from −20% and +20% time-stretched audio, possess pretty low difference with (b2) and (c2), −20% and +20% time-axis-stretched images of the original spectrogram. For another example, sub-figures (d3) and (e3) in Figure 1 are mostly composed of cool-color components, except that there are some warmer ones in the upper part of (d3) and the lower part of (e3). As these warmer patches are rather limited, sub-figures (d1) and (e1) in Figure 1, spectrogram images calculated from −30% and +30% pitch-shifted audio, can still be correctly matched to (d2) and (e2), images translated by −6 and +5 frequency-axis bins from the original spectrogram in terms of Equation 5. To conclude, since time stretching and pitch shifting of an audio signal can be modeled by the stretch and translation of its spectrogram image, we argue that image features robust to stretch and translation should also be able to resist time stretching and pitch shifting of the original audio signal.

3.3 Robust spectrogram image feature extraction

Inspired by the machine-hearing algorithm structure of [22], the basic idea of our algorithm is to seek robust spectrogram image features for audio fingerprinting. These features should be discriminative, scalable, and, more importantly, robust to various image distortions including stretch and translation.

In order to resist stretch and translation, local spectrogram image features following the line of implicit synchronization should be more effective than global features. During these last years, local image features have received much attention because of their efficiency for several computer vision problems such as image retrieval [25,26] and object recognition [27,28]. Also, these features have found their applications in audio analysis tasks. In [29], Yu and Slotine drew inspiration from the visual classification method of [30] and proposed to extract spectrogram block matching-based features for instrument classification. In [31], Matsui et al. first extracted SIFT keypoints [27] from the spectrogram and then clustered these keypoints based on their descriptors to form a musical feature for genre classification. In [32], Kaliciak et al. first generated a set of local spectrogram patches by combining a corner detector [33] with a random points generator and then characterized these local patches in the form of a co-occurence matrix or color moments as was done in [34]. These local patch descriptors are finally employed for music genre classification by using the 'bag-of-visual-words' approach.

3.3.1 Scale invariant feature transform (SIFT)

Among the proposed local image features, the SIFT-based features [27] are most invariant to image rotation and robust to changes in scale, illumination, and other image deformations. A typical SIFT feature extractor consists of four major stages briefly summarized as below.

- *Scale-space extrema detection*: The image is first convolved with Gaussian filters at different scales, then the difference of successive Gaussian-blurred images is taken. Potential keypoints are chosen as local maxima/minima of the Difference-of-Gaussians (DoG) that occur at multiple scales.
- *Keypoint localization*: The above detection produces too many keypoint candidates, some of which are unstable. In this step, keypoints that have low contrast are first discarded due to the sensitivity to noise and then those poorly located along edges are filtered out.
- *Orientation assignment*: Each keypoint is assigned one or more orientations based on local image gradient directions. By representing the keypoint descriptor relative to this consistent orientation, invariance to image rotation is achieved.
- *Generation of keypoint descriptor*: A set of orientation histograms are created on 4 × 4 pixel neighborhoods. Histograms contain eight bins each, and accordingly, a 128-dimensional (4 × 4 × 8) descriptor is obtained for each keypoint.

3.3.2 SIFT-based local spectrogram image feature extraction

In the literature, there have been a lot of different robust local image features proposed, among which the SIFT-based features possess the best results compared with other local features in the context of matching and recognition under various image deformations [35]. Naturally, we are inspired to employ SIFT feature extracted from the logarithmic spectrogram image for music identification. Although SIFT feature is originally designed for object recognition in natural images, we claim that its use in the spectrogram image is feasible. According to [29], a typical music piece usually involves lots of different sounds, and its spectrogram contains many partial areas with distinctive local spectral patterns. These patterns in the spectrogram can be regarded as 'objects' in a real image [31].

The output of the SIFT feature extractor is a set of keypoints represented by their location, scale, orientation, and 128-dimensional descriptor (see Figure 2). The SIFT descriptor measures local image gradients and is highly distinctive between different features and robust against a corpus of image transformations. Particularly, comparison tests carried out in [35] have shown that SIFT-based descriptors exhibit the highest matching accuracies for affine transformation such as stretch and translation compared with many other local descriptors. Based on these facts, we believe that the SIFT feature extracted from a spectrogram image is a good choice for music identification, especially considering that its invariance to image stretch and translation will endow the identification algorithm with a strong robustness against time stretching and pitch shifting.

In our method, we take the 128-dimensional SIFT descriptors calculated from the spectrogram image as sub-fingerprints of the underlying music signal. We also reserve the location of each SIFT keypoint for the estimation of time-stretching and pitch-shifting factor (see Section 5). The scale and orientation will not be used and are thus abandoned.

4 Robust matching of audio fingerprints

Following the procedure described in the previous section, we extract sub-fingerprints for each reference music signal and store them in the fingerprint database. When presented with an unlabeled query excerpt, we extract sub-fingerprints from it and independently match each of these sub-fingerprints against the fingerprint

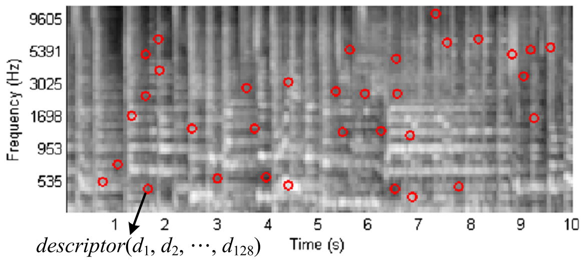

Figure 2 Illustration of SIFT local features extracted from the spectrogram image of a 10-s music excerpt. Each red circle indicates a SIFT keypoint, represented by a 128-dimensional descriptor.

database. The reference music signal which has the most matched sub-fingerprints with the query excerpt is finally returned as the identification result.

In this section, the mechanism of sub-fingerprint matching is described. It consists of two stages, i.e., nearest sub-fingerprint retrieval and matching determination.

4.1 Preliminaries of locality sensitive hashing

LSH [36] is an approximate nearest neighbor search technique that works efficiently even in high-dimensional spaces. Two 'similar' points in the original space can be hashed into a same bucket with high probability, which makes LSH appropriate to perform indexing in the retrieval task. It allows one to quickly find similar elements in large databases and has thus attracted plenty of attention from the research community. In recent years, LSH and its extensions have been successfully applied to a range of applications (e.g., [1,2,25,37,38]) and shown to significantly outperform conventional tree-based schemes such as BBF-Kd-Tree by comparison tests [39].

4.2 Nearest sub-fingerprint retrieval based on LSH

The matching of a query sub-fingerprint is performed by first retrieving its nearest neighbor, i.e., the sub-fingerprint in the fingerprint database that has minimum Euclidean distance to the query sub-fingerprint. However, audio databases in practical applications are usually large, of which corresponding fingerprint databases may contain millions of (or even more) sub-fingerprints. To find the nearest neighbor in such a large database using linear search is, in many cases, unacceptable. Also, owing to the high-dimensional SIFT-based sub-fingerprint vectors, traditional tree-like data structures succumb to the curse of dimensionality and perform no better than an exhaustive linear search.

The LSH-based nearest sub-fingerprint retrieval algorithm contains two phases: indexing and retrieval. In indexing, all the sub-fingerprints in the fingerprint database are inserted into L hash tables corresponding to L randomly selected hash functions $\{g_i, i = 1, \ldots, L\}$. Given a set of sub-fingerprints $\{\mathbf{p}\}$, each of the L hash functions is defined as:

$$g(\mathbf{p}) = (h_1(\mathbf{p}), \ldots, h_k(\mathbf{p})), \tag{6}$$

where k is the width parameter, and $\{h_j, j = 1, \ldots k\}$ are LSH functions satisfying the LSH property, i.e., sub-fingerprints that are close to each other have a higher probability to be hashed into the same bucket than sub-fingerprints that are far apart. Since our SIFT-based sub-fingerprints lie in the Euclidean space, we directly employ the LSH functions proposed in [40] as below:

$$h(\mathbf{p}) = \lfloor \frac{\mathbf{a}^{\mathrm{T}} \cdot \mathbf{p} + b}{r} \rfloor, \tag{7}$$

where $\lfloor x \rfloor$ rounds x to the nearest integer towards negative infinity, $\mathbf{a} \in \mathbb{R}^{128}$ is a random vector with elements chosen independently from a Gaussian distribution, r is a constant which is set to 2.8284 in our implementation following the suggestion of [41], and b is a real number chosen uniformly from the range of $[0, r]$.

In the retrieval phase of the nearest sub-fingerprint search algorithm, given a query sub-fingerprint $\mathbf{q}$, the algorithm iterates over the L hash tables. For each table considered, it compares $\mathbf{q}$ with the sub-fingerprints that are hashed into the same bucket as $\mathbf{q}$. The resulting nearest neighbor is identified as the compared sub-fingerprint which has the smallest Euclidean distance with $\mathbf{q}$ over the L hash tables.

4.3 Matching determination of sub-fingerprint

Using LSH, we first regroup similar elements in the fingerprint database and then, during retrieval, perform a nearest neighbor search for each of the query excerpt's sub-fingerprints within this reorganized database. Conventionally, nearest neighbors found in the database are returned as matched sub-fingerprints. However, since music signals are often distorted in a real-world environment, it is possible that a query sub-fingerprint does not have any correct counterparts in the fingerprint database so that nearest neighbors returned are actually false matches. Also, LSH is substantially an approximate similarity search algorithm; consequently, false positives do exist though very small. Considering these situations, additional measures apart from the basic LSH method must be taken to reduce the rate of false matching.

A natural way is to use a global threshold to the distance between the query sub-fingerprint and its nearest neighbor returned by LSH, rejecting those matches whose distances are larger than the threshold. However, due to the diversity of music signals, determining the threshold is an intractable problem in practical implementation. In this case, we turn to another more effective matching measure which is adopted in [27]. Given a query sub-fingerprint $\mathbf{q}$, we perform a two-nearest neighbor search using LSH and then compare the distance of the closest neighbor $\mathbf{v}$ to that of the second-closest neighbor $\mathbf{v}'$. Specifically, let $D(\cdot, \cdot)$ be the Euclidean distance between two sub-fingerprints and θ be a threshold, if:

$$D(\mathbf{q}, \mathbf{v}) < \theta \times D(\mathbf{q}, \mathbf{v}'), \tag{8}$$

sub-fingerprint $\mathbf{q}$ and $\mathbf{v}$ are judged to be matched.

5 Factor estimation of time stretching and pitch shifting

In some applications such as content-based audio authentication, it might be useful to know whether and how seriously an input music excerpt has been time-stretched or pitch-shifted [42]. In spite of this, to our knowledge, few related works have been reported in the literature. In this section, we design a novel estimation method under the framework of our audio fingerprinting algorithm.

As elaborated in the introduction, time stretching and pitch shifting applied to an audio can be equivalently reflected by time-axis stretch and frequency-axis translation of its logarithmic spectrogram, and it is natural to estimate factors of the two audio distortions by calculating factors of corresponding spectrogram image transformations. Let us take Figure 3 as an example of time-stretching factor estimation. In this figure, A and *A0* are spectrogram images of a query music clip and its reference audio, respectively. a_1 is a stable SIFT keypoint in A and a_2 is the keypoint with minimum time-axis distance to a_1 among all the stable keypoints in A whose time-axis coordinate values are larger than that of a_1. Note that stable keypoints here refer to the SIFT keypoints for which a matched keypoint can be found in the reference audio. This matched keypoint has the smallest Euclidean distance under the constraint of Equation 8 to the stable keypoint. In Figure 3, a_{01} and a_{02} are matched keypoints of a_1 and a_2, respectively.

Given the four keypoints a_1, a_2, a_{01}, and a_{02}, a candidate of time-axis stretch factor k_t between A and $A0$ can be estimated in terms of Equation (9):

$$k_t = \frac{d_t}{d_{t0}} - 1, \tag{9}$$

where d_t is the time-axis distance between a_1 and a_2, and d_{t0} is the time-axis distance between a_{01} and a_{02}. In general, dozens of stable SIFT keypoints can be extracted from spectrogram image A, and consequently, a series of factor candidates can be computed. The median of all these candidates, $\widetilde{k_t}$, is returned as the final estimation result of the time-axis stretch factor of A and also the time-stretching factor of the original query excerpt.

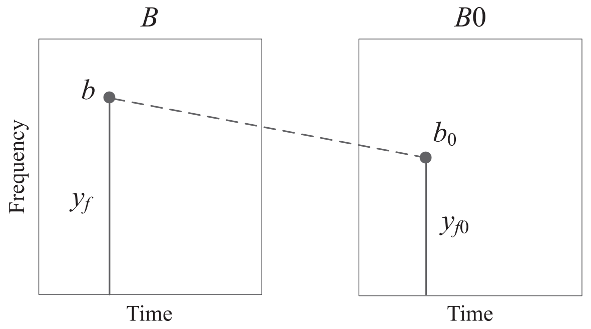

Figure 4 Example of pitch-shifting factor estimation.

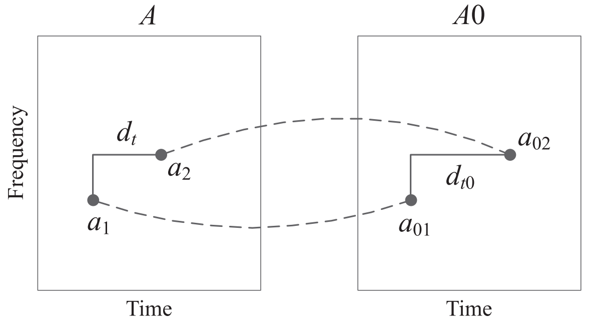

Figure 3 Example of time-stretching factor estimation.

Next, as shown in Figure 4, a candidate of the frequency-axis translation distance between spectrogram images of a query excerpt and its reference audio, B and *B0*, is simply calculated in terms of Equation 10:

$$\Delta y_f = y_f - y_{f0}, \tag{10}$$

where y_f and y_{f0} are the frequency-axis coordinate values of a pair of matched stable SIFT keypoints, b and b_0, respectively. Similarly, there exists a series of translation distance candidates, and the median, $\widetilde{\Delta y_f}$, is selected as the final result.

Remember that Equation 5 depicts the non-linear relation between the pitch-shifting factor ($\widetilde{k_f}$) of an audio signal and the frequency-axis translation distance ($\widetilde{\Delta y_f}$) of its logarithmic spectrogram image. Given $\widetilde{\Delta y_f}$ obtained as above, $\widetilde{k_f}$ can be straightly calculated according to Equation 11:

$$\widetilde{k_f} \approx 2^{\frac{\widetilde{\Delta y_f}}{12}} - 1. \tag{11}$$

6 Experimental results

To thoroughly evaluate the performance of our method, in this section, we first describe the establishment of a music database and affiliated fingerprint database, then experimentally determine several variable parameters, and finally tabulate and show the robustness and identification results. The performance of factor estimation for time stretching and pitch shifting is also presented in this section.

6.1 Database setup

To assess the proposed algorithm, we first collect a total of 10,641 music pieces of various genres such as pop, rock, disco, jazz, country music, classical music, and folk song. Each music signal is mono, 60 s long, and originally sampled at 44.1 kHz. These music pieces are then divided into two audio databases, namely DB_{train} containing 500 music pieces for parameter estimation, and DB_{test} containing 10,141 music pieces for robustness and identification testing. The affiliated fingerprint databases are called $FP\text{-}DB_{\text{train}}$ and $FP\text{-}DB_{\text{test}}$, respectively, where each sub-fingerprint is a 128-dimensional 8-bit integer vector extracted from the logarithmic spectrogram image using the SIFT algorithm implemented in VLFeat [43] with default setting.

Considering the large amount of high-dimensional sub-fingerprints which are found in $\textit{FP-DB}_{\text{train}}$ and $\textit{FP-DB}_{\text{test}}$, we index the two fingerprint databases using the LSH toolbox published by Shakhnarovich [41] (the E2LSH scheme in the toolbox is chosen) for more efficient sub-fingerprint retrieval. The indexed versions of the two fingerprint databases are denoted as $\textit{FP-DB}'_{\text{train}}$ and $\textit{FP-DB}'_{\text{test}}$, respectively.

As typical music identification algorithms usually identify unlabeled and distorted music fragments within the database, we also construct two query sets, i.e., QS_{train} and QS_{test}, for DB_{train} and DB_{test}, respectively. QS_{train} and QS_{test} consist of 10-s short excerpts randomly cut from distinct music pieces in DB_{train} and DB_{test}, respectively. To simulate real-world environments, all the query fragments are subjected to different audio signal distortions and synchronization attacks. The applied audio signal distortions include the following:

- *Lossy compression*: MPEG-1 layer 3 encoding/decoding at 32 kbps;
- *Echo adding*: 50% decay and 500-ms delay;
- *Equalization*: 10-band equalization with the settings of [8];
- *Noise addition*: White Gaussian noise with a signal-to-noise-ratio (SNR) of 18 dB;
- *Resampling*: Subsequent down and up sampling to 22.05 and 44.1 kHz, respectively;
- *Bandpass filtering*: Cutoff frequencies of 100 and 6,000 Hz.

The applied synchronization attacks include time stretching, pitch shifting, and time scaling. Since in the real world these three distortions mostly occur in the range of [−10%, +10%], we deform all the query clips with time stretching/scaling and pitch shifting of ±2%, ±5%, and ±10%. Meanwhile, to obtain the performance limit of the proposed algorithm, time stretching/scaling and pitch shifting out of the above range are also evaluated. To conclude, synchronization distortions we apply on the queries include the following:

- *Time stretching*: ±2%, ±5%, ±10%, ±20%, ±30%, +40%, and +50%;
- *Pitch shifting*: ±2%, ±5%, ±10%, ±20%, ±30%, ±40%, ±50%, +60%, +70%, +80%, +90%, and +100%;
- *Time scaling*: ±2%, ±5%, ±10%, ±20%, ±30%, and +40%.

Note that all the 10-s query music excerpts in the query sets are cut from corresponding original audio pieces starting at arbitrary offsets; accordingly, all distortions performed on the queries are indeed mixed with a precedent random cropping.

6.2 Parameter estimation

There are three parameters to be tuned in the algorithm. In this sub-section, we experimentally investigate their effect on the system performance and make a suitable setting for each of them.

The first parameter to be set is the threshold θ that controls the matching determination principle described in Equation 8. Due to the constraint of θ between the nearest and the second nearest neighbors, not every query sub-fingerprint is ensured to get a matching result, no matter true or false. For a specific sub-fingerprint, bigger θ will bring about more chance to get a result returned. Accordingly, for all query sub-fingerprints, more matching results will be returned with the increase of θ; within the returned results, true matches and false matches generally increase synchronously.

In Figure 5, we increase θ from 0.1 to 1 with a step size of 0.1 and in each step calculate the correct and the false match rates of all sub-fingerprints extracted from the original and differently distorted excerpts of QS_{train} against $\textit{FP-DB}_{\text{train}}$ without using LSH. More specifically, a correct (false) match rate here refers to the percentage of query sub-fingerprints for which we find correct (false) matches in the fingerprint database. A match is considered as correct if the query sub-fingerprint and its matched sub-fingerprint belong to the query excerpt and the reference audio of a same music signal, respectively. As can be seen in the figure, both the correct and the false match rates increase with the increment of θ. When $\theta < 0.8$, the false match rate increases slowly while the

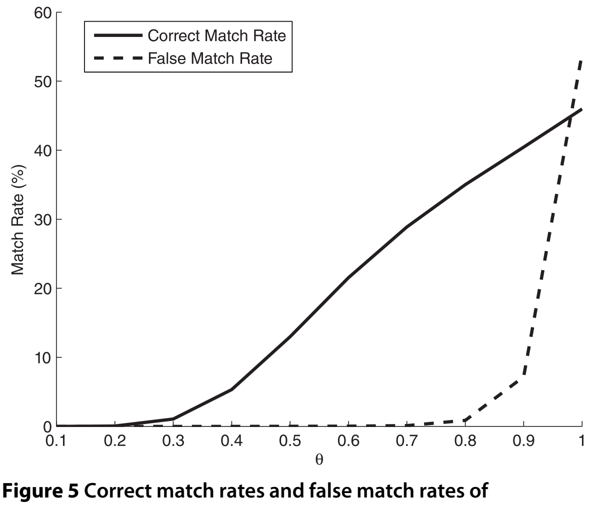

Figure 5 Correct match rates and false match rates of sub-fingerprints for different θs.

correct match rate increases significantly faster; when $\theta > 0.8$, the false match rate becomes more notable and soon exceeds the correct match rate. Based on these observations, we set $\theta = 0.8$ in our experiment. In contrast with the results when $\theta = 1$, this setting eliminates 98.41% of the false matches at the cost of lowering 23.86% of the correct matches.

The next two parameters to be set are related to LSH, i.e., k, the width parameter, and L, the number of hash tables. They directly affect the distribution of sub-fingerprints in the fingerprint database and thus affect the efficiency of nearest sub-fingerprint retrieval. To be specific, a larger k reduces the chance of hitting sub-fingerprints that are not nearest neighbors and thus makes the nearest neighbor retrieval faster. However, this speedup is at the expense of increasing the probability of missing true nearest neighbors. In contrast, a larger L enhances the probability of finding true nearest neighbors, but it increases the time consumption at the same time. Therefore, k and L should be comprehensively considered to balance the trade-off between the retrieval accuracy and speed. In addition, increasing k and L will both lead to more memory usage. In the following, we set $k = 3$ and $L = 10$, and experiments show that this combination prevails over other values on our machine[c].

As an approximate similarity retrieval technique, LSH is aimed to accelerate the retrieval speed at the cost of slight accuracy decrease. Figure 6 compares the performance of LSH with a linear search, where hit rate indicates the percentage of query excerpts in QS_{train} and all their distorted versions which are correctly identified within DB_{train}. It is clear that the matching time for a single sub-fingerprint using LSH is significantly reduced, about 25 times faster than a linear search in our experiment environment, with only 4.7% hit rate decreases.

6.3 Robustness tests

Several groups of experiments are performed in this subsection to evaluate the robustness of the proposed music identification algorithm, using audio database DB_{test}, query set QS_{test}, and the corresponding indexed fingerprint database $FP\text{-}DB'_{\text{test}}$. The performance of each experiment is measured using the hit rate, which refers to the percentage of queries that are correctly identified within the reference database.

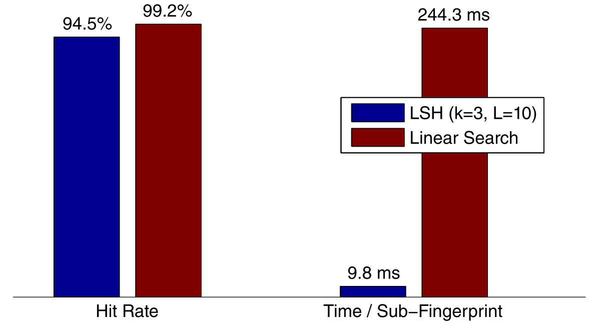

Figure 6 Comparison of accuracy and speed of music identification using LSH and linear search.

For comparison, identification results of the classic Shazam algorithm and state-of-the-art WavePrint [5] are also presented. The Shazam algorithm is implemented by Dan Ellis [44], and implementation of the WavePrint algorithm is available at [45].

Figure 7 compares the robustness against time stretching of the WavePrint, Shazam, and our algorithm. When there is no time stretching, the hit rates of WavePrint, Shazam, and our algorithm are 100%, 99.35%, and 100%, respectively. Under slight time stretching of ±2%, the hit rates of WavePrint and our algorithm remain approximately 100%, and Shazam drops to around 93%. When the query is further time stretched under −5% and +5%, both the WavePrint and our algorithm still maintain hit rates as high as about 99%, while the Shazam quickly drops to 60.78% and 67.7%, respectively. The reason is that the time intervals of key points, which are used to construct the fingerprint in the Shazam algorithm, are destroyed at such a level of time stretching. When queries are stretched at ±10%, both the WavePrint and our algorithm possess hit rates above 95%. However, when stretching factor goes up to −20% and +20%, the WavePrint algorithm begins to be inferior to our algorithm, with hit rates 50% vs. 96% and 80% vs. 98%, respectively. In more extreme cases where the stretch factor is bigger than ±30%, WavePrint's hit rates quickly drop down to below 35%, while our algorithm's results remain surprisingly around 80% or above. In summary, in terms of time stretching, the Shazam, WavePrint, and our algorithm exhibit successive increased robustness, from less than ±5%, to less than ±20%, to bigger than ±30%.

Identification results of differently pitch-shifted queries are shown in Figure 8. As stated in the introduction, pitch shifting of an audio signal can be equivalently modeled as the frequency-axis translation of its logarithmic spectrogram image; consequently, the translation-invariant SIFT image features introduced in the proposed algorithm bring strong robustness to the audio signal against frequency changes. Figure 8 shows that when query music fragments are pitch-shifted at different levels even up to −50% (one octave down) and +100% (one octave up), all hit rates of the proposed algorithm are still above 80%. Note that for our method, there is no linear relationship between identification results and pitch-shifting factors. For example, identification hit rates of −50% and +100% pitch-shifted queries are larger than those of nearby less distorted excerpts. In these two special cases,

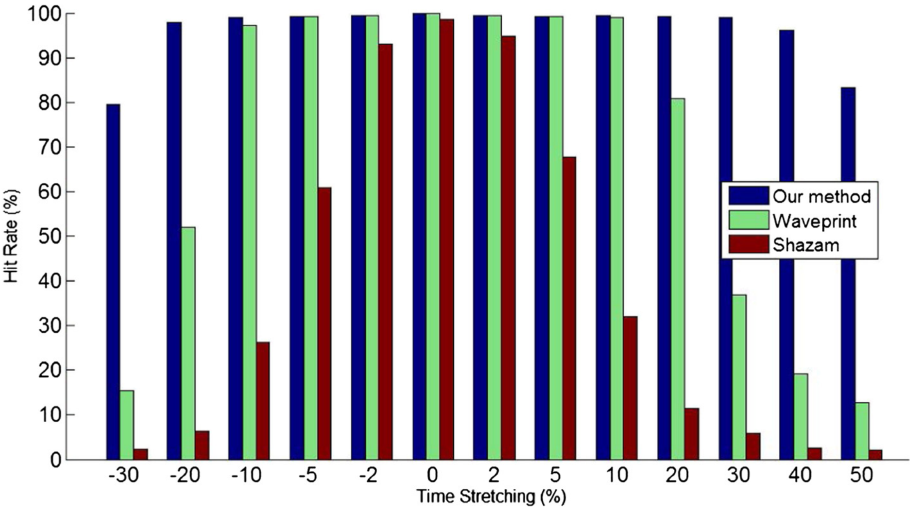

Figure 7 Identification hit rates under time stretching.

pitch shifting occurs on integer-multiples (−12 and +12) of semitone and thus causes more accurate spectrogram translations. Note that performances of the WavePrint algorithm and Shazam system are not displayed in the figure, because these algorithms are by nature very sensitive to frequency misalignments. Even for slight pitch shifting of −2% and +2%, identification hit rates are only about 72.2% and 37.3%, respectively, for WavePrint and about 11.1% and 13.3%, respectively, for Shazam. And when the distortion becomes more serious, the result gets even worse and quickly drops to near zero.

As mentioned in the introduction, time scaling can be approximately modeled as the combination of time stretching and pitch shifting. Therefore, SIFT features calculated from an audio logarithmic spectrogram image should also possess certain robustness against time scaling since they have been demonstrated to be rather stable under time stretching and pitch shifting. Figure 9 illustrates the hit rates with respect to different time-scaling levels. It shows that when music queries are deformed with a common time scaling of −10% ∼ +10%, identification results of our algorithm are pretty good, i.e., all above 98%. When the scaling gets even harder, i.e., to the factors of ±20 and +30%, our algorithm can still obtain hit rates of more than 90%, which outperforms other state-of-the-art algorithms like [3,9,10,17,21] (±6%, ±10%, +5% ±15%, −21% ∼ +26%). Finally, when the music queries are time-scaled up to −30% and +40%,

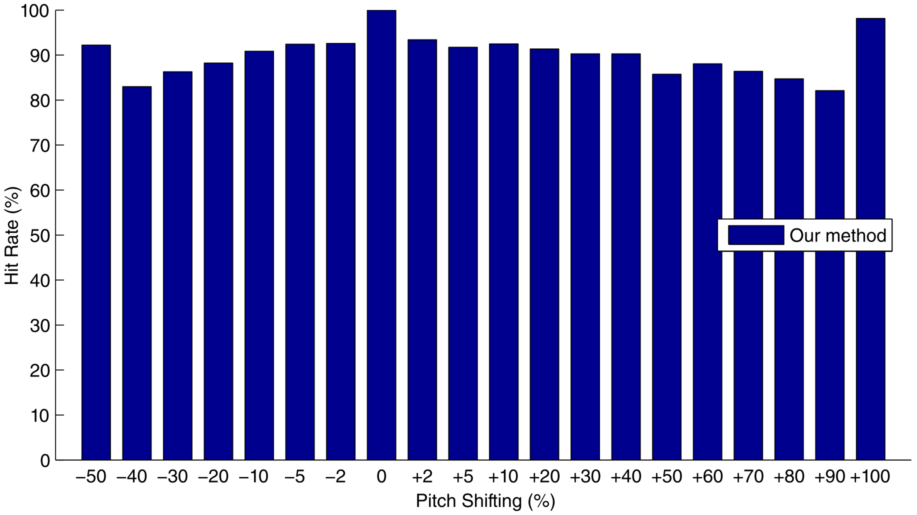

Figure 8 Identification hit rates under pitch shifting.

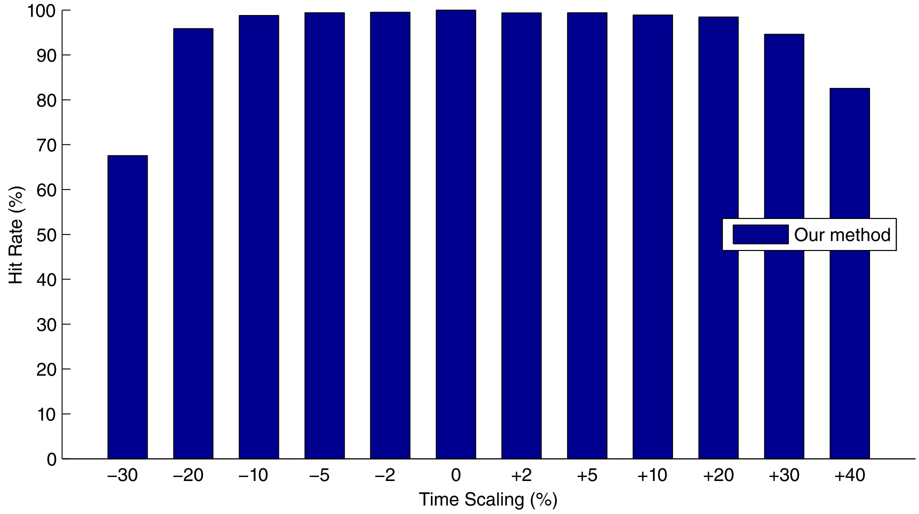

Figure 9 **Identification hit rates under time scaling.**

which has been beyond the scaling scope of previous algorithms' experiments, the hit rates drop to around 70% and 80%, respectively. Note that similar to the case of pitch shifting, identification results of the WavePrint algorithm and Shazam algorithm are neither illustrated in the figure. Due to poor tolerance to pitch distortions, the WavePrint algorithm only exhibits certain robustness against a time scaling of +2% (hit rate = 89.5%). The Shazam algorithm is worse, with only 9.1% and 8.7% hit rates under time scaling of −2% and +2%, respectively. And when the distortion becomes a bit more serious, both of the WavePrint's and Shazam's hit rates drop quickly to zero.

In addition to the above time- and frequency-domain synchronization distortions, music queries are often contaminated by various signal distortions in the real-world environment. Figure 10 compares the robustness against audio signal distortions of the WavePrint, Shazam, and our algorithm. Under the cases of lossy compression, noise addition, resampling, and bandpass filtering, both the WavePrint and our algorithm exhibit almost 100% hit rates. The results of Shazam are also excellent (at least 95%), only slightly weaker. In terms of equalization and echo addition, our algorithm's hit rates drop to around 90% and 80%, respectively, inferior to those of the WavePrint and Shazam. This is as expected, for the two

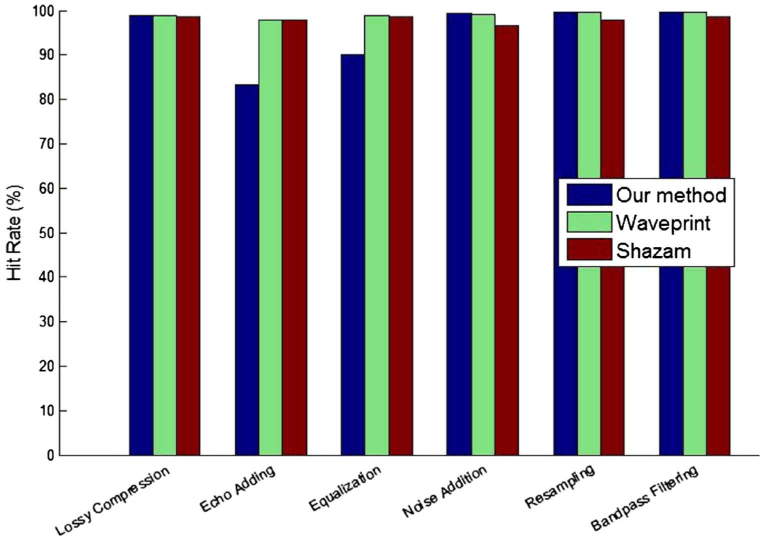

Figure 10 **Identification hit rates under audio signal distortions.**

distortions have greatly affected the energy distributions on spectrograms of the query excerpts and thus have more negative impact on the extraction and matching of SIFT features than other signal distortions.

In the above experiments, the source codes are written in Matlab and run on a workstation (3.2-GHz Intel Xeon CPU and 8-GB memory). The average time of extracting the SIFT features of a 10-s music query is approximately 0.23 s, which is acceptable for the identification task.

6.4 Factor estimation of time stretching and pitch shifting

In this subsection, we assess the factor estimation method of time stretching and pitch shifting proposed in Section 5. The test dataset is composed of 10-s audio excerpts randomly cut from distinct music signals in DB_{test}, and each of them is time-stretched from −30% to +50% and pitch-shifted from −50% to +100%, in accordance with the above robustness tests. After identifying the queries within DB_{test} using the proposed fingerprinting algorithm, the corresponding reference music signal and query signal are compared to estimate the factor of time stretching or pitch shifting in light of Equations (9-11).

Let k_{est} be the estimated factor and k_{ref} be the reference one. The distribution of $k_{\text{est}} - k_{\text{ref}}$ is illustrated in Figure 11 for time stretching and Figure 12 for pitch shifting. As shown in the figures, our proposed method provides highly accurate factor estimation results, and more than 95% of the estimated factors are in the ±0.05 scope of the reference ones. This phenomenon actually demonstrates from a distinct aspect that treating time stretching and pitch shifting of an audio signal as the time-axis stretch and frequency-axis translation of its logarithmic spectrogram, respectively, is a reasonable way to go.

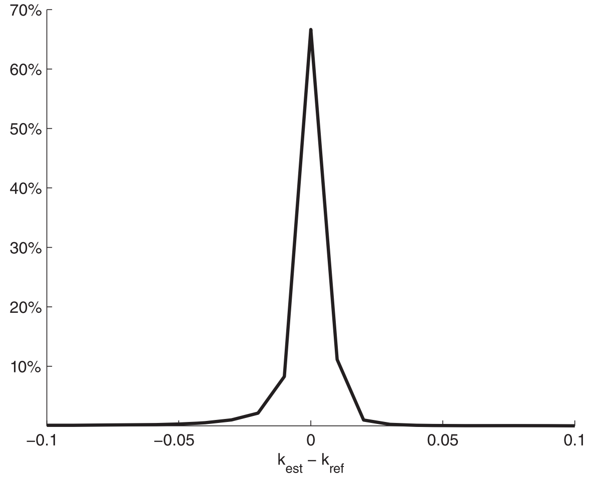

Figure 11 Factor estimation result of time stretching.

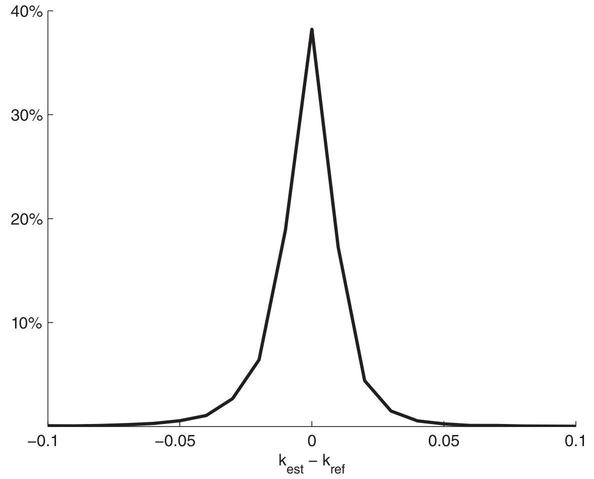

Figure 12 Factor estimation result of pitch shifting.

7 Conclusions

In this paper, a novel and robust music identification method is proposed. By combining computer vision technique, the SIFT descriptor of a spectrogram image to be exact, with locality sensitive hashing, this algorithm exhibits good performance in robustness, accuracy, and speed. What is most attractive is that even when query audio excerpts are seriously time-stretched from −30% to +50% or pitch-shifted from −50% to +100%, this method still exhibits good identification hit rates, which has been beyond all other existing algorithms, to our knowledge. Moreover, by comparing the locations of stable SIFT keypoints, a novel method is developed to estimate the distortion factor of time-stretched or pitch-shifted audio signals. In future work, we intend to combine the proposed SIFT-based feature with other spectral features to further improve the robustness under common audio signal distortions. To apply this proposed feature to other audio-related applications is also an interesting way to go.

Endnotes

[a] In this paper, positive (negative) factors of time stretching/scaling indicate the increase (decrease) of duration of a music piece. For example, +4% (−4%) time stretching/scaling lengthens (shortens) an audio signal to 104% (96%) of its original length. Similarly, positive (negative) factors of pitch shifting mean the increase (decrease) of pitch.

[b] http://www.shazam.com/.

[c] Experiments are performed on a workstation with a 3.2-GHz Intel Xeon CPU and 8-GB memory.

Competing interests
The authors declare that they have no competing interests.

Authors' information

Xiu Zhang received the B.S. degree in computer science and technology from Xidian University, Xi'an, China, in 2012. She is currently a master student in computer science at Fudan University, Shanghai, China. Her research interests include robust audio fingerprinting and audio content authentication.

Bilei Zhu received the B.S., Ph.D degree in computer science from Fudan University, Shanghai, China, in 2009 and 2014, respectively. His research interests include robust audio fingerprinting and audio sound source separation. Now he is working at SAP Labs China.

Linwei Li is currently a senior student in the Department of Physics, Fudan University, Shanghai, China. His research interests include audio fingerprint and automatic music transcription.

Wei Li received the B.S and M.S. degrees from Jilin University, China, in 1992 and 1995, respectively, both in applied physics, and the Ph.D. in computer science from Fudan University, Shanghai, China, in 2004. Since then, he has been with the School of Computer Science and Technology, Fudan University. He is currently a professor, leading the Audio Information Processing and Multimedia Information Security Laboratory. He has published 40 refereed papers in the area of audio information processing and multimedia information security including international leading journals and key conferences such as IEEE TMM, IEEE TASLP, CMJ, EURASIP JASMP, EURASIP JASP, IWDW, ACM SIGIR, and ACM Multimedia. His current research interests include audio fingerprinting, singing voice detection and separation, rhythm analysis, cover song identification, robust speaker recognition, robust audio watermarking, etc.

Xiaoqiang Li received the Ph.D. in computer science from Fudan University, Shanghai, China, in 2004. Since then, he has been with the School of Computer Engineering and Science, Shanghai University, Shanghai, China. He is now an associate professor; his research interests include image processing and analysis and multimedia information security.

Wei Wang received the Ph.D. in computer science from Fudan University, Shanghai, China, in 2012. Since then, he has been with the Naval Medical Research Institute, Shanghai, China. He is now an associate professor; his research interests include image processing and analysis and multimedia information security.

Peizhong Lu received the M.S. degree in mathematics from the University of Information Engineering, Zhengzhou, China, in 1987, and the Ph.D. degree in mathematics from the Chinese Academy of Sciences, Beijing, China, in 1998. In 1998, he joined the Department of Computer Science and Engineering, Fudan University where he is currently a professor in the field of multimedia and communication. His research interests include error-control coding for digital communication, multimedia, and information security. Dr. Lu is a member of the IEEE Information Theory Society and the Communication Society. He received the National Award for an Excellent Doctoral Dissertation in 2000.

Wenqiang Zhang received the Ph.D. in computer science from Shanghai Jiaotong University, Shanghai, China, in 2004. Since then, he has been with the School of Computer Science, Fudan University, Shanghai, China. He is now an associate professor; his research interests is multimedia information processing.

Acknowledgements

This work is supported by NSFC (61171128).

Author details

[1]School of Computer Science, Fudan University, Shanghai 201203, China. [2]SAP Labs, Shanghai 201203, China. [3]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 200433, China. [4]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China. [5]Naval Medical Research Institute, Shanghai 200433, China.

Received: 29 May 2014 Accepted: 15 January 2015
Published online: 12 February 2015

References

1. Y Yu, R Zimmermann, Y Wang, V Oria, Scalable content-based music retrieval using chord progression histogram and tree-structure LSH. IEEE Trans. Multimedia. **15**(8), 1969–1981 (2013)
2. Y Yu, M Crucianu, V Oria, E Damiani, in *Proceedings of ACM International Conference on Multimedia (ACM MM)*. Combining multi-probe histogram and order-statistics based LSH for scalable audio content retrieval (ACM Firenze, Italy, 2010), pp. 381–390
3. F Kurth, T Gehrmann, M Müller, in *Proceedings of the International Society for Music Information Retrieval(ISMIR)*. The cyclic beat spectrum: tempo related audio features for time-scale invariant audio identification (Victoria, Canda, 2006), pp. 35–40
4. Y Ke, D Hoiem, R Sukthankar, in *Proceedings of the, IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer vision for music identification (IEEE San Diego, CA, USA, 2005), pp. 597–604
5. S Baluja, M Covell, Waveprint: efficient wavelet-based audio fingerprinting. Pattern Recognit. **41**(11), 3467–3480 (2008)
6. P Cano, E Batlle, T Kalker, J Haitsma, A review of audio fingerprinting. J. VLSI Signal Process. **41**(3), 271–284 (2005)
7. B Zhu, W Li, Z Wang, X Xue, in *Proceedings of the, ACM International Conference on Multimedia (ACM MM)*. A novel audio fingerprinting method robust to time scale modification and pitch shifting (ACM Firenze, Italy, 2010), pp. 987–990
8. J Haitsma, T Kalker, in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*. A highly robust audio fingerprinting system (Paris, France, 2002), pp. 107–115
9. J Haitsma, T Kalker, in *Proceedings of the, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 4*. Speed-change resistant audio fingerprinting using auto-correlation (IEEE Hong Kong, China, 2003), pp. IV–728
10. JS Seo, J Haitsma, T Kalker, in *Proceedings of the IEEE Workshop on Model based Processing and Coding of Audio*. Linear speed-change resilient audio fingerprinting (IEEE Leuven, Belgium, 2002), pp. 45–48
11. C Bellettini, G Mazzini, A framework for robust audio fingerprinting. J. Commun. **5**(5), 409–424 (2010)
12. S Sukittanon, LE Atlas, JW Pitton, Modulation-scale analysis for content identification. IEEE Trans. Signal Process. **52**(10), 3023–3035 (2004)
13. JS Seo, M Jin, S Lee, D Jang, S Lee, CD Yoo, in *Proceedings of the, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 3*. Audio fingerprinting based on normalized spectral subband centroids (IEEE Philadelphia, Pennsylvania, USA, 2005), pp. iii–213
14. M Malekesmaeili, RK Ward, in *Proceedings of the, International Workshop on Multimedia Signal Processing (MMSP)*. A novel local audio fingerprinting algorithm (IEEE Banff, Canada, 2012), pp. 136–140
15. AL Wang, in *Proceedings of, International Society for Music Information Retrieval (ISMIR)*. An industrial strength audio search algorithm (Baltimore, Maryland, USA, 2003), pp. 7–13
16. S Fenet, G Richard, Y Grenier, in *Proceedings of International Society for Music Information Retrieval (ISMIR)*. A scalable audio fingerprint method with robustness to pitch-shifting (Miami, USA, 2011), pp. 121–126
17. E Dupraz, G Richard, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Robust frequency-based audio fingerprinting (IEEE Dallas, Texas, USA, 2010), pp. 281–284
18. L Worms, *Reconnaissance dextraits sonores dans une large base de données*. (Practical lessons, Ircam, 1998)
19. M Ramona, G Peeters, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Audio identification based on spectral modeling of bark-bands energy and synchronization through onset detection (IEEE Prague, Czech Republic, 2011), pp. 477–480
20. M Ramona, G Peeters, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Audioprint: An efficient audio fingerprint system based on a novel cost-less synchronization scheme (IEEE Vancouver, British Columbia, Canada, 2013), pp. 818–822
21. R Bardeli, F Kurth, in *AES 25th International Conference on Metadata for Audio*. Robust identification of time-scaled audio (Springer London, UK, 2004)
22. RF Lyon, Machine hearing: an emerging field. IEEE Signal Process. Mag. **27**(5), 131–139 (2010)
23. RF Lyon, J Ponte, G Chechik, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Sparse coding of auditory features for machine hearing in interference (IEEE Prague, Czech Republic, 2011), pp. 5876–5879
24. W Li, Y Liu, X Xue, in *Proceedings of the, International ACM SIGIR Conference on Research and Development in Information Retrieval*. Robust audio identification for mp3 popular music (ACM Geneva, Switzerland, 2010), pp. 627–634
25. Y Ke, R Sukthankar, L Huston, in *Proceedings of the, ACM International Conference on Multimedia (ACM MM)*. An efficient parts-based

near-duplicate and sub-image retrieval system (ACM New York, NY, USA, 2004), pp. 869–876
26. K Mikolajczyk, C Schmid, in *Proceedings of the International Conference on Computer Vision (ICCV)*. Indexing based on scale invariant interest points (IEEE Vancouver, British Columbia, Canada, 2001), pp. 525–531
27. D Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision. **60**(2), 91–110 (2004)
28. V Ferrari, T Tuytelaars, L VanGool, Simultaneous object recognition and segmentation from single or multiple model views. Int. J. Comput. Vision. **67**(2), 159–188 (2006)
29. G Yu, JJ Slotine, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Audio classification from time-frequency texture (IEEE Taipei, Taiwan, 2009), pp. 1677–1689
30. G Yu, JJ Slotine, in *Proceedings of the International Conference on Pattern Recognition (ICPR)*. Fast wavelet-based visual classification (IEEE Tampa, Florida, USA, 2008), pp. 1–5
31. T Matsui, M Goto, JP Vert, Y Uchiyama, in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. Gradient-based musical feature extraction based on scale-invariant feature transform (Barcelona, Spain, 2011), pp. 724–728
32. L Kaliciak, B Horsburgh, D Song, N Wiratunga, J Pan, in *Proceedings of the Asia Information Retrieval Societies Conference (AIRS)*. Enhancing music information retrieval by incorporating image-based local features (Tianjin, China, 2012), pp. 226–237
33. J Shi, C Tomasi, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Good features to track (IEEE Seattle, WA, USA, 1994), pp. 593–600
34. L Kaliciak, D Song, N Wiratunga, J Pan, in *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*. Novel local features with hybrid sampling technique for image retrieval (ACM Toronto, Canada, 2010), pp. 1557–1560
35. K Mikolajczyk, C Schmid, A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Machine Intelligence. **27**(10), 1615–1630 (2005)
36. P Indyk, R Motwani, in *Proceedings of the ACM Symposium on Theory of Computing*. Approximate nearest neighbors: towards removing the curse of dimensionality (ACM Dallas, Texas, USA, 1998), pp. 604–613
37. G Shakhnarovich, P Viola, T Darrell, in *Proceedings of the International Conference on Computer Vision(ICCV)*. Fast pose estimation with parameter-sensitive hashing (IEEE Nice, France, 2003), pp. 750–757
38. M Casey, M Slaney, in *Proceedings of the, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 4*. Fast recognition of remixed music audio (IEEE Honolulu, Hawaii, USA, 2007), pp. IV–1425
39. A Auclair, L Cohen, N Vincent, in *Proceedings of the International Workshop on Adaptive Multimedia Retrieval*. How to use SIFT vectors to analyze an image with database templates (Paris, France, 2007), pp. 224–236
40. M Datar, N Immorlica, P Indyk, VS Mirrokni, in *Proceedings of the twentieth annual symposium on Computational Geometry*. Locality-sensitive hashing scheme based on p-stable distributions (ACM Barcelona, Spain, 2004), pp. 253–262
41. G Shakhnarovich, An implementation of locality sensitive hashing algorithm (2008). http://ttic.uchicago.edu/~gregory/download.html
42. X Xue, W Li, Y Yin, in *Proceedings of the ACM International Conference on Multimedia (ACM MM)*. Towards content-based audio fragment authentication (ACM Scottsdale, AZ, USA, 2011), pp. 1249–1252
43. A Vedaldi, B Fulkerson, VLFeat: An open and portable library of computer vision algorithms (2008). http://www.vlfeat.org
44. D Ellis, Robust landmark-based audio fingerprinting (2009). http://labrosa.ee.columbia.edu/~dpwe/resources/matlab/fingerprint
45. C Sergiu, Duplicate songs detector via audio fingerprinting (2012). http://www.codeproject.com/Articles/206507/Duplicates-detector-via-audio-fingerprinting